Maduranga *et al.*
*Journal of Electrical Systems and Inf Technol* (2024) 11:10
https://doi.org/10.1186/s43067-024-00138-0

Journal of Electrical Systems and Information Technology

RESEARCH Open Access

# Improved-RSSI-based indoor localization by using pseudo-linear solution with machine learning algorithms

M. W. P. Maduranga[1], Valmik Tilwari[2*] and Ruvan Abeysekera[1]

*Correspondence:
valmik@korea.ac.kr

[1] IIC University of Technology, Phnom Penh 121206, Kingdom of Cambodia
[2] School of Electrical Engineering, Korea University, Seoul, South Korea

**Abstract**

With the rapid advancement of the Internet of Things and the popularization of mobile Internet-based applications, the location-based service (LBS) has attracted much attention from commercial developers and researchers. Received signal strength indicator (RSSI)-based indoor localization technology has irreplaceable advantages for many LBS applications. However, due to multipath fading, noise, and the limited dynamic range of the RSSI measurements, precise localization based on a path-loss model and multilaterate becomes highly challenging. Therefore, this study proposes a machine learning (ML)-based improved RSSI-based indoor localization approach in which RSSI data is first augmented and then classified using ML algorithms. In addition, we implement an experimental testbed to collect the RSSI value based on Wi-Fi using various reference and target nodes. The received RSSI measurements undergo pre-processing using pseudo-linear solution techniques for closed-form solutions, approximating the original system of nonlinear RSSI measurement equations with a system of linear equations. Finally, the RSSI measurement are trained using ML models such as linear regression, polynomial regression, support vector regression, random forest regression, and decision tree regression. Consequently, the experimental results express in terms of root mean square error and coefficient of determinant compared with various machine learning models with hyper-parameter tuning.

**Keywords:** Internet of Things, Indoor localization, Machine learning, Received signal strength indicator, Artificial intelligence

## Introduction

ML/AI-based IoT application development is considered one of the hot topics among developers as well as academia. Among these IoT applications, location-based applications are critical. A few examples of location-based IoT services are locating people in a shopping complex, locating mobile robots on factory floors, attendance management in smart campuses, etc. In indoor environments, finding the location of a moving object is quite challenging due to Non-Line of Sight (NLOS) environments and multipath fading [1–3]. In indoor wireless localization, additional hardware is not required to get the location information. By employing the broadcasting signals from the sensor node can assess its position. Further, the already implemented Wireless

Sensor Network (WSN) for sensing purposes could be upgraded to know the location without any additional cost. Radio signals from mobile sensor nodes are used as input for an algorithm to estimate the location. Generally, indoor positioning systems are based on wireless technologies such as Bluetooth Low Energy (BLE), Wi-Fi, LoRaWAN, UWB, Zigbee, etc. Each wireless technology has its pros and cons. For instance, BLE has less power consumption and a very short communication range, and LoraWAN has high power consumption and a long sensing range [4, 5].

Numerous of the prominent algorithms available in the study for indoor localization are mainly focus on statistical, deterministic, or filter-based [6–8]. Such algorithms are highly complex and impractical to deploy on real hardware setups. Further, various hardware devices are used in Indoor Positioning Systems (IPS) based on classical algorithms, increasing the cost and significantly limiting the location accuracy.

ML algorithms are mostly employed in localization to extract the signals' essential properties. Based on these derived features, clustering is carried out using the fingerprint method. For NLOS identification and mitigation, feature extraction is also crucial. Current research endeavors focus on advancing machine learning-based indoor localization techniques tailored for IoT systems, enabling their diverse application in innovative scenarios [9–12]. Some works are based on regressor types of algorithms, classifier-type algorithms, or deep learning-based algorithms. Yet, proposed ML models have limitations. Often, proposed methods for ML-based localization are limited to a single ML algorithm, and no comparison of performances with other algorithms is available. Also, few works are based on simulated datasets, and no experimental testbed is implemented and evaluated. Further, there is less or no consideration of hyper-parameter tuning in algorithms.

The main contribution of this study is as follows:

- The RSSI measurement values are gathered using a Wi-Fi-based testbed featuring anchor nodes and target nodes designed using Espressif(ESP) 12 devices, operating on the IEEE 802.11 b/g/n protocol within an indoor environment.
- We introduce a pseudo-linear solution (PLS) as an innovative approach, offering a closed-form solution that approximates the original system of nonlinear RSSI measurement equations with a set of linear equations.
- To effectively manage measurement errors, our PLS method employs a weighted least-squares approach, with the weights carefully determined by considering the statistical properties of errors in both RSSI measurements and reference node locations.
- Finally, the received RSSI data is subjected to training with a selection of ML models: linear regression, polynomial regression, support vector regression, random forest regression, and decision tree regression, followed by a comparative evaluation of their respective performances.

This paper is organized as follows. Section "Related works" explains the recent works available; Section "Experimental testbed design" presents the details of designing the experimental testbed, Section "System model" expresses the details of ML models used and how they were trained; and finally, the results and conclusions.

## Related works

Several studies have been conducted to estimate the precise location of a sensor node in indoor environments with various localization techniques using numerous machine learning algorithms. This section briefly describes the recent studies and highlights the fundamental methodology used for Machine Learning-based indoor localization: In the article [13], the authors have investigated using an ML regressor for indoor localization. The authors of this paper used neural network technologies to carry out localization procedures based on the RSSI parameter. We compared the location estimate outcomes with two approaches (the ANN and the Decision tree) and the RSSI dataset. In order to evaluate the location for each triplet of RSSI, they initially used an artificial neural network with three inputs. We calculated the means error value for each location acquired for this ANN architecture. The same task is done for the ANN architecture with four inputs, where they estimate the location for each of the four inputs and determine the means error value for those estimates.

In [10], Ultra-Wide Band(UWB) has been used as the wireless technology for the Indoor Positioning Systems(IPS). For the UWB IPS system, an ML-based algorithm built on Naive Bayes(NB) principles has been developed. The suggested techniques exhibit a considerable improvement in localization precision. The outcome shows that as the distance between the anchors and tags grows, so does the error between the measured and actual distance. The area under the curve for the NB method is 87%, demonstrating that it has high classification properties. The suggested algorithm will also retain good placement accuracy in both Line of Sight (LoS) and NLoS environments. In work [14], authors analyzed contemporary resolution technologies to locate objects inside buildings accurately. Then, they showed how positioning errors increased when training and testing fingerprinting techniques on various platforms and devices. Received Signal Strength (RSS) computations produce varied results when multiple platform types and devices are used for the precise location and time. The model was trained using Support Vector Machine (SVM) combined with Error-Correcting Output Codes (ECOC) One-Versus-One and Long Short-Term Memory (LSTM) models. To determine the accuracy of the model, Root Mean Square Error (RMSE) was performed to show an error in meters between the true position and the predicted position.

In work [15], detailed comparison of LR, PR, DTR, SVR, and RFR performances for a Wi-Fi-based IPS. According to their findings, the DTR algorithm fared the best as compared with the other algorithms examined. The number of forests in DTR significantly minimizes error and improves location estimation accuracy. It was noted that the accuracy and error were greatly enhanced once the test-reference bed's nodes were increased. Our research predicts that supervised machine learning algorithms will produce better outcomes than deterministic localization.

On the contrary, proposed ML-based methods in related works can provide good accuracy in estimation over classical localization algorithms. However, it can be observed that RSSI is highly fluctuating and needs to apply string filtering techniques and linearization methods over the RSSI dataset before it trains using ML models.

## Experimental testbed design

We designed and implemented the testbed using two sensor nodes: the target node and the reference node. The target node is required to evaluate the position and reference nodes positioned in a fixed position in the indoor location. The experimental setup is established in an electronics engineering laboratory, as shown in Fig. 1. The location is about 8.02 square meters, spanning an open area surrounded by walls, and also consists of some furniture. The IoT architecture used in the RSSI data collection systems is denoted in Fig. 2. Both the target node is implemented using ESP-12E and the anchor nodes are implemented using ESP-01 modules. ESP modules incorporate the IEEE 802.11 standard employed in completely indoor locations (Fig. 3). This system supports IPv4, TCP, MQTT protocol, UDP, and HTTP in communication between nodes. A self-regulating 3.3 V DC power source through an ADP7158 linear regulator was used to power up the nodes, as depicts in Fig. 4a, b. Also, ESP-12E employs a lithium polymer secondary battery source for the storage.

In the testbed arrangement, 34 known location is identified with their x and y axis. Before taking RSSI readings, all the Wi-Fi-enabled devices, such as Wi-Fi access points, were turned off in the environment. During the data collection, the references were fixed on the wall at 2 feet height from the ground level, and the mobile node was kept on marked the places. During the experiment, the mobile node was kept in all 34 locations for one minute, and recorded the RSSI values via an IoT cloud architecture. The actual image of the testbed is shown in Fig. 1.

The RSSI data collection and publication to a cloud storage server are done using the IoT cloud architecture, is shown in Fig. 3. The mobile node's private Wi-Fi network data collection for RSSI is made public on the internet, which is a public network. The hardware platform and the online RSSI data gathering are linked through the IoT cloud. The Internet of Things cloud is a widely dispersed mosquito MQTT broker that publishes the information collected to a distant server. Wi-Fi and internet technologies are used to send the acquired data between the hardware platform and the distant server, respectively. Figure 5 demonstrate the process of location estimating with reference nodes.

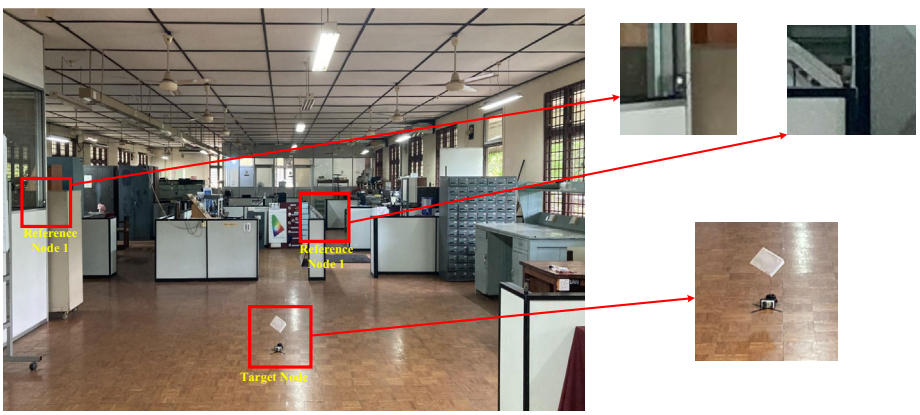



**Fig. 1** Experimental testbed location

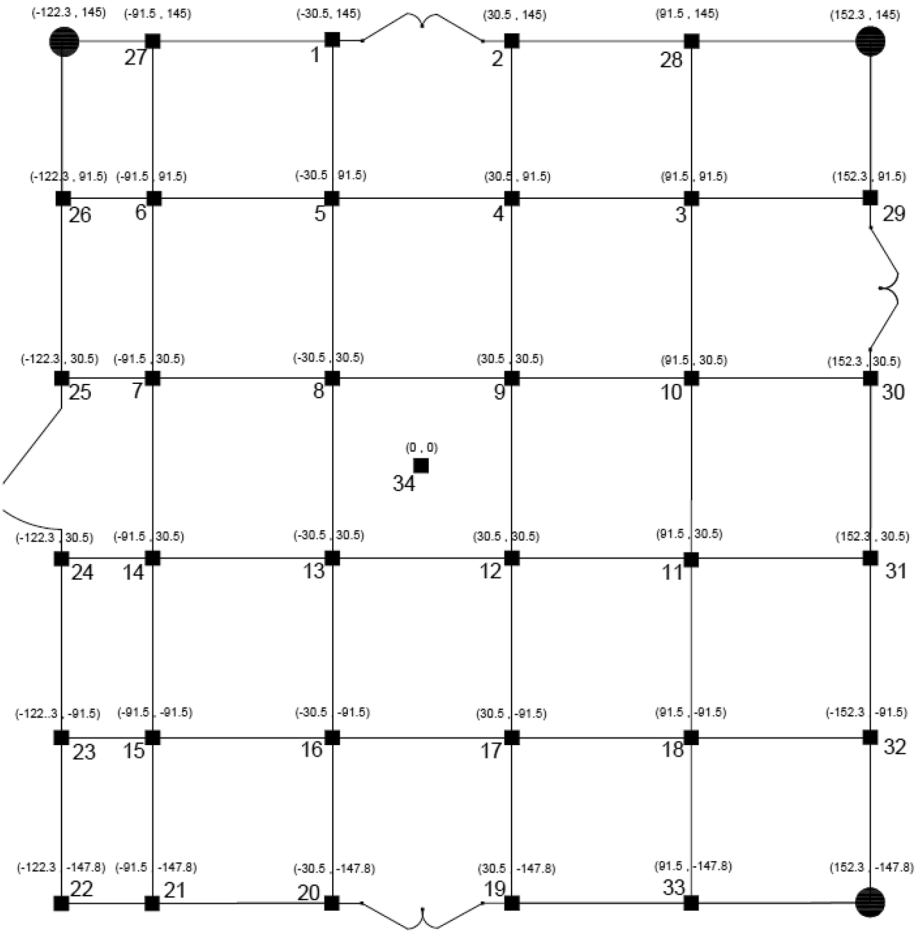



**Fig. 2** Arrangement of reference nodes and mobile nodes

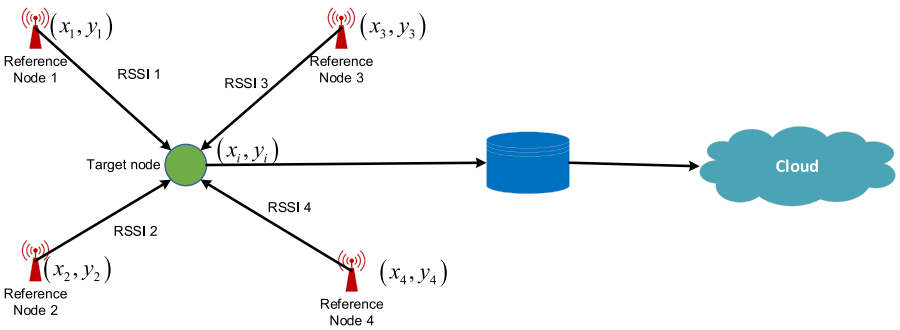
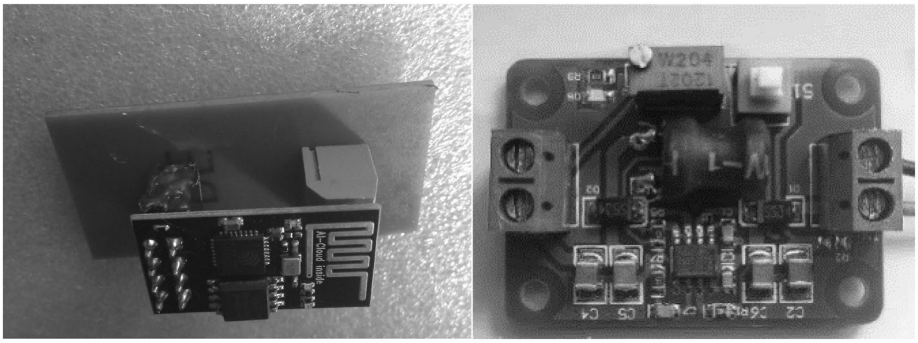



**Fig. 3** The RSSI-based localization system

## System model

The RSSI-based localization of the target node is estimated by using multiple reference nodes. Let the target node is denoted as $(x_b, y_b)$ with the fixed reference node locations at $(x_i, y_i), i = 1, 2, \ldots, M$. i.e., M ≥ 3. The target node's RSSI measurement is included with noise due to signal fluctuation. The noisy reference location at the target node is represented as $(\widetilde{x}_i, \widetilde{y}_i)$ and the subsequent RSSI estimation is represented as $\widetilde{p}_i$. An additive



(a) (b)

**Fig. 4** **a** Reference sensors node, **b** 3.3 V DC

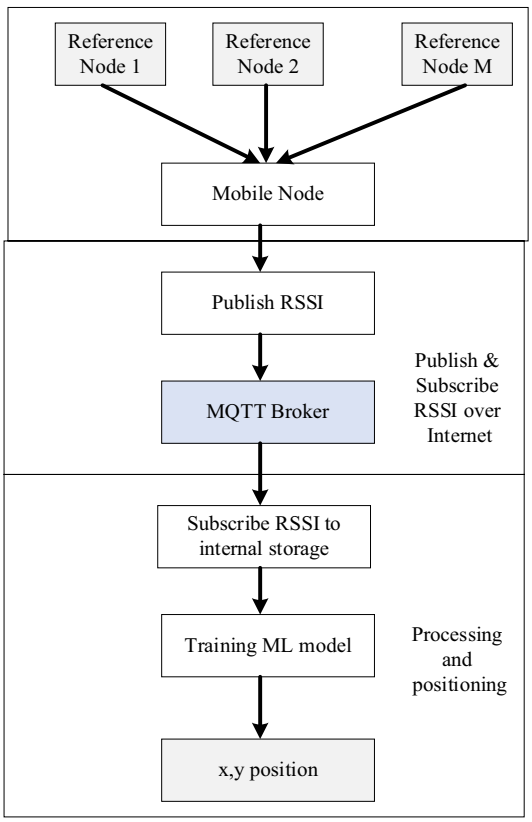



**Fig. 5** Process of location estimating

independent with zero-mean Gaussian noise affects the anchor node location information with a standard deviation indicated as $\sigma_{a_i}$ [16]. There is variation of $\sigma_{a_i}$ values due to the multiples reference nodes. On the other hand, it considers the identical for both the x and y coordinates of a targeted node.

$$\widetilde{x}_i = x_i + n_{x_i} \tag{1}$$

$$\widetilde{y}_i = y_i + n_{y_i} \tag{2}$$

$n_{x_i}, n_{y_i} \sim N\left(0, \sigma_{a_i}^2\right)$

Similarly, the RSSI measurement by log-normal shadowing system model of radio signal path-loss is also employed [17]. So that the target node of the transmitted signal from the $i$th reference nodes is represented as $\tilde{p}_i$ (dBm). The perturbation $n_{\sigma_{p_i}}$ in $\widetilde{p}_i$ is denotes an additive noises with independent zero-mean Gaussian and standard deviation is denoted as $\sigma_{p_i}$ (dB), such that.

$$\begin{aligned} \tilde{p}_i &= \overline{p}_i + n_{\sigma_{p_i}} \\ n_{\sigma_{p_i}} &\sim N(0, \sigma_{p_i}^2) \end{aligned} \tag{3}$$

Moreover, the shadowing path loss system model represents the correlation between the ith mean of the power and the distance among the target source and the ith reference nodes, i.e.,

$$d_i = \sqrt{(x_i - x_b)^2 + \left(y_i - y_b\right)^2} \tag{4}$$

as

$$p_i = p_0 - 10\eta \log_{10} \frac{d_i}{d_0}$$

where $d_0$ defines the reference nodes distance, $p_0$ defines received source power value at the reference distances, and $\eta$ is the pathloss exponent value, respectively. Assumed the perturbed value $p_i$, the RSSI-caused measure of the distance amongst the target source and the ith reference nodes is represented by $\tilde{d}_i$, and it is computed as

$$\tilde{d}_i = d_0 10^{\frac{p_0 - \tilde{p}_i}{10\eta}} \tag{5}$$

This study considers the challenges of computational efficiency and energy resource constraints for location estimation of the target node by using the reference nodes. In this manner, the RSSI location measurement from every reference node is accessible to the target node at any period for localization. To cope with the challenges mentioned as above, this study proposed a PLS to solve the autonomous-localization issue described below:

The basic idea of the proposed algorithm is to find the near-optimal position of the target node that decreases the sum of the squared error values. As denoted earlier, the reference nodes position $\left(x_i, y_i\right)$ and its subsequent distances $d_i$, $i = 1, 2, \ldots, M$, the target node location is computed by intersecting the circles described as

$$(x - x_i)^2 - \left(y - y_i\right)^2 = d_i^2, \quad i = 1, 2, \ldots, M. \tag{6}$$

To cope with the system's nonlinearization nature of Eqs. (6), subtraction of the equation regarding from the i = 1 to the other outcomes in a system of linearization equations is defined as

$$2As = b \tag{7}$$

here

$$A = \begin{bmatrix} x_1 - x_c & y_1 - y_c \\ x_2 - x_c & y_1 - y_c \\ \cdots & \cdots \\ x_i - x_c & y_1 - y_c \end{bmatrix}, b = \begin{bmatrix} d_c - d_1^2 + k_1 - k_c \\ d_c - d_2^2 + k_2 - k_c \\ \cdots \\ d_c - d_M^2 + k_M - k_c \end{bmatrix}, s = \begin{bmatrix} x \\ y \end{bmatrix}, k_i = x_i^2 + y_i^2$$

$x_c = \frac{1}{M}\sum_{i=1}^{M} x_i, y_c = \frac{1}{M}\sum_{i=1}^{M} y_i, d_c = \frac{1}{M}\sum_{i=1}^{M} d_i^2$, and $k_c = \frac{1}{M}\sum_{i=1}^{M} k_i$,

It is observed that Eq. (7) is an over-determined set of nonlinear equations, thus the objective is to find a solution **s** by decreasing the subsequent sum of the square-error function

$$J(s) = \arg\min_s \left[ \|b - As\|_2^2 \right] \tag{8}$$

The solution of (8) is

$$s = \frac{1}{2}\left(A^T A\right)^{-1} A^T b. \tag{9}$$

It is noted that, only noisy information $\tilde{x}_i, \tilde{y}_i$, and $\tilde{d}_i$ are accessible rather than actual $x_i, y_i$, and $d_i$. To factor in the change of the scale as well as numerical attribute values that included with multiple reference node's location and distance estimations of Eq. (8), the minimization of the sum of square errors as

$$\widetilde{J}(s) = \arg\min_s \left[ \left\| W^{\frac{1}{2}}(b - As) \right\|_2^2 \right] \tag{10}$$

where

$$\widetilde{A} = \begin{bmatrix} \widetilde{x}_1 - \widetilde{x}_c & \widetilde{y}_1 - \widetilde{y}_c \\ \widetilde{x}_2 - \widetilde{x}_c & \widetilde{y}_1 - \widetilde{y}_c \\ \cdots & \cdots \\ \widetilde{x}_i - \widetilde{x}_c & \widetilde{y}_1 - \widetilde{y}_c \end{bmatrix}, b = \begin{bmatrix} \widetilde{d}_c - \widetilde{d}_1^2 + \widetilde{k}_1 - \widetilde{k}_c \\ \widetilde{d}_c - \widetilde{d}_2^2 + \widetilde{k}_2 - \widetilde{k}_c \\ \cdots \\ \widetilde{d}_c - \widetilde{d}_M^2 + \widetilde{k}_M - \widetilde{k}_c \end{bmatrix}, \widehat{s} = \begin{bmatrix} \widehat{x} \\ \widehat{y} \end{bmatrix}, \widetilde{k}_i = \widetilde{x}_i^2 + \widetilde{y}_i^2$$

$$\widetilde{x}_c = \frac{1}{M}\sum_{i=1}^{M} \widetilde{x}_i, \widetilde{y}_c = \frac{1}{M}\sum_{i=1}^{M} \widetilde{y}_i, \widetilde{d}_c = \frac{1}{M}\sum_{i=1}^{M} \widetilde{d}_i^2, \text{ and } \widetilde{k}_c = \frac{1}{M}\sum_{i=1}^{M} \widetilde{k}_i,$$

and W denoted as $M \times M$ weighted matrix. Then, the explanation $\widehat{s}$ of (10) is

$$\widehat{s} = \frac{1}{2}\left(\widetilde{A}^T W^{-1} \widetilde{A}\right) \widetilde{A}^T W^{-1} \widetilde{b} \tag{11}$$

To evaluate the weight matrix (W), it is noted that the error vector $\widetilde{b} - \widetilde{A}s$ in (10) contains two noise elements, one is in the reference node's location and another one is in distance measurement. The vector $\widetilde{b}$ comprises the squares of the noise elements, which basically lead the impact of noise in $\widetilde{A}$ to the error vector covariance. Thus, it is considered that the W represents the covariance matrix of $\widetilde{b}$. Thus, $\widetilde{b}$ is simplified as

$$\widetilde{b} = \left(I - \frac{1}{M}11^T\right)b_1 \tag{12}$$

where

$$b_1 = \left[\widetilde{k}_1 - \widetilde{d}_1^2, \widetilde{k}_2 - \widetilde{d}_2^2, \ldots, \widetilde{k}_M - \widetilde{d}_M^2\right]$$

Hence, we have

$$W = \left(I - \frac{1}{M}11^T\right)\mathrm{cov}(b_1)\left(I - \frac{1}{M}11^T\right)$$

where

$$\mathrm{Cov}(b_1) = \mathrm{dig}\left(\mathrm{Var}\left(\widetilde{k}_1 - \widetilde{d}_1^2\right), \mathrm{Var}\left(\widetilde{k}_2 - \widetilde{d}_2^2\right), \ldots., \mathrm{Var}\left(\widetilde{k}_M - \widetilde{d}_M^2\right)\right)$$

Reflecting the assumptions mentioned above is independent features of the noises of the reference node's location and RSSI-induced distances, () is defined as

$$\mathrm{Var}\left(\widetilde{k}_i - \widetilde{d}_i^2\right) = \mathrm{Var}(\widetilde{k}_i) + \mathrm{Var}(\widetilde{d}_i^2)$$

It is notable that the $\widetilde{k}_i$ represent the summation of the square with independent normal distributed random variable $\widetilde{x}_i$, and $\widetilde{y}_i$ as well as a non-zero mean. Thus, variance $\frac{\widetilde{k}_i}{\sigma_{a_i}^2}$ is defined as

$$\mathrm{Var}\left(\frac{\widetilde{k}_i}{\sigma_{a_i}^2}\right) = 2\left(2 + 2\frac{x_i^2 + y_i^2}{\sigma_{a_i}^2}\right)$$

And consequently

$$\mathrm{Var}\left(\widetilde{k}_i\right) = 4\sigma_{a_i}^2\left(\sigma_{a_i}^2 + \left(x_i^2 + y_i^2\right)\right). \tag{13}$$

Thus $\mathrm{Var}\left(\widetilde{d}_i^2\right)$ is computed as [11]

$$\mathrm{Var}\left(\widetilde{d}_i^2\right) = \exp\left(4\mu_{d_i}\right)\left[\exp\left(8\sigma_{a_i}^2\right) - \exp\left(4\sigma_{a_i}^2\right)\right] \tag{14}$$

where

$$\mu_{d_i} = \mathrm{In}\, d_i \,\mathrm{and}\, \sigma_{p_i} = \frac{\mathrm{In}\ 10}{10\eta}\sigma_{p_i}$$

The noisy values of $\widetilde{x}_i, \widetilde{y}_i,$ and $\widetilde{d}_i$ are used to compute Eqs. (13) and (14) because of the actual values $x_i, y_i,$ and $d_i$ are not accessible.

Moreover, it is noted that Eq. (11) has multiple sources of bias. The matrix $\widetilde{A}$ contains noise, the errors in $\widetilde{b}$ are not additive as well as zero-mean, and there is a relationship among the errors in $\widetilde{A}$ and $\widetilde{b}$. To evaluate the bias into the system model algorithm taking an additive error, Eq. (9) is simplified as

$$\widetilde{A} = A + N, \widetilde{b} \approx b + e \tag{15}$$

By using Eqs. (15) and (11), the $E[\widehat{s}]$ is written as

$$\begin{aligned} E[\widehat{s}] = & \underbrace{E\left[\left(\widetilde{A}^T W^{-1}\widetilde{A}\right)^{-1}\right]A^T W^{-1}b}_{I} + \underbrace{E\left[\left(\widetilde{A}^T W^{-1}\widetilde{A}\right)^{-1}N^T\right]W^{-1}b}_{II} \\ & + \underbrace{E\left[\left(\widetilde{A}^T W^{-1}\widetilde{A}\right)^{-1}N^T W^{-1}e\right]}_{III} + \underbrace{E\left[\left(\widetilde{A}^T W^{-1}\widetilde{A}\right)^{-1}A^T W^{-1}e\right]}_{IV} \end{aligned} \tag{16}$$

In Eq. (16), the expansion of $\widetilde{A}^T W^{-1}\widetilde{A}$ to $(A+N)^T W^{-1}(A+N)$, to make the equation simpler has been avoided. It is assumed that part I in Eq. (16) is the correspond to the target node location $\widehat{s}$ and the remaining of the parts, II, III, and IV are the bias parts owing to estimation errors.

Part II provides the bias owing to the noise in $\widetilde{A}$. Part III provides the statistical dependence among $\widetilde{A}$ and $\widetilde{b}$ i.e., $E\left[N^T e\right] \neq 0$. Moreover, part IV provides the non-additive nature of perturbation in $\widetilde{d}_i$ i.e., $E[e] \neq 0$. To compensate of the bias parts II, III, and IV, the expectation for concerning noise covariance is then subtraction in Eq. (11) is written as

$$\widehat{s}_{bc} = \frac{1}{2}\left(\widetilde{A}^T W^{-1}\widetilde{A} - E\left[N^T W^{-1}N\right]\right)^{-1} \times \left\{\widetilde{A}^T W^{-1}\left(\widetilde{b} - E\left[\widetilde{b}\right]\right) - E\left[N^T W^{-1}\widetilde{b}\right]\right\} \tag{17}$$

To compute $E\left[N^T W^{-1}N\right]$ and $E\left[N^T W^{-1}\widetilde{b}\right]$, N can be written as

$$N = N_1 - N_2 \tag{18}$$

where

$$N_1 = \begin{bmatrix} n_{x_1} & n_{y_1} \\ n_{x_2} & n_{y_2} \\ \cdots & \cdots \\ n_{x_M} & n_{y_M} \end{bmatrix}, N_2 = \begin{bmatrix} n_{x_c} & n_{y_c} \\ n_{x_c} & n_{y_c} \\ \cdots & \cdots \\ n_{x_c} & n_{y_c} \end{bmatrix}$$

$n_{x_c} = \frac{1}{M}\sum_{i=1}^{M} n_{x_i}$, and $n_{y_c} = \frac{1}{M}\sum_{i=1}^{M} n_{y_i}$

Thus. We have

$$\begin{aligned} L &= E\left[N^T W^{-1}N\right] \\ &= E\left[N_1^T W^{-1}N_1\right] + E\left[N_2^T W^{-1}N_2\right] - E\left[N_2^T W^{-1}N_1\right] - E\left[N_1^T W^{-1}N_2\right] \end{aligned} \tag{19}$$

Representing (i, j)th is the element of $W^{-1}$ by $w'_{ij}$, and the entries of (19) are estimated as

$$E\left[N_1^T W^{-1}N_1\right] = \text{diag}\left\{\sum_{i=1}^{M} w'_{ii}\sigma^2_{n_{x_i}}, \sum_{i=1}^{M} w'_{ii}\sigma^2_{n_{y_i}},\right\},$$

$$E\left[N_2^T W^{-1} N_2\right] = \text{diag}\left\{\frac{1}{M^2}\sum_{i=1}^{M} w'_{ii}\sigma^2_{n_{x_i}}, \frac{1}{M^2}\sum_{i=1}^{M} w'_{ii}\sigma^2_{n_{y_i}},\right\},$$

$$E\left[N_2^T W^{-1} N_1\right] = \text{diag}\left\{\frac{1}{M}\sum_{i=1}^{M}\sigma^2_{n_{x_i}}\left(\sum_{j=1}^{M} w'_{ji}\right), \frac{1}{M}\sum_{i=1}^{M}\sigma^2_{n_{y_i}}\left(\sum_{j=1}^{M} w'_{ji}\right)\right\},$$

And

$$E\left[N_1^T W^{-1} N_2\right] = diag\left\{\frac{1}{M}\sum_{i=1}^{M}\sigma^2_{n_{x_i}}\left(\sum_{j=1}^{M} w'_{ij}\right), \frac{1}{M}\sum_{i=1}^{M}\sigma^2_{n_{y_i}}\left(\sum_{j=1}^{M} w'_{ij}\right)\right\},$$

The bias owing to the dependence of noises in the $\widetilde{A}$ and $\widetilde{b}$ can be written as

$$E\left[N^T W^{-1}\widetilde{b}\right] = E\left[N_1^T W^{-1}\widetilde{b}\right] - E\left[N_2^T W^{-1}\widetilde{b}\right] \tag{20}$$

where

$$E\left[N_1^T W^{-1}\widetilde{b}\right] = \begin{bmatrix} \frac{2}{M}\sum_{i=1}^{M} x_i\sigma^2_{n_{x_i}}\left(\sum_{j=1}^{M} w'_{ji}\right) \\ \frac{2}{M}\sum_{i=1}^{M} y_i\sigma^2_{n_{y_i}}\left(\sum_{j=1}^{M} w'_{ji}\right) \end{bmatrix}$$

and

$$E\left[N_2^T W^{-1}\widetilde{b}\right] = -\begin{bmatrix} \frac{2}{M^2}\sum_{i=1}^{M} x_1\sigma^2_{n_{x_i}} 1^T W^{-1} 1 \\ \frac{2}{M^2}\sum_{i=1}^{M} y_1\sigma^2_{n_{y_i}} 1^T W^{-1} 1 \end{bmatrix}$$

To compensate of the bias provided by the non-additive feature of the perturbation in the $\widetilde{d}_i$ [part IV in Eq. (16)], $\mathrm{E}\left[\widetilde{b}\right]$ with its i-*th* entry can be computed as

$$E\left[\widetilde{b}_i\right] = \mathrm{E}\left[\widetilde{b}_c - \widetilde{d}_i^2 + \widetilde{k}_i - \widetilde{k}_c\right]. \tag{21}$$

It can be considered that the noise is independent of the reference 's location and RSSI-induced distances; thus Eq. (21) is expressed as

$$E\left[\widetilde{b}_i\right] = E\left[\widetilde{d}_c\right] - E\left[\widetilde{d}_i^2\right] + E\left[\widetilde{k}_i\right] + E\left[\widetilde{k}_c\right]. \tag{22}$$

To compute $E\left[\widetilde{d}_i^2\right]$, it is noted that the $\widetilde{d}_i^2$ employing in Eq. (5) is equal to

$$\widetilde{d}_i^2 = d_i^2 \exp\left(\sqrt{2}un_{p_i}\right)$$

where

$$u = \frac{\text{In } 10}{5\sqrt{2}\eta}$$

Therefore,

$$\begin{aligned} E\left[\widetilde{d}_i^2\right] &= d_i^2 E\left[\exp\left(\sqrt{2}un_{p_i}\right)\right] \\ &= d_i^2 \exp\left(u^2\sigma_{n_{p_i}}^2\right) \end{aligned} \tag{23}$$

It is noted that the value of $u^2\sigma_{n_{p_i}}^2$ always small even though values of $\sigma_{n_{p_i}}$ is high. In this manner, by employing the second-order expansion of the Taylor-series for the function $\exp\left(u^2\sigma_{n_{p_i}}^2\right)$ near to zero, (23) is estimated as

$$E\left[\widetilde{d}_i^2\right] = d_i^2 + d_i^2\left(u^2\sigma_{n_{p_i}}^2 + \frac{u^4\sigma_{n_{p_i}}^4}{2}\right) \tag{24}$$

By considering the assumption, $E\left[\widetilde{d}_c^2\right]$ is correspond to

$$E\left[\widetilde{d}_c\right] = \frac{1}{M}\sum_{i=1}^{M} E\left[\widetilde{d}_i^2\right]$$

The term $\mathrm{E}\left[\widetilde{k}_i\right]$ in (22) corresponds to

$$E\left[\widetilde{k}_i\right] = x_i^2 + y_i^2 + 2\sigma_{n_{p_i}}^2 \tag{25}$$

And mentioned assumption $E\left[\widetilde{k}_c\right]$ develop into

$$E\left[\widetilde{k}_c\right] = \frac{1}{M}\sum_{i=1}^{M} E\left[\widetilde{k}_i\right]$$

Employing (24) and (25) $\mathrm{E}\left[\widetilde{b}\right]$ is expressed as

$$E\left[\widetilde{b}\right] = b + t$$

here the ith entry for the t is

$$t_i = \left(u^2\sigma_{n_{p_i}}^2 + \frac{u^4\sigma_{n_{p_i}}^4}{2}\right)\left(d_c^2 - d_i^2\right) + 2\left(\sigma_{n_{a_i}}^2 - \frac{1}{M}\sum\sigma_{n_{a_i}}^2\right)$$

It is noted that the $d_i$ is not available, thus the subsequent noise measurement values are employed in the estimation of the t.

Computation estimation shows that evaluation of the bias owing to the included of the noise in the $\widetilde{A}$ and $\widetilde{b}$ employing (20) is approximate actual value only when low noise exists in the reference node's location. Thus, it is dependent on the bias on $(x_i, y_i)$ and becomes the poor evaluation performance is provided with higher values of the $\sigma_{a_i}$. The target node estimated location, that is bias compensated in the presented PLS algorithm, the bias-compensated solution $\widehat{s}_{bc}$ in (17), is computed as a closed form equation as:

$$\widehat{s}_{bc} = \frac{1}{2}\left(\widetilde{A}^T W^{-1}\widetilde{A} - L\right)^{-1}\widetilde{A}^T W^{-1}\left(\widetilde{b} - t\right) \tag{26}$$

## Data collection and pre-processing

The CloudMQTT IoT server is used as an IoT cloud platform in this experiment. The MQTT broker delivers a lightweight mode to perform message. MQTT employs a different model called pub-sub instead of the more common Request/Response for communication on networks that protocols like HTTP use. The Req/Res model's foundation is the server/client architecture. In this architecture, clients communicate by addressing a particular server with a request. The server then responds by giving the client the requested data or service. In this architecture, the client should ask a certain server directly. For web or mobile apps that require one or more strong servers to fulfill client requests, the Req/Res architecture works perfectly [18]. The MQTT protocol has certain advantages, such as using certain Quality of Service (QoS) settings that can ensure delivery; regardless of the condition of the subscribing server, a device can publish its data. When it is ready, the subscribing server can connect and receive the data. The mobile nodes publish RSSI data from the three reference nodes to the MQTT broker over the internet and subscribes to a remote storage server for the RSSI data. The collected raw data set is illustrated in Fig. 6 in both the time and frequency domains. Where normalized frequency values provide valuable insights in frequency content analysis, noise and interference detection and allows to compare results and findings in a standardized manner.

## Machine learning models development

Supervised ML methods are used to predict the position of the mobile sensor node. Generally, these ML algorithms are applied in two stages. Data acquired and delivered to the algorithm in the initial step, the training stage, so it may learn patterns and build a model to categorize data or forecast its attributes. A new dataset is compared to the model created at time of training phase in the second step, known as the testing phase, to determine the model's efficacy. Supervised learning algorithms are a type of two-phase learning algorithm. In this work, it has trained SVM, LR, PR, DTR, and RFR. All the machine algorithms are implemented using Python 3 on Jupiter's Notebook

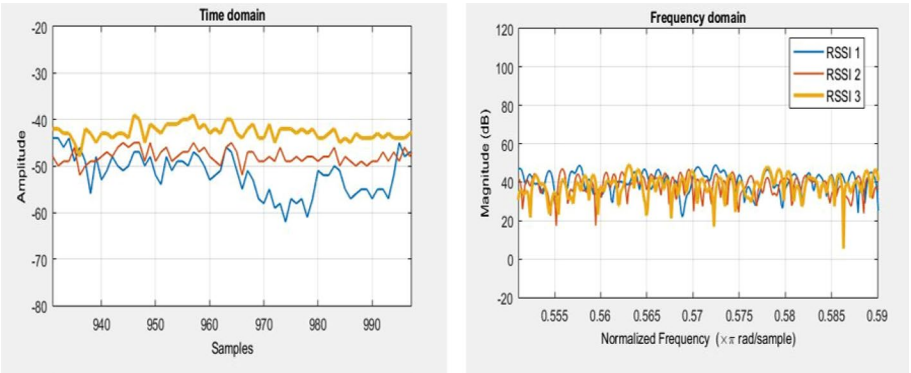



**Fig. 6** Time-domain and frequency-domain representation of RSSI signals

using Sci-kit-learn machine learning library on Intel(R) Core (TM) i5-10210U CPU @ 1.60 GHz 2.11 GHz. For visualizations, MATLAB 2020R is used.

#### Linear regression (LR)

Linear regression (LR) could consider the simplest ML algorithm available. In LR, it is the best-fit linear line between the independent and dependent variables. Defining the best-fit linear line and the ideal intercept and coefficient value so that the error is decreased is the major aim of a LR model. The first variable is the independent variable, whereas the second is regarded as a dependent variable. Moreover, this algorithm is easy to implement and requires less computational power to train the model [19, 20].

#### Polynomial regression (PR)

Polynomial regression is the improved version of the LR. As a specific case of multiple LR, PR is a kind of linear regression that assess the connection as a nth-degree polynomial. PR is suitable for scenarios such as when the dataset consists of nonlinear data. In such scenario, LR fails to create a best-fit line. Consider the accompanying graphic, it depicts a nonlinear correlation, and the outcomes of LR, which accomplish poorly and are not at all realistic. To cope this challenges, PR is used, which identifies the curvilinear correlation between the independent and dependent variables. Moreover, this model is also less complex and easy to implement in even low-power hardware devices [21, 22].

#### Support vector regressor (SVR)

SVR is a powerful ML algorithm used in indoor localization. It is more effective since SVM models linear and nonlinear relations with superior generalization performance and adopts the kernels technique to detect the difference among two points of the two distinct classes. However, when the number of SVs increases, SVM-based approaches become time-consuming and memory-intensive [23, 24].

#### Decision tree regression (DTR)

A decision tree is a supervised machine learning method that could be employed to cope classification and regression challenges, although it is utmost frequently used when coping with classification challenges. It is a tree-structured classifier, in which internal nodes characterize the feature of a datasets, and branches shows the procedure of making decisions, and each leaf node is the classification result. There are basically two nodes such as decision node and leaf node. When it comes to indoor localization, compared to other categorization techniques like K-NN and Neural Network, Decision Tree-based indoor localization performs better in terms of increasing localization accuracy. When the Decision Tree categorizes continuous numerical data, there is a chance that some information will be missed [25, 26].

#### Random forest regression (RFR)

A machine learning ensemble technique using many decision trees is called a random forest regression (RFR). A voting system is employed in RFR to raise the performance of numerous weak students (in this case, decision trees). The primary properties of random forests include random feature selection, bootstrap sampling, out-of-bag error estimates,

and full-depth decision tree growth. Random forest improves the performance of regression trees by combining several regression trees. Using a random forest eliminates the need for cross-validation because the forest is constructed using native out-of-bag error estimates. In some tests, the out-of-bag error estimation is considered impartial [27].

## Result and discussion

Algorithms, DTR, LR, PR, SVM, and RFR are used to train supervised machine learning algorithms to estimate the x and y geographical coordinates of the target node. For all the models, the coefficient of determination ($R^2$) and the Root Mean Squared Error (RMSE) were calculated. Firstly, the experiment taking place with three reference nodes, and step by step, the number of anchor nodes elevate to four and five, respectively, and new data sets were generated. Finally, RMSE and $R^2$ were calculated under different hyper-parameter conditions.

### Root mean squared error

Figure 7a, b denotes the RMSE values changes in the x coordinate as we change the number of anchor nodes for the x coordinate and y coordinate, respectively. In the experimental setup, we changed the number of anchor nodes to 3, 4, and 5, respectively. In each case, RSSI values were collected and trained using ML models. It observed that as the number of anchor nodes increases, there is a significant reduction in RMSE values for all the models. The LR and PR show the higher RMSE values and SVR, DTR, and RFR show relatively lower RMSE values. Where DTR outperformed in terms of RMSE. This trend is because the model trains very well when the number of trainable parameters increases.

Figure 8a, b denotes the RMSE value variation against the sample size for the x coordinate and y coordinate, respectively. It is observed that RMSE decreases as the number of samples increases in all the models. LR and PR showed relatively high RMSE and SVR, while DTR and RFR showed the lowest RMSE values. Where DTR is outperformed for both coordinates, giving the lowest RMSE value. For all the models, the RMSE value decreases as the number of samples increases. In ML models, the standard deviation decreases as the number of samples increases.

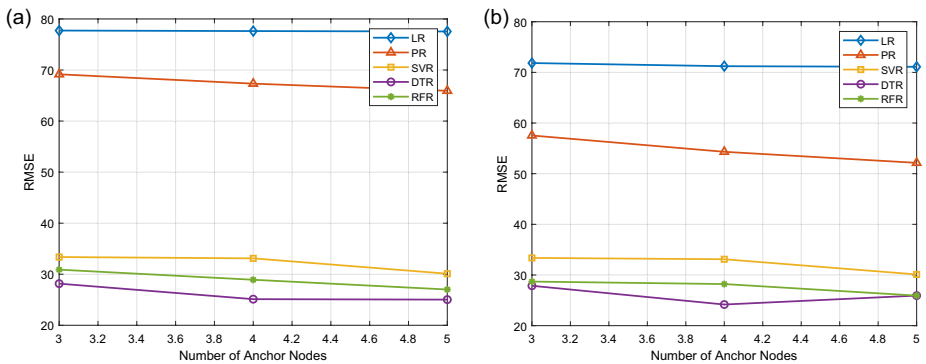



**Fig. 7** RSME value with the number of anchor nodes **a** x coordinate, **b** y coordinate

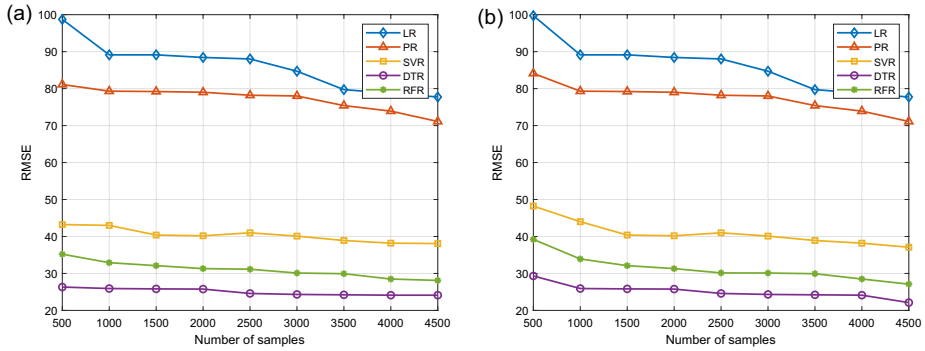



**Fig. 8** RSME value with number of samples **a** x coordinate, **b** y coordinate

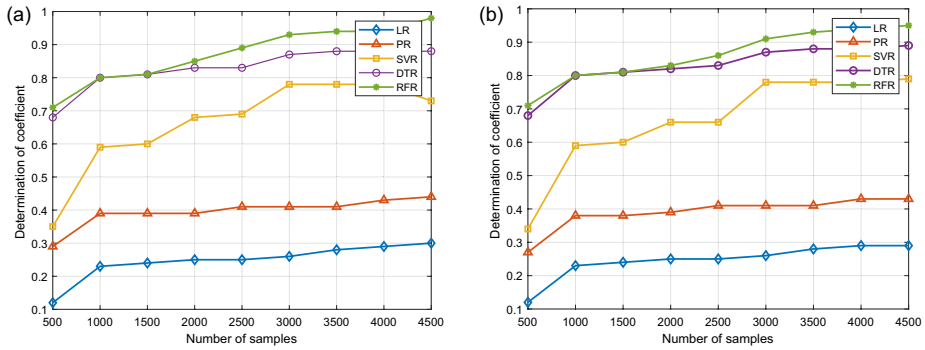



**Fig. 9** Coefficient of determination with number of samples **a** x coordinate, **b** y coordinate

### Coefficient of determination ($R.^2$)

Figure 9a, b shows the change of coefficient value determination against the number of samples for the x coordinate and y coordinate, respectively. For machine learning models, the coefficient of determination, or R-squared value, ranges from 0.0 to 1.0 and reflects the correlation of the variance proportionate to the real and estimated node position. All dataset points perfectly lie at the estimated line of best fit when the R-squared values are closer to 1.0, indicating that the estimated position is entirely defined concerning the higher accuracy. For all the models, $R^2$ values rapidly increase till 1000 samples, and after 1000, it increases normally. DTR and RFR show better $R^2$ score, which is closer to 1. LR and PR show less than 0.5, meaning that models do not fit well with the data.

### Hyper-parameter of the ML models

Figure 10a shows the impact of the hyper-parameter and the number of forests in RFR against the accuracy of the estimation. It can be observed that as the number of forests increases, RMSE is significantly decreasing. In RFR as the number of forests increases, the model is well trained with the data and gives better accuracy. However, the model required a higher computational power in hardware devices with a high number of forests.

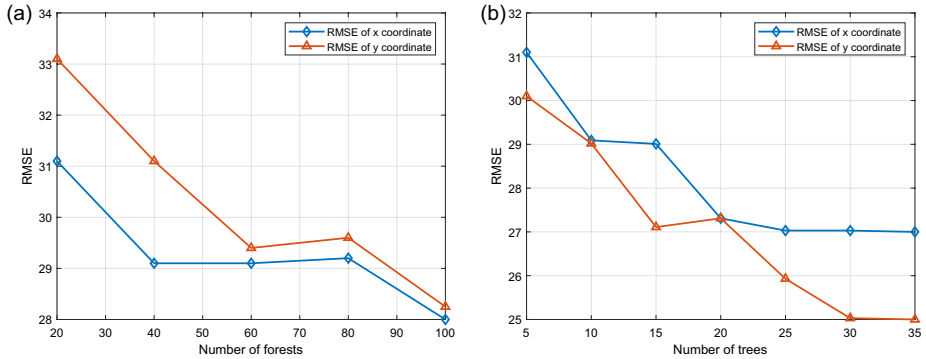



**Fig. 10** RSME value for x and y coordinated **a** number of forests, **b** number of trees

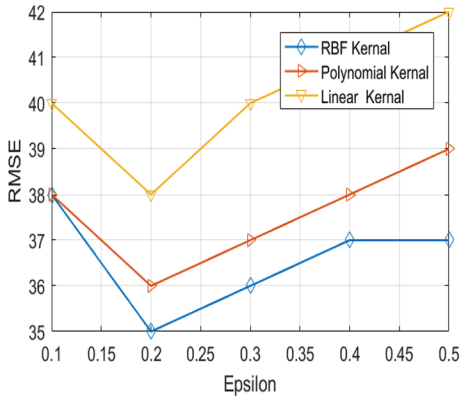



**Fig. 11** RSME value versus epsilon

The number of tree hyper-parameters used in tree-based ensemble methods must be adjusted, directly affecting the computational cost. Sufficient trees must be chosen to find a trade-off between forecast accuracy and computational time. According to the foundations of tree-based algorithms, a model with more trees will be optimized and have the lowest possible prediction error. It shows that model performance depends on the maximum tree depth and that deeper trees perform better. Figure 10b illustrates the impact of the number of trees versus RMSE in the DTR algorithm. It can be observed that RMSE is significantly decreasing as the number of trees increases.

#### RMSE value with the epsilon for different kernel functions in SVR

Figure 11 illustrates the change of RMSE value against the epsilon for different kernel functions in SVR. Firstly, the input dataset forwarded into the kernels, which then transforms it into the desired form. Various SVM algorithms use different kernel functions. There are several forms of these functions. For instance, linear, nonlinear, polynomial, sigmoid, and radial basis functions (RBF). Describe the kernel functions for vectors, text, pictures, graphs, and sequence data. RBFs are the utmost prevalent types of kernel functions. since it responds locally and infinitely throughout the entire x-axis. The kernel functions return the inner product between two locations in an appropriate feature space. Thus, a notion of similarity is defined even in very high-dimensional areas with

low computational expense. The experimental results show that all the kernel functions are giving decrement RMSE from 0.1 to 0.2 and after $\varepsilon > 0.2$, RMSE is rapidly increasing. Based on the observations, the RBF kernel is outperformed.

#### RMSE value with the C parameter in SVR

Figure 12 illustrates the RMSE value change against the c parameter in SVR. Where gamma set 0.1 for RBF kernel. It is observed that when *C* is increasing, RSME is significantly decreasing. For each erroneously classified data point, the *C* parameter provides a penalty value. In the event that c is low, selecting a decision boundary with a high margin comes at the expense of more misclassifications for the reason that the penalty for incorrectly classified points is low. SVM attempts to decrease the number of erroneously classified instances owing to a high penalty when *C* is large, which leads to a decision boundary with a narrower margin. Not all instances of misclassification get a similar penalty. It is contrarily relationship with the partition from the decision boundary.

## Conclusions

This study presents an ML-based approach that could apply to robust indoor location scenarios. An experimental testbed was designed, including five reference nodes and one target node. The target node was placed at known geographic coordinates, and RSSI data were gathered using an IoT cloud architecture. The collected dataset was pre-processed using a PLS for a closed-form solution. It approximated the original system of nonlinear RSSI measurement equations with a system of linear equations. The dataset was trained using several ML algorithms. It is evident from the experiment with many supervised algorithms under various circumstances that the DTR outperformed the other algorithms that experimented the best. Hyper-parameters, number of trees in DTR, number of forests in RFR, penalty parameter, and explosion in SVR significantly affect localization accuracy. Moreover, accuracy and error were greatly improved once the reference nodes of the network are increased. Future research can delve into creating and refining ensemble-type machine-learning models designed to enhance indoor localization accuracy. These models can leverage the strengths of various algorithms and techniques, combining them synergistically to improve localization performance. Investigating novel ensemble strategies and assessing their effectiveness in real-world scenarios will

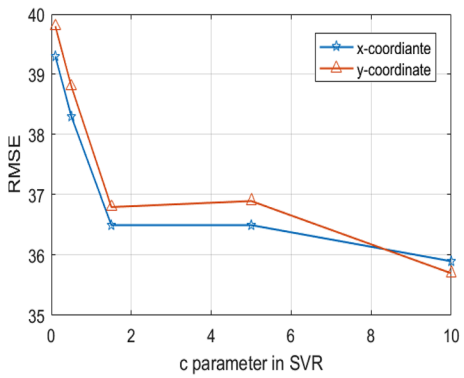



**Fig. 12** RSME value with C parameter in SVR

be crucial. Research efforts should focus on accommodating dynamic indoor environments, diverse IoT device types, and varying network conditions. This will help ascertain the adaptability of the models to a wide range of real-world settings.

**Acknowledgements**
Not applicable.

**Author contributions**
MWP and VT: Writing—original draft, software. MWP, VT and RA: Writing—original draft, reviewing, and editing.

**Funding**
This research did not receive any specifc grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Availability of data and materials**
Not applicable.

## Declarations

**Competing interests**
The authors declare that there is no conflict of interest regarding the publication of this paper.

Received: 4 August 2023 Accepted: 2 February 2024
Published online: 17 February 2024

**References**

1. Zafari F, Gkelias A, Leung KK (2019) A survey of indoor localization systems and technologies. IEEE Commun Surv Tutor 21(3):2568–2599
2. Fonseka P, Sandrasegaran K (2018) Indoor localization for IoT applications using fingerprinting. IEEE, pp 736–741
3. Ibwe K, Pande S, Abdalla AT et al (2023) Indoor positioning using circle expansion-based adaptive trilateration algorithm. J Electr Syst Inf Technol 10:10. https://doi.org/10.1186/s43067-023-00075-4
4. Mohar SS, Goyal S, Kaur R (2018) A survey of localization in wireless sensor network using optimization techniques. IEEE, pp 1–6
5. Sandamini C, Maduranga MWP, Tilwari V, Yahaya J, Qamar F, Nguyen QN, Ibrahim SRA (2023) A Review of Indoor Positioning Systems for UAV Localization with Machine Learning Algorithms. Electronics 12:1533. https://doi.org/10.3390/electronics12071533
6. Maduraga MWP, Abeysekara R (2021) Comparison of supervised learning-based indoor localization techniques for smart building applications. In: 2021 international research conference on smart computing and systems engineering (SCSE), Colombo, Sri Lanka, pp 145–148. https://doi.org/10.1109/SCSE53661.2021.9568311
7. Mingyi YOU, Annan LU (2021) A robust TDOA based solution for source location using mixed Huber loss. J Syst Eng Electron 32(6):1375–1380
8. Yongsheng Z, Dexiu HU, Yongjun Z, Zhixin LIU (2020) Moving target localization for multistatic passive radar using delay, Doppler and Doppler rate measurements. J Syst Eng Electron 31(5):939–949
9. Rahman SA, Tout H, Talhi C, Mourad A (2020) Internet of things intrusion detection: centralized, on-device, or federated learning? IEEE Netw 34(6):310–317. https://doi.org/10.1109/MNET.011.2000286
10. Kimothi S, Thapliyal A, Singh R, Rashid M, Gehlot A, Akram SV, Javed AR (2023) Comprehensive database creation for potential fish zones using IoT and ML with assimilation of geospatial techniques. Sustainability 15:1062. https://doi.org/10.3390/su15021062
11. Kherraf N, Alameddine HA, Sharafeddine S, Assi CM, Ghrayeb A (2019) Optimized provisioning of edge computing resources with heterogeneous workload in IoT networks. IEEE Trans Netw Serv Manag 16(2):459–474. https://doi.org/10.1109/TNSM.2019.2894955
12. Okereke GE, Bali MC, Okwueze CN et al (2023) K-means clustering of electricity consumers using time-domain features from smart meter data. J Electr Syst Inf Technol 10:2. https://doi.org/10.1186/s43067-023-00068-3
13. Gadhgadhi A, Hachaĭchi Y, Zairi H (2020) A machine learning based indoor localization. IEEE, pp 33–38
14. Abbas HA, Boskany NW, Ghafoor KZ, Rawat DB (2021) Wi-Fi based accurate indoor localization system using SVM and LSTM algorithms. IEEE, pp 416–422
15. Maduranga MWP, Abeysekara R (2021) Supervised machine learning for RSSI based indoor localization in IoT applications. Int J Comput Appl 183(3):26–32
16. Itoh KI, Watanabe S, Shih JS, Sato T (2002) Performance of handoff algorithm based on distance and RSSI measurements. IEEE Trans Veh Technol 51(6):1460–1468
17. Schulten H, Kuhn M, Heyn R, Dumphart G, Trosch F, Wittneben A (2019) On the crucial impact of antennas and diversity on BLE RSSI-based indoor localization. IEEE, pp 1–6
18. Yang B, Guo L, Guo R, Zhao M, Zhao T (2020) A novel trilateration algorithm for RSSI-based indoor localization. IEEE Sens J 20(14):8164–8172
19. Jianyong Z, Haiyong L, Zili C, Zhaohui L (2014) RSSI based bluetooth low energy indoor positioning. IEEE, pp 526–533
20. Chen W-C, Kao K-F, Chang Y-T, Chang C-H (2018) An RSSI-based distributed real-time indoor positioning framework. IEEE, pp 1288–1291

21. Goldoni E, Savioli A, Risi M, Gamba P (2010) Experimental analysis of RSSI-based indoor localization with IEEE 802.15. IEEE, pp 71–77
22. Nazir U, Shahid N, Arshad MA, Raza SH (2012) Classification of localization algorithms for wireless sensor network: a survey. IEEE, pp 1–5
23. Zhang L, Peng H, He J, Zhang S, Zhang Z (2022) Three-dimensional localization algorithm of mobile nodes based on received signal strength indicator-angle of arrival and least-squares support-vector regression. Int J Distrib Sens Netw 18(7):15501329221111960
24. Wu S, Huang W, Li M, Xu K (2022) A novel RSSI fingerprint positioning method based on virtual AP and convolutional neural network. IEEE Sens J 22(7):6898–6909
25. Lapčak M, Ovseník LU, Oravec J, Zdravecký N (2022) Design of hard switching for FSO/RF hybrid system based on prediction of RSSI parameter and environmental conditions. IEEE, pp 1–6
26. Hassen WF, Mezghani J (2022) CNN based approach for indoor positioning services using RSSI fingerprinting technique. IEEE, pp 778–783
27. Jia B, Liu J, Feng T, Huang B, Baker T, Tawfik H (2022) TTSL: an indoor localization method based on temporal convolutional network using time-series RSSI. Comput Commun 193:293–301

## Publisher's Note