Indonesian Journal of Electrical Engineering and Computer Science
Vol. 37, No. 3, March 2025, pp. 2021∼2031
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v37.i3.pp2021-2031

Assessing fingerprinting and machine learning approaches for wireless indoor localization

Azkario Rizky Pratama[1], Muhammad Evan Anindya Wahyuaji[1], Muhammad Fadhil Nur Hidayat[1], Bimo Sunarfri Hantono[1], Nur Abdillah Siddiq[2]
[1]Department of Electrical and Information Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia
[2]Department of Nuclear Engineering and Engineering Physics, Universitas Gadjah Mada, Yogyakarta, Indonesia

Article Info

Article history:

Received Apr 2, 2024
Revised Sep 29, 2024
Accepted Oct 7, 2024

Keywords:

Ambient intelligence
Bayesian estimation
Bluetooth low energy
Fingerprint feature extraction
Fingerprinting
Indoor localization
Machine learning

ABSTRACT

This paper presents a comparative analysis of fingerprinting and machine learning techniques for bluetooth low energy (BLE)-based localization. Two fingerprinting algorithms, namely fingerprint feature extraction (FPFE) and Bayesian estimation (BE), along with various machine learning approaches including support vector regression (SVR), ensemble learning, and instance-based learning, are investigated. The selection of techniques depends on the availability of training data or the fingerprint database, explored in both ideal scenario and real-world scenario. In ideal scenario where the system administrator can collect fingerprint data through users' devices, FPFE emerges as the preferred algorithm, achieving superior performance with a mean error of 0.50 m. In the context of real-world scenario, where data collection from multiple devices is limited, the system administrator may gather fingerprint data for localization using one or a few specific devices. Our experiments reveal that when there is a scarcity of fingerprint data, BE and SVR exhibit acceptable performance, reaching a mean error of 1.785 m and 1.965 m, respectively.

Corresponding Author:

Azkario Rizky Pratama
Department of Electrical and Information Engineering, Universitas Gadjah Mada
Yogyakarta, Indonesia
Email: azkario@ugm.ac.id

1. INTRODUCTION

Wireless positioning and localization techniques are crucial for various applications, including indoor navigation, asset tracking, and location-based services. However, one of the biggest challenges in achieving accurate localization in large deployments is terminal heterogeneity, such as the presence of smartphones from different brands in indoor environments [1]. While solutions have been proposed to address the localization of heterogeneous devices indoors, such as the development of more robust algorithms and standardization efforts, many researchers still struggle to achieve optimal localization performance [2]. In this research, we aim to investigate and evaluate different methods to understand their performance under varying conditions. This study seeks to provide valuable recommendations for researchers developing indoor localization systems (ILS).

There are essentially two prominent methods for localization: fingerprinting and machine learning-based approaches. Fingerprinting and machine learning methods each have unique strengths and weaknesses. Fingerprinting, a traditional technique, involves creating a reference database of pre-collected signal characteristics from known locations [3]. These fingerprints are then compared to the measured signal characteristics to estimate device location. On the other hand, machine learning-based approaches generate a learning model trained on data, which can then predict device location using captured signal data. Data collection is a critical aspect of both techniques. Fingerprinting requires extensive site surveys to collect signal characteristics at different locations, while machine learning-based approaches rely on data capturing with ground truth location information for model training. The latter requires diverse and representative datasets to ensure robustness and generalization.

The selection of the most appropriate method is generally based on the evaluation of classification systems. Generally, fingerprinting can provide high performance in well-surveyed environments, but it may suffer from decreased accuracy due to changes in environmental conditions or the introduction of new devices. Machine learning-based approaches, with their ability to learn complex relationships between signal characteristics and locations, offer adaptability and can handle variations better. However, claims of effectiveness of these two approaches cannot be directly compared each other, due to different metrics and procedures to measure the performance of the different indoor localization proposals [4], [5]. In fact, different experiment configuration and procedures, such as beacon density, may affect localization performance. While higher density usually bring higher localization accuracy [6], this relationship (i.e., density-accuracy) may not be straightforward beyond a certain threshold [7].

Researchers often use different assessment method to either measure the quality of hard decision or the quality of system scores [8]. The earlier, the classifier directly outputs the predicted class label for each instance in the dataset without any additional information. For example, Subakti *et al.* [9] achieved an average localization error of 0.68 m in a 5 m x 8 m area using 4 beacon nodes. This work outperforms other methods that also use the same average error metrics [10]-[13]. Alternatively, community also accept the measure of quality of the system. For example, the Bayesian estimator (BE) proposed by Faragher and Harle [7] reported that an error of less than 3 m is achieved 95% over the time (cumulative probability). Researchers often struggle to choose the best method due to differences in metrics used and experiment procedures, resulting in suboptimal performance. In this paper, we investigate various techniques in both domains (i.e., fingerprinting and machine learning-based) to determine the most suitable approach for indoor localization in specific scenarios. This study contributes by demonstrating the adaptability of various methods across diverse conditions and offering recommendations for implementing ILS.

The structure of the paper is as follows: section 2 provides an overview of related work, encompassing bluetooth low energy (BLE) fingerprinting and machine learning. In section 3 details the obtained data and localization techniques. The evaluation of various scenarios employing diverse techniques is presented in section 4. Finally, section 5 summarizes and discusses the findings presented in the paper.

2. RELATED WORKS

Various methods have been employed for object localization in a given space, often relying on either fingerprinting or machine learning techniques [14]. Fingerprinting is a technique that involves collecting and using unique signatures (or "fingerprints") of a specific area to determine the location of an object. Conversely, machine learning techniques leverage data-driven models to estimate location based on signal characteristics.

2.1. Machine learning

Many researchers have delved into the use of machine learning methods to determine the location of tracked objects, whether in terms of coordinates or within specific room locations [15], [16]. Bai *et al.* [17] combine trilateration-based method and fingerprinting-based method before supplying to the machine learning classifier. Furthermore, the location classification in this study divides the location into 36 grids (each grid is 1 m $\times$ 1 m) and the machine learning classifier is assigned to classifying the existing data according to the grid. The authors reported a good accuracy of more than 90% with various machine learning classifier methods such as Naive Bayes (NB), sequential minimal optimization (SMO), random forest (RF), BayesNet, and J48.

Maduranga and Abeysekera [18] utilize feed forward neural network (FFNN) in classifying a location. The authors divide the space into four zones, and assign the input into the zones. The experiment takes place on the first floor of Western Michigan University's Waldo library. The FFNN model has successfully predicted the location with 86% accuracy. Sthapit *et al.* [19] conducted experiments on indoor positioning using BLE with a machine learning approach, as discussed in their work. In their study, the authors employed machine learning techniques such as support vector machines (SVM) and logistic regression (LR) to determine the position. Unlike previous studies, they partitioned the space into sub-areas known as radio maps. SVM and LR were then utilized to calculate the probability of predicting the radio map based on received signal strength indicator (RSSI) samples. The estimated position was subsequently calculated based on the generated probability. Despite the authors reporting a relatively low average error, it is worth noting that the environment and dataset used in their research were smaller compared to those in other studies like [17], [18].

Alexander *et al.* [20] use machine learning with regression tasks to estimate position . Several methods have been explored, including artificial neural network (ANN), multiple linear regression (MLR), RF, and support vector regression (SVR). The machine learning models are assigned to generating x position estimation ($\hat{x}$) and y position estimation ($\hat{y}$) directly based on preprocessed data (RSSI distance, x coordinate, and y coordinate). The best two machine learning models were then obtained, namely machine learning regression for x-coordinate position estimation and machine learning regression for y-coordinate position estimation. In the testbed with size 4 m x 6 m, Alexander reported the best model with a mean a mean error of 134.92 cm using SVR.

2.2. Bluetooth low energy fingerprinting

ILS using beacon fingerprinting have been implemented and evaluated in several papers. These systems may combine BLE beacons with techniques such as multilateration (MLT) and fingerprinting to determine the location of a mobile device indoors. Recent work highlights the potential of fingerprinting over the MLT with 79.31% accuracy [21]. BLE even may outperform global positioning system (GPS) when beacons' positions are known, the density is sufficient, and data is available for both calibration and training [7].

Faragher and Harle [7] reported that BLE readings in the same coordinate will differ over time. BE is a method based on probability theory that takes advantage on this principle. Probability theory can be implemented in ILS to predict tracked device (TD) coordinate from RSSI readings. In a 600 m^2 area with 19 beacons, accuracy reaches less than 3 m in 95% of the time. If 7 beacons is utilized instead, accuracy reaches less than 8.5 m in 95% of the time and less than 3.1 m in 66% of the time.

Subakti *et al.* [9] introduce fingerprint feature extraction (FPFE) method, which utilizes fingerprinting and provides two extraction choices: autoencoder (AE) or principal component analysis (PCA). In an 8 m $\times$ 5 m area with four beacons, FPFE with AE extraction attains the highest accuracy mean of 0.70 m, while with PCA extraction, the top accuracy mean in the same space is 0.68 m. It is important to note that these accuracy values are reported based on a single mobile phone. It is possible that the performance could be affected if there are variations, such as using a different mobile phone for fingerprint collection.

The aforementioned studies offer valuable insights into employing different methods with BLE technology. However, directly comparing their performance could be more challenging due to different environments, different test areas, and different performance metrics. Potortì *et al.* [4] this study aims to overcome this challenge by implementing prospective approaches, particularly those proposed by Faragher and Harle [7] and Subakti *et al.* [9], using a consistent setup and two standard scenarios. Finally, this study provides guidance on best practices for readers.

3. METHOD

3.1. Data collection

Given a set of beacons $B = \{b_1, \ldots, b_N\}$ installed in the testbed, we collect M RSSI during database collection phase in reference points (RPs) as shown in Figure 1, where $M = \{b_1^{RP}, \ldots, b_N^{RP}\}$. In the localization phase, we collect L RSSI in the same points using a TD, where $L = \{b_1^{TD}, \ldots, b_N^{TD}\}$.

We use three mobile phones as TDs from various brands and models, including Realme C20, Realme 5 Pro, and Samsung Galaxy A32. We develop an android application that reads transmitted BLE signal with interval 2,000 ms. In our testbed, we install N=6 BLE beacons transmitting signal power -4 dBm with advertising interval 1,500 ms. We collect 211 sampling data using three mobile phones in 42 RPs. Each RP is separated 1 m with other RPs. In total, we collect $26,586$ instances. These data are divided as fingerprint database (or training), M, and localization test, L. Figure 1 ilustrates the testbed in our living lab.

We consider two scenarios of device utilization during database collection and testing phase, namely:

1. Ideal scenario: fingerprinting and localization using whole known devices (i.e., M and L are collected using three devices). We sample 80% of the whole collected dataset (stratified random sampling, resulting $M = 21,268$ instances) for fingerprinting or training machine learning models, and use the rest of the data for localization ($L = 5,318$ instances). The aim of this scenario is to simulate a condition where the system knows all types of mobile phones in advance.

2. Real-world scenario: fingerprinting with 1 device and evaluation with other devices (i.e., M are collected using a device and L are collected using two other devices). We select one device as fingerprinting device and randomly sample 80% of the collected data (resulting $M = 7,089$ instances) as fingerprint or training data. We thus localize the TDs using the rest of dataset (resulting $L = 19,497$ instances). We repeat this process using the other two devices as fingerprint collectors. This scenario results three localization errors (each uses a mobile device as a fingerprint collector). The aim of this scene is to simulate typical / real-life scenario where the system does not know in advance all types of mobile phones.

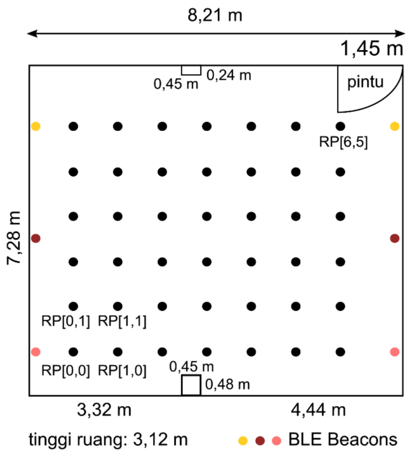


Figure 1. Floor plan of testing environment

3.2. Fingerprinting

Fingerprinting is a technique used to determine the location of an object or TDs within an indoor environment. The location is determined by comparing the signal characteristic collected by TD to the fingerprints stored in the database. Once a matching fingerprint is found, the system estimates the TD's location based on the known location associated with that fingerprint. In this research, we select a method based on probabilistic models to estimate the current location and another based on dimensionality reduction. A probabilistic model, such as a BE, is chosen due to its superior capability in handling uncertainty, particularly in dynamic environments with varying signal propagation. Conversely, a dimensionality reduction technique is employed to enhance generalization, filter out noise, and retain only the most informative features. The detailed discussion of these methods follows.

3.2.1. Bayesian estimator

BE is fingerprinting technique based on probabilistic model. We begin the process by doing normalization using min-max scaling. Min-max scaling scales and transforms numerical features of a dataset within a range, between 0 and 1. The process involves determining the minimum and maximum values of the feature and then scaling each data point proportionally, as shown in (1). This step benefits in processing and comparing data by machine learning algorithms.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{1}$$

Next, we calculate euclidean distance between mean RSSI captured by TD and mean RSSI captured in each RP using (2),

$$dist(B, L, M) = \sqrt{\sum_{i=1}^{N} \frac{(L(b_i) - M(b_i))^2}{N}} \tag{2}$$

where L is signals received during localization by a tracked device TD, $L = \{b_1^{TD}, \ldots, b_N^{TD}\}$, and M is reference RSSI collected during fingerprinting phase, $M = \{b_1^{RP}, \ldots, b_N^{RP}\}$.

Finally, we calculate likelihood based on the distance using Bayesian likelihood function as (3):

$$p = exp\left(-\frac{dist^2}{2\sigma^2}\right) \tag{3}$$

where σ is standard deviation that represents noises during fingerprinting. Prior distribution is assumed to be a constant as localization is on one shot, not tracking. Once Bayesian likelihood values are obtained, location of tracked device is estimated using maximum a posteriori (MAP).

3.2.2. Fingerprint feature extraction

FPFE aims to extract characteristics of beacon fingerprint using AE or PCA as proposed by Subakti *et al.* [9]. We begin the fingerprinting process by normalizing fingerprint data as in (1). Subsequently, FPFE extracts features as a dimensionality reduction process. Namely, the an initial set of data in high-dimensional space is projected to low dimension space without losing important information Fingerprint data for an RP are of the shape of 4 × 200. They are transformed to be of the shape of 1 × 800 Autoencoder for FPFE. AE is a type of ANN that encodes higher-dimension input features to be a lower-dimension internal representation called the code. An AE model consists of three parts: encoder, code, and decoder. The encoder compresses the input features to generate the code, and the decoder then reconstructs the input from the generated code. In this work, the BLE beacon node RSSI values from a RP are used as input features (1 × 800 dimension). They are encoded as a code (1 × 8 dimension), which in turn is decoded as output features (1 × 800 dimension). Figure 2 illustrates the architecture of the AE used in this work. The AE model is associated with the Adaptive moment estimation (Adam) optimizer [22] and the mean squared error (MSE) loss function. The number of epochs of training the AE model is 1,000.

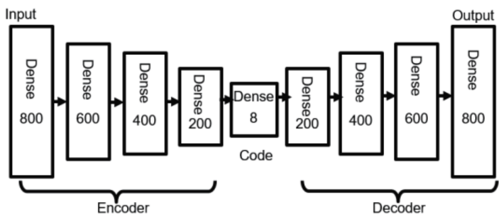


Figure 2. The AE structure adopted by the FPFE method [9]

Minkowski distance is then used as the fingerprint similarity measurement to select k RP candidates with the smallest distances. Minkowski distance can be calculated with as (4) and (5):

$$D(x, y) = (\sum_{i=1}^{d} |x_i - y_i|^p)^{1/p} \tag{4}$$

$$D(L, M) = (\sum_{i=1}^{N} |L(b_i) - M(b_i)|^p)^{1/p} \tag{5}$$

where p is order of Minkowski distance ($p = 2$ is Euclidean distance).

In the FPFE methods using AE feature extraction, the p value is 8, because the feature extraction output of the AE is 8 features. By calculating the Minkowski distance between features of the TD and all RPs, k RPs with k smallest Minkowski distances are selected. They are called RP candidates whose positions are used to estimate the TD's position

The tracked device location is finally calculated by averaging coordinates of the k selected RP candidates. The TD's position (x, y) is calculated simply as the centroid of the k RP candidates as in (6) where (x_i, y_i) is the coordinate of selected RP_i with the k smallest distance.

$$(x, y) = \frac{1}{k}\sum_{i=1}^{k}(x_i, y_i) \tag{6}$$

3.3. Machine learning models

As an alternative to fingerprinting, we investigate machine learning methods for TD localization. As the localization outcome is greatly influenced by preprocessing, we meticulously preprocess the input data through multiple phases outlined below. We thus utilize machine learning techniques specified for regression tasks that involve distance calculation and some techniques based on ensemble learning.

- Conversion signal strength to miliwatt. The received signal strength is quantified in decibels milliwatt (dBm). dBm is a unit of measurement in a logarithmic unit that compares the ratio of P, the power being measured in mW, with the reference power of one milliwatt ($1mW$) as shown in (7).

$$dBm = 10 \cdot log_{10}\left(\frac{P_{(mW)}}{1mW}\right) \tag{7}$$

In approaches based on machine learning, we convert dBm to linear value in $P_{(mW)}$ as shown in (8) to mitigate the issue of machine learning models that are sensitive to the scale of features.

$$P_{(mW)} = 1_{(mW)} \cdot 10^{\left(\frac{P_{(dBm)}}{10}\right)} \tag{8}$$

- Min-max scaling. We transform the signal strength within the range $[0, 1]$ by subtracting the minimum value of the RSSI and dividing by the range between the maximum and minimum RSSI. This normalization aids machine learning algorithms that rely on distance calculations, promoting a more equitable influence of each feature in the subsequent analyses.
- Splitting train test data. A significant portion of the data, 80% is allocated as the training set, while reserving the remaining portion for testing. This approach helps prevent overfitting, where the model becomes too tailored to the training data and fails to generalize effectively.

3.3.1. Support vector regressor

This technique is an extension of the SVM algorithm for regression problems. SVR works by finding a hyperplane that best fits the data within a defined tube (margin) while allowing for a certain level of error [23]. It leverages the kernel trick to map the data into a higher-dimensional space, making it possible to capture complex relationships in the data. We use polynomial kernel with degree of 3, $\epsilon = 0.1$, and regularization parameter $C = 100$.

3.3.2. K-neighbors regressor

K-neighbors regressor is an instance-based learning algorithm, also known as lazy learning. K-nearest neighbors (k-NN) algorithms make predictions based on the similarity of instances in the training data. We use euclidean distance. For regression tasks, the predicted value for the target data point is often the mean (or median) of the target values of its k nearest neighbors. In this work we use $k = 3$.

3.3.3. Random forest regressor

This technique is an ensemble learning algorithms that operates by constructing many decision trees during training and outputs mean prediction of the individual trees [24]. In our work, the number of trees is 100. each tree is built using a different bootstrap sample drawn from the original dataset during training process. The sample is randomly picked up with replacement while the number of samples to draw is the same with the number of dataset. For each bootstrap sample and at each node of each decision tree, the algorithm selects the best split among the randomly chosen subset of features. To measure the quality of a split, we use MSE. The tree continues to split until a stopping criterion is met.

3.3.4. Gradient boosting

Gradient boosting (GBoost) works by sequentially adding weak learners (i.e., regression trees), each focusing on correcting the errors of the previous model [25]. In this study, we use 100 estimators with a maximum depth of 15. It starts by training the first tree on the data and updating weights based on prediction errors. We use MSE function to calculate errors during training. Subsequent trees are trained to focus on the previously misclassified examples with learning rate 0.1. The final prediction is made by aggregating the predictions of all the trees. This approach gradually learns complex patterns by combining multiple simple trees.

3.4. Performance measures

We calculate the localization error by computing the euclidean distance between the predicted position (x', y') and the known position of RPs (x, y). We then calculate the mean and standard deviation of the error. We also provide cumulative distribution function (CDF) 95%. Finally, we generate and plot a CDF for a normal distribution based on the mean and standard deviation calculation for comparing the performance of the approaches.

4. RESULTS AND DISCUSSION

4.1. Ideal scenario

In an ideal condition, each tracked device should have been used for data collection in building a fingerprint database. In other words, the fingerprinting process need to be repeated whenever a new tracked device will be used in the localization. While this is ideal, the repeated process of collecting fingerprint requires much effort. This scenario accommodates such condition; we tune models based on some part of collected data and test with the other part of the data. In this scenario, we use $7,089$ instances from each mobile phone (i.e., 80% of collected dataset) for fingerprinting or training machine learning models ($M = 21,268$ from 3 phones). We test on $1,773$ instances collected from each mobile phone ($L = 5,318$ from 3 phones).

When fingerprinting data source is from 3 mobile phones, best results achieved using FPFE, reaching average error of 0.50 m, as shown in Table 1. Using BE, the mean error may reach average error of 1.636 m. There are several possible explanations for this result. First, FPFE has sucessfully extracted features from fingerprints collected by three different mobile phones. This makes the localization using FPFE resulting lower error. Second, BE struggles to generalize the collected data, particularly when constructing a single BE model from a dataset gathered using three different devices. The challenges lie in estimating parameters like the prior distribution and likelihood function, which characterize the data distribution. This limitation results in suboptimal outcomes and imprecise inferences.

As shown in Table 1, the worst localization is resulted using SVR with mean localization error of 2.057 m. It is probable the polynomial kernel with degree of 3 does not learn the training data and generalize toward testing data. The other machine learning techniques tend to perform better than SVR on localization. Ensemble models, i.e., RF and GBoost, result mean localization error of 0.847 m - 0.909 m, while K-neigbors regressor provides slightly lower mean error of 0.829 m but higher CDF 95%.

Among the considered machine learning methods, RF is more likely to perform better in the test, as 95% probability of localization error is below 2.488 m. RF slightly outperforms GBoost with 95% of localization error less than 2.637 m. RF build multiple decision trees in parallel that learn the same data. The best features are selected from these trees and will be combined and combined to exhibit predictions. Compared to another ensemble learning, i.e., GBoost, trees are built sequentially. The most recent tree model will be used to predict the output. In scenarios like localization, where signal data is subject to fluctuations from factors like signal fading and device diversity, RF slightly outperforms GBoost, primarily owing to its utilization of the most effective features rather than relying solely on the latest model.

Table 1. Localization error in ideal scenario (i.e., 3 tracked devices, each has been fingerprinted) in m^2

	Mean	Std. Dev.	CDF 95% (m)
BE [7]	1.636	1.437	4.000
FPFE [9]	0.502	0.815	1.844
SVR	2.057	0.988	3.683
K-neigbors regressor	0.829	1.220	2.836
RF	0.847	0.997	2.488
GBoost	0.909	1.051	2.637

4.2. Real-world scenario

In a real-world scenario, we operate under the assumption that gathering fingerprint data from every mobile phone is unfeasible. Therefore, we can rely only on fingerprint dataset collected from a single mobile phone. We use M=7,089 instances collected from one mobile phone as fingerprint data, and test on L=19,497 instances (from three mobile phones, i.e., 1,773; 8,862; and 8,862 instances, respectively).

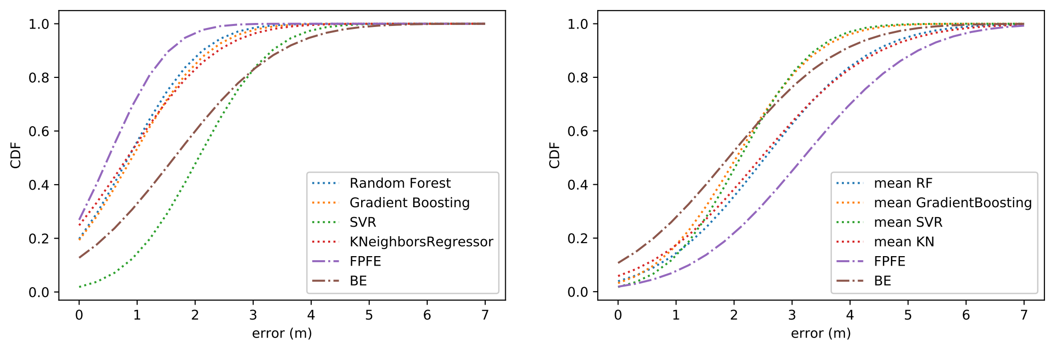
Table 2 shows the optimal outcomes obtained with BE, with a mean error of 1,765 m depending on the device utilized as the fingerprint collector. It is probable that BE can assume a distribution from signals collected by one mobile phone collector. This distribution appears to resemble the dataset distribution from other mobile phones. Conversely, FPFE, which excelled in the earlier scenario (i.e., Ideal scenario), now yields the poorest localization, with a mean error of 3.2 m. This outcome could be attributed to the fact that FE extracts features exclusively from a particular phone. The features extracted are not applicable to datasets from other mobile phones.

Examining machine learning techniques, SVR demonstrates relatively superior performance compared to ensemble methods and K-neigbors regressor, as shown in Table 2. A plausible reason is that SVR operates on hyperplanes that effectively fit the data. When trained with RSSI data exclusively collected from one phone, SVR can learn its patterns and apply them to new datasets collected from other mobile phones.

Table 2. Localization error in real world scenario (i.e., 3 tracked devices, 1 fingerprint collector) in m^2

		Fingerprint collector		
		Phone1	Phone2	Phone3
BE [7]	Mean err.	1.965	1.785	1.956
	Std. Dev.	1.557	1.582	1.450
	CDF 95%	4.526	4.386	4.341
FPFE [9]	Mean err.	3.048	3.257	3.283
	Std. Dev.	1.440	1.584	1.582
	CDF 95%	5.417	5.863	5.885
SVR	Mean err.	2.118	1.965	2.239
	Std. Dev.	1.018	0.982	0.982
	CDF 95%	3.792	3.579	3.855
K-neigbors regressor	Mean err.	2.711	2.439	2.294
	Std. Dev.	1.687	1.780	1.275
	CDF 95%	5.485	5.366	4.391
RF	Mean err.	2.527	2.787	2.335
	Std. Dev.	1.492	1.577	1.270
	CDF 95%	4.981	5.381	4.425
GBoost	Mean err.	2.383	1.907	2.007
	Std. Dev.	1.262	1.294	1.238
	CDF 95%	4.458	4.035	4.044

To better compare localization performance, we have summarized and plotted the cumulative distribution function in Figure 3. What is particularly noticeable in this figure is the pattern of localization accuracy across different scenarios. Notably, the localization accuracy in an ideal scenario in Figure 3(a) is generally superior to that in a real-world scenario in Figure 3(b), as indicated by a lower CDF of localization error. This observation suggests that localization performance is significantly influenced by the availability and quality of the training data or fingerprint database.



(a) (b)

Figure 3. CDF of localization error, (a) ideal scenario and (b) real world scenario

5. CONCLUSION

One common challenge encountered in the development of ILS is ensuring accuracy and compatibility across a range of devices. To address this issue, researchers have explored multiple methodologies to enhance localization performance while accommodating device diversity. However, these studies often employ disparate datasets and setups to construct their individual proof of concepts. In our investigation, we address this gap by evaluating several promising methodologies within a unified test environment. Specifically, we compare two fingerprinting algorithms: one based on feature extraction (referred to as FPFE) and another based on probability theory (referred to as BE). Furthermore, we assess various machine learning approaches, including SVR, which utilizes geometric boundaries (hyperplanes), ensemble learning, and instance-based learning. The effectiveness of these methods depends on factors like the availability of training data or the fingerprint database, as explored through the Ideal scenario and Real-world scenario. Our findings indicate that outcomes can vary based on the specific conditions. Therefore, in this study, we propose methods suitable for particular circumstances.

In the context of three fingerprint collectors (Ideal scenario), the system administrator has the option to gather fingerprint data through users' devices. When such a scenario occurs, wherein the algorithm is provided with all possible data variations, it is advisable to employ FPFE. FPFE demonstrates superior performance compared to alternative algorithms, achieving a mean error of 0.50 m. This indicates that FPFE has the capability to effectively extract features from fingerprint data. While FPFE excels in fingerprinting, the performance of fingerprinting with BE is subpar. BE struggles to achieve generality in estimating common parameters from fingerprint data collected across diverse devices. FPFE, based on AE as outlined in this study, demands substantial resources for feature extraction. In situations where computational resources are limited, opting for machine learning techniques becomes a viable alternative. Machine learning-based approaches provide adaptability and scalability by harnessing algorithmic power to deduce locations. We observe that ensemble learning, exemplified by RF and GB, outperforms other machine learning techniques. Specifically, RF benefits from random feature selection across multiple trees.

In the case of one fingerprint collector (Real-world scenario), there is limitation in collecting data using other devices. In other words, system admin may collect fingerprint data for localization using one or a few device (but not all devices). When the fingerprint data is available, BE and SVR lead the performance, according to our experiment. This indicates that BE may set parameters that describe the distribution of data collected using a mobile phone. Such model is then applicable the other users' devices that have the same characteristic. Machine learning techniques may also be an option in localization while there is a limitation in a new fingerprint collection. When only a batch data from a single data source (i.e., single mobile phone) with a specific characteristic is available, SVR might be chosen. SVR performs better due to its ability to set hyperplane using the data from one source.

In our future work, we intend to assess the asymmetric distribution power transmission in each beacon. A symmetric configuration might enhance performance as a measure to counteract the multipath fading effect. Furthermore, there is potential to create a single BE model for each fingerprinting collector. These individual BE models could then be employed and collectively contribute to the final decision. Contrary to this approach, in our current study, a single generic BE model was constructed using fingerprint data from three distinct devices.

ACKNOWLEDGEMENT

The authors acknowledged the funding support from Doctoral Competency Improvement Program Universitas Gadjah Mada Number 7743/UN1.P.II/Dit-Lit/PT.01.03/2023. The authors thank to M. Nauval Rafli and Rasyid Aulia Alba for their assistance in the data collection process.

REFERENCES

[1] F. Dwiyasa and M. H. Lim, "A survey of problems and approaches in wireless-based indoor positioning," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, Oct. 2016, pp. 1–7, doi: 10.1109/IPIN.2016.7743591.

[2] E. Jimenez and R. Wei, "Indoor localization of ubiquitous heterogeneous devices," in *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*, Jun. 2013, pp. 698–703, doi: 10.1109/CSCWD.2013.6581045.

[3] A. A. Khudhair, S. Q. Jabbar, M. Q. Sulttan, and D. Wang, "Wireless indoor localization systems and techniques: survey and comparative study," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 3, no. 2, pp. 392–409, Aug. 2016, doi: 10.11591/ijeecs.v3.i2.pp392-409.
[4] F. Potortì *et al.*, "Comparing the performance of indoor localization systems through the EvAAL framework," *Sensors*, vol. 17, no. 10, p. 2327, Oct. 2017, doi: 10.3390/s17102327.
[5] Y. Ibnatta, M. Khaldoun, and M. Sadik, "Exposure and evaluation of different indoor localization systems," in *Lecture Notes in Networks and Systems*, vol. 216, 2022, pp. 731–742.
[6] Y. Zhuang, J. Yang, Y. Li, L. Qi, and N. El-Sheimy, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, no. 5, p. 596, Apr. 2016, doi: 10.3390/s16050596.
[7] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.
[8] L. Ferrer, "Analysis and comparison of classification metrics," *arxiv*, 2022, [Online]. Available: http://arxiv.org/abs/2209.05355.
[9] H. Subakti, H.-S. Liang, and J.-R. Jiang, "Indoor localization with fingerprint feature extraction," in *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)*, Oct. 2020, pp. 239–242, doi: 10.1109/ECICE50847.2020.9301994.
[10] T.-M. T. Dinh, N.-S. Duong, and K. Sandrasegaran, "Smartphone-based indoor positioning using BLE iBeacon and reliable lightweight fingerprint map," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10283–10294, Sep. 2020, doi: 10.1109/JSEN.2020.2989411.
[11] M. Li, L. Zhao, D. Tan, and X. Tong, "BLE fingerprint indoor localization algorithm based on eight-neighborhood template matching," *Sensors (Switzerland)*, vol. 19, no. 22, 2019, doi: 10.3390/s19224859.
[12] S. Subedi, H. S. Gang, N. Y. Ko, S. S. Hwang, and J. Y. Pyun, "Improving indoor fingerprinting positioning with affinity propagation clustering and weighted centroid fingerprint," *IEEE Access*, vol. 7, pp. 31738–31750, 2019, doi: 10.1109/ACCESS.2019.2902564.
[13] P. Martins, M. Abbasi, F. Sa, J. Celiclio, F. Morgado, and F. Caldeira, "Intelligent beacon location and fingerprinting," *Procedia Computer Science*, vol. 151, pp. 9–16, 2019, doi: 10.1016/j.procs.2019.04.005.
[14] Y. Zhuang, C. Zhang, J. Huai, Y. Li, L. Chen, and R. Chen, "Bluetooth localization technology: principles, applications, and future trends," *IEEE Internet of Things Journal*, vol. 9, no. 23, pp. 23506–23524, Dec. 2022, doi: 10.1109/JIOT.2022.3203414.
[15] A. R. Pratama, A. Lazovik, and M. Aiello, "Office multi-occupancy detection using BLE beacons and power meters," in *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, Oct. 2019, pp. 0440–0448, doi: 10.1109/UEMCON47517.2019.8993008.
[16] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214945–214965, 2020, doi: 10.1109/ACCESS.2020.3039271.
[17] L. Bai, F. Ciravegna, R. Bond, and M. Mulvenna, "A low cost indoor positioning system using bluetooth low energy," *IEEE Access*, vol. 8, pp. 136858–136871, 2020, doi: 10.1109/ACCESS.2020.3012342.
[18] M. W. P. Maduranga and R. Abeysekera, "Bluetooth low energy (BLE) and feed forward neural network (FFNN) based indoor positioning for location-based IoT applications," *International Journal of Wireless and Microwave Technologies*, vol. 12, no. 2, pp. 33–39, Apr. 2022, doi: 10.5815/ijwmt.2022.02.03.
[19] P. Sthapit, H.-S. Gang, and J.-Y. Pyun, "Bluetooth based indoor positioning using machine learning algorithms," in *2018 IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia)*, Jun. 2018, pp. 206–212, doi: 10.1109/ICCE-ASIA.2018.8552138.
[20] I. Alexander and G. P. Kusuma, "Predicting indoor position using bluetooth low energy and machine learning," *International Journal of Scientific and Technology Research*, vol. 8, no. 9, pp. 1661–1667, 2019.
[21] A. H. Elhusseiny, M. Zamzam, and Y. Zaghloul, "Precision localization: an experimental study on BLE fingerprinting and trilateration with ESP32," in *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, Apr. 2023, pp. 60–65, doi: 10.1109/ICAECIS58353.2023.10170249.
[22] J. L. Ba and D. P. Kingma, "Adam: a method for stochastic optimization," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.
[23] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, Apr. 2011, doi: 10.1145/1961189.1961199.
[24] L. Breiman, "Random forest," *Machine learning*, vol. 45, pp. 5–32, 2001, doi: 10.1023/a:1010933404324.
[25] T. Hastie, R. Tibshirani, and and J. H. F. J. H. Friedman, "The elements of statistical learning: data mining, inference and prediction," *Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005, doi: 10.1007/BF02985802.

BIOGRAPHIES OF AUTHORS

Azkario Rizky Pratama completed the Ph.D. degree in Computer Science from the University of Groningen, The Netherlands, in 2020. He serves as an Assistant Professor in the Department of Electrical and Information Engineering at Universitas Gadjah Mada, Indonesia. His main research interests include Ambient intelligence, context-awareness, and mobile computing. He can be contacted at email: azkario@ugm.ac.id.

Muhammad Evan Anindya Wahyuaji has graduated from bachelor degree at the Department of Electrical and Information Engineering, Universitas Gadjah Mada, Indonesia, in 2022. Currently, he is working as a back-end software developer. He can be contacted at email: evananindya@mail.ugm.ac.id.

Muhammad Fadhil Nur Hidayat received the B.S. degree in Electronics and Instrumentation from the Indonesian Polytechnic of Nuclear Technology, Yogyakarta, in 2022. He is currently a post-graduate student in M.Eng. program at the Department of Electrical Engineering and Information Technology, Gadjah Mada University, Yogyakarta, Indonesia. His research interests include smart systems, indoor positioning, UWB-based data communication, and localization technologies. He can be contacted at email: mfadhilnurhidayat@mail.ugm.ac.id.

Bimo Sunarfri Hantono received a B.Eng. and Dr. from Universitas Gadjah Mada, Indonesia and a M.Eng. from Nanyang Technological University, Singapore. He has worked for Department of Electrical and Information Engineering at Universitas Gadjah Mada since 2002. His research interest includes human computer interaction, mobile sensing, and web-based application. He can be contacted at email: bhe@ugm.ac.id.

Nur Abdillah Siddiq received the Dr. in Physics and B.E. degree in Engineering Physics from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, with PMDSU scholarship from Ministry of Research, Technology and Higher Education Indonesia. Since 2019, he is a full-time lecturer in Engineering Physics Study Program at Universitas Gadjah Mada. His research interests include smart windows, optical sensors, optical waveguide, and plasmonics. He can be contacted at email: nur.a.s@ugm.ac.id.