Pamukkale Univ Muh Bilim Derg, 29(1), 36-43, 2024

Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi

Pamukkale University Journal of Engineering Sciences

A novel hybrid model for bluetooth low energy-based indoor localization using machine learning in the internet of things

Nesnelerin internetinde iç mekân lokalizasyonu için makine öğrenimini kullanan bluetooth düşük enerji tabanlı özgün hibrit model

Yasin GÖRMEZ[1*], *Halil ARSLAN*[2], *Yunus Emre IŞIK*[1], *Sercan TOMAÇ*[3]

[1]Management Information Systems, Faculty of Economics and Administrative Sciences, Sivas Cumhuriyet University, Sivas, Turkey.
yasingormez@cumhuriyet.edu.tr, yeisik@cumhuriyet.edu.tr

[2] Computer Engineering, Faculty of Engineering, Sivas Cumhuriyet University, Sivas, Turkey.
harslan@cumhuriyet.edu.tr

[3] Detay Consultancy, Computer Services Industry and Foreign Trade Joint Stock Company, İstanbul, Turkey.
sercan.tomac@detaysoft.com

Received/Geliş Tarihi: 02.12.2022 Revision/Düzeltme Tarihi: 25.02.2023 doi: 10.5505/pajes.2023.57088
Accepted/Kabul Tarihi: 08.03.2023 Research Article/Araştırma Makalesi

Abstract

Indoor localization involves pinpointing the location of an object in an interior space and has several applications, including navigation, asset tracking, and shift management. However, this technology has not yet been perfected, and many methods, such as triangulation, Kalman filters, and machine learning models have been proposed to address indoor localization problems. Unfortunately, these methods still have a large degree of error that makes them ill-suited for difficult cases in real-time. In this study, we propose a hybrid model for Bluetooth low energy-based indoor localization. In this model, the triangulation method is combined with several machine learning methods (naïve Bayes, k-nearest neighbor, logistic regression, support vector machines, and artificial neural networks) that are optimized and tested in three different environments. In the experiment, the proposed model performed similarly to the solo triangulation model in easy and medium cases; however, the proposed model obtained a much smaller degree of error for hard cases than either solo triangulation or machine learning models alone.

Keywords: Internet of Things, Indoor localization, Bluetooth low energy, Machine learning, Triangulation.

Öz

İç mekân konumlandırma, bir nesnenin iç mekândaki konumunun tam olarak belirlenmesi olarak tanımlanabilir ve navigasyon, varlık takibi ve vardiya yönetimi olmak üzere bir çok uygulama alanı bulunmaktadır. İç mekân konumlandırma problemlerini çözmek için üçgenleme, Kalman filtreleri ve makine öğrenmesi modelleri gibi birçok yöntem önerilmiştir ancak hala istenilen başarı oranları elde edilememiştir. Bu yöntemler deney ortamlarında başarılı sonuçlar elde etse de, gerçek zamanlı durumlarda hata oranları çok fazla olabilmektedir. Bu çalışmada, Bluetooth düşük enerji tabanlı iç mekân konumlandırma için hibrit bir model önerilmiştir. Bu modelde, üçgenleme yöntemini, üç farklı ortamda optimize edilmiş ve test edilmiş birkaç makine öğrenmesi yöntemiyle (Naive Bayes, k-en yakın komşu, lojistik regresyon, destek vektör makineleri ve yapay sinir ağları) birleştiren hibrit bir yaklaşım kullanılmıştır. Çalışmada önerilen model, kolay ve orta durumlarda üçgenleme modeline benzer şekilde performans göstermiş; ancak önerilen model, zor durumlar için üçgenleme veya tek başına makine öğrenimi modellerinden çok daha küçük bir hata oranı elde etmiştir.

Anahtar kelimeler: Nesnelerin interneti, İç mekân konumlandırma, Buluetooth düşük enerji, Makine öğrenmesi, Üçgenleme.

1 Introduction

Tracking people, objects, or animals is essential for many situations, including navigation, shift management, and asset management. An accurate estimation of an object's outdoor position can be achieved using a Global Positioning System (GPS), however, indoor localization remains challenging. Interior spaces have many factors that can affect signal quality such as metal density, crowds of people and thick walls. Because smartwatches, smartphones and other Bluetooth devices became ubiquitous, improving indoor localization using Bluetooth Low Energy (BLE) became one of the most popular challenges in the Internet of Things (IoT) field.

Indoor Localization (IL) can be defined as the determination of the location of people or objects in the interior where satellite systems such as GPS are insufficient. IL has many benefits such as better management of resources, real-time information acquisition and marketing. In today's technology, IL applications or research and development studies have been done frequently to solve many problems. Calderoni et al. focused on the field of health and they developed an IL system for human resource management, patient location detection and to find lost property in a hospital in Italy [1]. Álvarez-Díaz and Caballero-Gil argued about use case of IL for staff management and they concluded that, thanks to IL, staff performance can be known, daily activities can be followed and staff planning can be done more accurately. In addition to this, they showed that staff tracking with IL can be applied in many areas such as sports fields, universities and shopping centers [2]. IL have a great importance for smart airports. It is seen that IL is used for various purposes at the most important airports in the world [3]. Considering these applications and studies, it is seen that making location estimation with less errors will be beneficial for many business processes.

*Corresponding author/Yazışılan Yazar

To date, many systems have been developed for indoor localization. For instance, Jianyong et al. proposed indoor localization methods based on received signal strength indication (RSSI) and Bluetooth methods, which use Gaussian filters to preprocess, a Taylor series for de-noising, and active learning; they obtained an 80% probability of locating an object with an error of less than 1.5 m [4]. Meanwhile, Mussina et al. developed an RSSI-based indoor localization algorithm using mathematical filtering functions such as median, mode, single direction outlier removal, shifting and feedback filtering and a trilateration algorithm with a goal of achieving accuracy within 2m [5]. Yoon et al. increased the accuracy using a Kalman Filter (KF), a Rauch-Tung-Striebel smoother and a trilateration algorithm [6]. Xu et al. showed an optimal fingerprint length effect on localization accuracy and real time performance using long short-term memory (LSTM) [7]. Dinh et al. proposed a hybrid method that consists of a regression model, line intersection-based trilateration, and a median filter; they obtained an average error of 0.8 m, and 70% of the errors were less than 1 m [8]. Iqbal et al. proposed convolutional neural networks and artificial neural networks based on BLE and obtained 99.9% accuracy [9]. Giuliano et al. proposed an indoor localization system using feedforward neural networks and were accurate with an error of less than 1m [10]. Peng et al. developed an indoor localization system using an iterative weighted k-nearest neighbors algorithm and decreased the error from 2.7 m to 1.5 m [11]. Teran et al. implemented a k-nearest neighbor algorithm and k means clustering for zone prediction and obtained 70.2% accuracy [12]. Boronti et al. produced a new dataset for indoor localization problems to be added to the literature [13]. Sadowski and Spachos compared four indoor localization technologies (Wi-Fi, BLE, LoRaWAN, and Zigbee) and found that Wi-Fi is the most accurate, BLE uses the least power, and LoRaWAN has the greatest transmission range [14]. Hou and Arslan proposed Monte Carlo localization algorithms for indoor positioning and demonstrated that their proposed system does not demand the high deployment density of BLE beacons that is required of triangulation and trilateration-based indoor positioning algorithms [15]. Röbesaat et al. proposed a novel positioning method using Kalman filtering combined with dead reckoning-based fixes and trilateration-based fixes and successfully increased the accuracy [16]. Mackey and Spachos compared three of the most popular beacons for indoor localization and demonstrated that accuracy is affected by the environment and that filtering is necessary to improve the beacons' performance [17]. Ji et al. compared how beacon positioning affects the accuracy of BLE signals and Wi-Fi signals, finding that many more BLE beacons are needed than Wi-Fi devices to obtain the same level of accuracy and that there is a strong relationship between accuracy and the position of the beacons [18]. Qureshi et al. showed that the multiple transmission power level negatively impacted the accuracy of indoor localization systems, increasing the error from 2 m to 5 m [19]. Qureshi et al. also found that, although Wi-Fi signals are more stable than BLE signals, BLE is sufficiently accurate for indoor localization. In addition, because BLE operates at much lower transmission power levels, it consumes less energy [19]. Kayış et al. mentioned that there is no general solution or technology for indoor localization problems and they proposed web based modelling system and A* based indoor navigation system [20]. As a result of the literature review, it is concluded that the performance of BLE-based IL systems varies a lot according to the environment. Various preprocessing steps were also used in IL systems to minimize possible errors. However, in these studies, it was ignored that sometimes the traditional triangulation method and sometimes the signature based methods are better than the other, depending on the environment. Therefore, in our study, a hybrid approach was proposed to solve this problem.

An indoor localization system generally consists of stationary gateways and a mobile Bluetooth transmitter. The distance from the mobile transmitter to the gateways is calculated using RSSI, and the location of the mobile transmitter is calculated using the distances between three gateways in a triangulation algorithm. The real distance of the mobile transmitter from a gateway is crucial for accurate distance calculation. For example, the RSSI measures -82 if the real distance is between 12 and 14 meters and measures -92 if the real distance is between 39 and 45 meters (measured power is assumed to be -65 and the free space factor is assumed to be 2). Therefore, if the target is far away from a gateway, the accuracy of the system decreases because of the characteristics of BLE signals. Considering the literature, even if the demo systems achieve an error of one meter, achieving this degree of accuracy in the real system is costly. To achieve accuracy within one meter, the distance between the gateways should be four meters at most. Thus, an area of 64 square meters (8m × 8m) needs at least 9 gateways. These environments are expensive, and real fields are often much larger than 64 square meters. Another problem in the systems with real-time data is too much noise. BLE uses a low transmission level to preserve battery life and is therefore easily affected by the environment. Another approach for indoor localization is using footprint and machine learning algorithms to estimate position. For this purpose, the area is divided into regions, and data from each region are collected separately. If the regions are large, then the accuracy of the system is worse than when using triangulation, and, if the regions are small, it is difficult to collect data because the number of regions will be extremely high. In these situations, triangulation sometimes performs better than footprinting but not always. Therefore, we propose a hybrid model, which is the first novelty of this paper, that uses footprinting and triangulation together. In this model, signals from transmitters are checked with a formula, and, if at least three of them intersect, the position will be calculated by triangulation. Otherwise, the machine learning model will estimate the position. Second novelty of this study is evaluated as a generating real time dataset for indoor localization problems

2 Methods

2.1 Triangulation

Triangulation is a method of finding the location of a transmitter using radial distance. In this method, the distance of a transmitter from the gateways in the gravitational field is measured using signal power. At least three of the measured distances are selected and at least three circles with radius of the measured distance are generated. The intersection of these circles gives the position of the transmitter. The area of the intersection point measures the accuracy of the positioning system. Several approaches can be used for a triangulation algorithm, and detailed information about the approach used in this study can be found in papers by Hou et al. [21].

2.2 Fingerprinting

Fingerprint (FP) approach is another method frequently used in location detection systems. FP approach aims to create the

RSSI fingerprint of the region by measuring the RSSI values in a particular region. FP approach basically consists of two main stages: offline and online. In the offline stage, RSSI values from determined regions are collected and stored. For this purpose, it is necessary to collect data by physically walking in the region. At this stage, the information of the region where the RSSI values are collected should also be stored. In the online stage, the data collected in the first stage is used to estimate the real-time location. The detailed information about FP approach can be found in the study proposed by Yu et al. [22]. In our study, dataset collection part for machine learning model can be thought as an offline stage of FP and the machine learning part can be thought as and online stage of FP.

2.3 Classification methods

2.3.1 K-Nearest neighbor

The k-nearest neighbor (k-nn) algorithm is a supervised machine learning method that classifies based on the data stored in the training set. In many of the supervised classification algorithms, some parameters are determined by pre-training processes that use the training set, and the test data are classified using these parameters without the need for training data. In the k-nn algorithm, there is no need for pre-training. Instead, the test data are classified using the training set each time the algorithm is run. Therefore, in the first step of the k-nn algorithm, a training set is created with the help of labeled data. Then the k parameter and a distance function (Minkowski, Euclid, etc.) are selected. When new data is encountered, the distance of this data to the data in the training set is calculated one by one using the selected distance algorithm. Then, the classification set is created by choosing k data with the smallest distance from the training sets. In the final step, the class of the new data is determined by the majority class in the classification set, and the model is terminated. The detailed information about k-nn algorithm can be found in the document written by Petersen [23].

2.3.2 Naïve bayes

The Naïve Bayes algorithm is one of the simplest machine learning algorithms and aims to find the probability of samples belonging to each class based on the Bayes theorem shown in Equation 1.

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \quad (1)$$

In this equation, X represents the feature vector, C represents the class label, and P (C|X) represents the probability of sample X belonging to C. In this equation, P (X) can be eliminated for the naïve Bayes algorithm because it is the same for all classes and thus will not affect the result. In the final step of the naïve Bayes algorithm, the probability of sample X, which belongs to each class, is calculated using Eq. 1 with the sample assumed to belong to the class with the greatest probability value. The detailed information about Naïve Bayes classifier can be found in the Murphy's study [24].

2.3.3 Logistic regression

Logistic regression (LR) is a data analysis technique that uses mathematics to find relationships between two data factors. LR then uses this relationship to estimate the value of one of these factors based on the other. The prediction usually has a limited number of outcomes such as yes or no. LR models are mathematically less complex than other machine learning methods [25]. LR models can process large volumes of data at high speed as they require less computational capacity such as memory and processing power. In most cases, more than one explanatory variable affects the value of the dependent variable. LR formulas assume a linear relationship between different independent variables to model such sets of input data. The function of LR for such sets is shown in Equation 2.

$$y = f(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n) \quad (2)$$

In this equation, f represents the logit function, x_n represents the observed value for the n^{th} variable, and β_n represents the regression coefficient for n^{th} variable. The detailed information about LR can be found at book chapter written by Wright [26].

2.3.4 Artificial neural networks

Artificial Neural Networks (ANN) is a supervised machine learning method that aims to learn by imitating the human nervous system. An ANN model is created by connecting the neurons in one layer to the neurons in the following layer. The multilayer perceptron neural network (MLP) model, which is the most commonly used artificial neural network model, consists of three layers: an input layer, a hidden layer, and an output layer. The input layer is where data is read. Since each neuron represents a feature, the model contains as many neurons as the number of features. The output layer is where classes are determined. This layer can contain a single neuron or as many neurons as the number of class varieties, depending on the model. Finally, the hidden layer, situated between the input layer and the output layer, is where data is subjected to intermediate processing. Although there is no standard number of hidden layers or of neurons within a hidden layer, these two factors greatly affect the accuracy of training. The MLP model is also known as feedforward ANN because the learning is done from one layer to the next layer. These training algorithms aim to update the weights to minimize the error. The detailed information about ANN classifier can be found in the study written by Jain et al. [27].

2.3.5 Support vector machines

The support vector machine (SVM) is capable of separating data into two or more classes with separation mechanisms in linear form in two-dimensional space, planar in three-dimensional space, and hyperplane in multi-dimensional space. The SVM moves data to a higher dimensional space through kernel functions to make them linearly separable, assuming that the data consisting of N elements to be used for training is $Q = \{x_i, y_i\}$ where $i = 1,2, \ldots N$, x_i indicates the feature vector and y_i indicates the class values. Although an infinite number of multiple planes can be drawn that can classify the dataset, the goal is to select the hyperplane that will result in the smallest unknown classification error. The detailed information about SVM classifier can be found in the study written Noble [28].

3 Proposed model

In this study, hybrid model that uses both triangulation and machine learning algorithms was proposed. This system consists of six layers: data collection, a broker, pre-processing, distance calculation, filtering, and decision-making. The system architecture is shown in Figure 1.

The RSSI value of each beacon is measured by gateways in the data collection layer and is then transferred to the broker. The broker is an interface that takes data from sensors with several transfer protocols, such as TCP, UDP, HTTP, and MQTT, and

decomposes the data to extract information. This broker is horizontally scalable and based on a Software as a Service (SaaS) system. It was developed using a microservice architecture, allowing new features that are programming language-independent to be added easily. In the literature, it is shown that using full-text index search algorithm with the document based database improves the performance of the search system [29]. Because of this reason, our broker also contains advanced system such as elastic search and MongoDB. Detailed information about this broker can be found in our previous work [30].

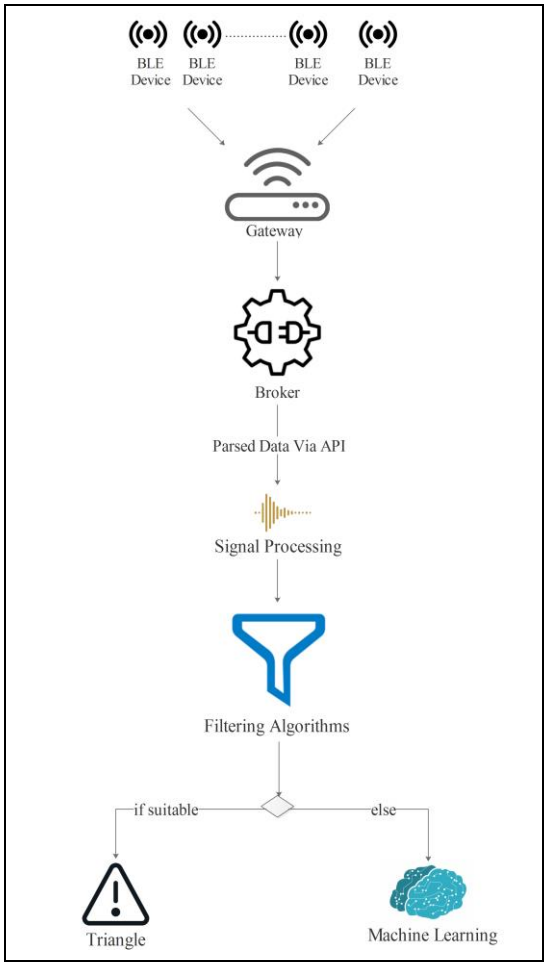

Figure 1. Architecture of the proposed system

The RSSI value of each sensor will be extracted from the data parsed by the broker. As BLE may contain noise, the mean or median of M RSSI values from each sensor will be calculated in the third phase. M represents the number of last received values from sensors, and this number will be optimized. When the mean or median is calculated, three RSSI values measured by the most closest gateways for each beacon will be selected (i.e., three gateways with the highest measuredPower-RSSI values, which is explained in Equation 3). Next, the RSSI value is converted to distance, measured in meters, using Equation 3.

$$d = 10^{\frac{measuredPower-RSSI}{10*N}} \quad (3)$$

In this equation, "measured power" represents the RSSI value of a gateway measured in one meter, "RSSI" represents the measured RSSI of a sensor, and "N" represents constants that depend on environmental factors. For each gateway, assume that there is a circle with a radius d, and the target is on that circle. In the fifth phase, whether these three circles are suitable for triangulation or not will be determined. Most common pattern of three circles for a triangulation on a two-dimensional plane are shown in Figure 2 [31]. The circles can be tangent to each other, but, in this case, only one intersection point exists.

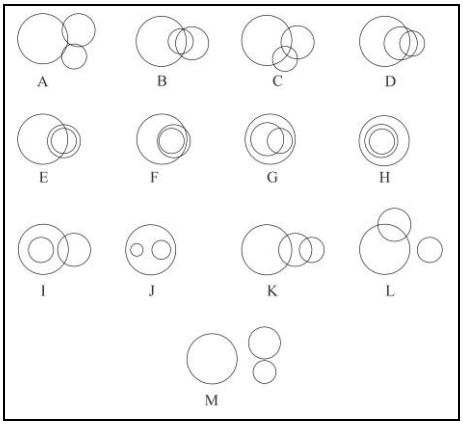

Figure 2. Three circles arranged in 13 possibilities.

In this figure, positions A, B, C, and D are suitable for triangulation, but the others are not. To triangulate, the intersection of three circles must be known. The intersection points of two circles can be found using the following equations.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (4)$$

$$a = ((r_1)^2 - (r_2)^2 + d^2)/(2 \times \mathrm{d}) \quad (5)$$

$$h = \sqrt{(r_1)^2 - a^2} \quad (6)$$

$$x_{int1} = x_1 + a * \frac{(x_2 - x_1)}{d} + h * (\frac{(y_2 - y_1)}{d}) \quad (7)$$

$$x_{int2} = x_1 + a * \frac{(x_2 - x_1)}{d} - h * (\frac{(y_2 - y_1)}{d}) \quad (8)$$

$$y_{int1} = y_1 + a * \frac{(y_2 - y_1)}{d} + h * (\frac{(x_2 - x_1)}{d}) \quad (9)$$

$$y_{int2} = y_1 + a * \frac{(y_2 - y_1)}{d} - h * (\frac{(x_2 - x_1)}{d}) \quad (10)$$

In these equations, x_1, x_2, y_1 and y_2 represent the center coordinates of two circles while r_1 and r_2 represent the radius of two circles, which are calculated using the RSSI to distance formula, and x_{int1}, x_{int2}, y_{int1}, and y_{int2} represent the coordinates of the intersections' points. These equations are applied three times (for each pair of circles), thus calculating six intersection points. When the intersection points are calculated, one needs to ensure that the circles are like circles A, B, C, D, E, and G in Figure 2. This can be verified using Equation 11.

$$k = \sqrt{(x_{int1} - x_3)^2 + (y_{int1} - y_3)^2} \quad (11)$$

In this equation, x_{int1} and y_{int1} represent the intersection point of circle1 and circle2, and x_3 and y_3 represent the center coordinates of circle3. If k> r_3, where r_3 represents the radius of circle3, then the intersection point of circle1 and circle2 is in circle3, which verifies that the circles are like A, B, C, D, E, and G in Figure 2. However, only positions A, B, C, and D are suitable for triangulation. Therefore, this calculation must be

augmented. Thus, we use the d value shown in Equation 4. If $d > r_1 + r_2$, then the two circles are separate. If $d < |r_1 - r_2|$, then one circle is covered by another. If $d = 0$ and $r_1 = r_2$, then the circles are coincident. When these rules are applied to each pair of circles, circle E and G can be eliminated because, in these cases, one circle is covered by at least one other circle. In the final phase of the proposed model, if the target signal meets the necessary conditions, its position is calculated by triangulation. Otherwise, a trained machine learning model will estimate the target's position. For each application, the accuracy of a machine learning model is determined by preliminary work. In these steps, the necessary parameters for the machine learning method are optimized and appropriate methods are selected by a final decision-maker.

4 Experiment results

In this study, datasets needed for the machine learning model are generated from the area, which is divided into equal-sized square regions. Bluetooth gateways are placed at the corners of each region. RSSI values are measured from different parts of each region with Bluetooth beacons, and each region is assigned a name. Here, the number of classes is equal to the number of regions, and the number of features is equal to the number of gateways. For this study, two different fields were created for data observation, differentiated by the distance between the gateways. In addition, three datasets were generated in different field conditions that can be named as easy, medium and hard cases.

To test our model, three different experimental environments were created to simulate easy, medium, and hard cases. In the easy case, a 144 square meter noise-free area was divided into 9 units of 16 square meters. In the medium case, some noise, such as noise from a crowd, was added during the data collection and testing phases. These two cases can also be described as a simulation environment, because no real-time testing has been done. In the hard case, an environment was created in a İstanbul airport. In this case, a 6400 square meter area was divided into 16 units of 400 square meters. The layouts of these environments are shown in Figure 3. Note that this environment is a real time testing area in İstanbul airport.

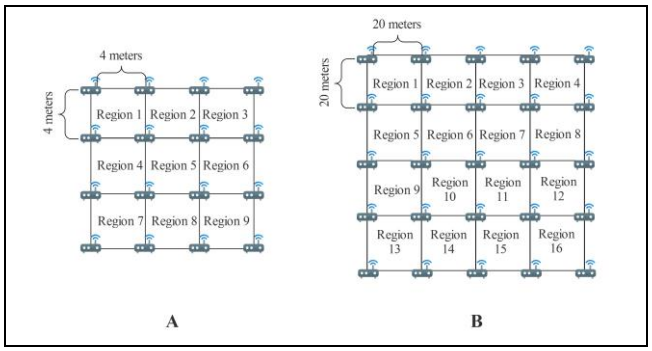

Figure 3. Layouts of the created environments (A for easy and medium cases, B for the hard case).

For both cases, many samples were observed from different parts of each region. The number of classes is equal to the number of regions; therefore, the easy and medium cases have nine different classes, and the hard case has sixteen different classes. Thanks to this information, we observed 180 samples for the easy case, 360 samples for the medium case, and 640 samples for the hard case. In addition, we ensured a balanced dataset. For this purpose, 20 samples from each region of easy case, 40 samples from each region of medium case and 40 samples from each region of hard cases were collected. These samples vary in terms of time collected, location of the collector in the region, position of the Beacon on the body, noise intensity of environment and the collector of the data. In the dataset collection process, it was aimed to collect data appropriate for the noises that may occur by moving the Beacon at different points of the body. For each dataset, 10% of the data was randomly selected as a validation set and 10% was selected as a test set. The remaining 80% was used to train the model. This test set is used to measure the accuracy of the machine learning model. Five machine learning algorithm, which are k-nn, Naïve Bayes, LR, ANN and SVM, were selected for our hybrid model, because they are most common used machine learning algorithm and easy to implement. The main aim of the study was to measure effect of hybrid model, which uses combination of machine learning algorithm and triangulation, for the performance of IL. Therefore, it is assumed that, effect of hybrid model on performance of IL can be measured quickly using these five machine learning models. The proposed model was applied to real-time streaming data to measure its performance. The scikit-learn library in Python was used for all machine learning models [32]. Using training and validation sets, we optimized the k parameters for the k-nn algorithm, the C parameter for LR, the C and gamma parameters for SVM, and the learning rate, number of units in hidden layer, and the number of epochs for ANN. Optimum parameters and search space are shown in Table 1. In this table, E represents the easy case, M represents the medium case, and H represents the hard case.

After parameter optimization, the models were trained and tested using three different datasets. Table 2 shows the accuracy, precision, recall and F1-score of each model for each dataset. As mentioned before, our data was labeled with more than two classes. For this purpose, weighted average technique was used to compute precision and recall. The formulas of these performance metrics can be found at related information document [33]. Data collection, optimization, training, and test phases will be repeated every time the environment change. In this table, blond values represent the best score for each metric of each environment.

When the machine learning model predicts the target's position as a region, the target can be anywhere in the region. Therefore, even when the machine learning model predicts the region correctly, the mean error will be 2 meters for the easy and medium cases and 10 meters for the hard case. If the position can be triangulated, it will be more accurate, however, triangulation is not always possible. Thus, in the final phase, the best model for each environment was selected, and the model was trained using a concatenation of the training and validation sets. As shown in Table 2, k-nn and LR are the best models for the easy case while, for the medium and hard cases, k-nn is the best model based on the accuracy metric. Based on all other metrics, k-nn is the best model for all environments. Although k-nn was the most accurate for all environments, our experiments indicate that k-nn is not suitable for use with real-time streaming data. As mentioned previously, k-nn uses a training set in every prediction, which causes a delay in position estimation. Therefore, LR, which is the second-best algorithm for all environments, is used for the real-time prediction model. The solo triangulation and machine learning models were both tested with real-time streaming data.

Table 1. Parameter optimization results and search space.

Method	Parameter Name	Search Space	Optimum Values
k-nn	k	{1,3,5,7,9,11,13,15,17}	E = 3, M = 3, H = 5
LR	C	$\{2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}\}$	E = 0.25, M = 0.5, H = 0.25
SVM	C	$\{2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}\}$	E = 1, M = 0.5, H = 0.25
SVM	gamma	$\{2^{-6}, 2^{-5}, 2^{-4}, 2^{-3}, 2^{-2}, 2^{-1}, 2, 2^{1}, 2^{2}, 2^{3}, 2^{4}, 2^{5}, 2^{6}\}$	E = 0.625, M = 0.03125, H = 0.5
ANN	learning rate	{0.5,0.3,0.1,0.05,0.03,0.01,0.005,0.003,0.001}	E = 0.01, M = 0.01, H = 0.03
ANN	hidden layer	{10,20,30,40,50,60,79,80,90,100,110,120,130,140,150}	E = 80, M = 30, H = 120
ANN	epochs	{50,75,100,125,150,175,200,225,250,275,300,325,350}	E = 225, M = 325, H = 150

Table 2. Accuracy of machine learning models for three environments.

Environment	Performance Metric	k-nn	Naïve Bayes	LR	ANN	SVM
Easy Case	Accuracy	**94.44%**	77.70%	**94.44%**	88.88%	77.70%
	Precision	93.75%	72.52%	**95.65%**	86.66%	75.25%
	Recall	**95.74%**	81.48%	93.61%	90.69%	82.02%
	F1-Score	**94.73%**	76.74%	94.62%	88.63%	78.49%
Medium Case	Accuracy	**91.66%**	66.60%	88.88%	83.33%	69.44%
	Precision	**90.42%**	64.70%	87.84%	82.44%	61.32%
	Recall	**93.40%**	73.33%	89.83%	85.16%	73.50%
	F1-Score	**91.89%**	68.75%	88.82%	83.78%	66.86%
Hard Case	Accuracy	**85.93%**	70.31%	84.37%	81.25%	64.06%
	Precision	**87.30%**	71.26%	85.04%	79.68%	63.63%
	Recall	**85.19%**	73.37%	84.00%	81.75%	65.62%
	F1-Score	**86.23%**	72.30%	84.52%	80.70%	64.61%

Table 3. Mean error of models in three environments, measured in meters.

Environment	Triangulation	Machine Learning	Proposed Model
Easy Case	**1.3 meters**	2.4 meters	1.6 meters
Medium Case	**1.8 meters**	2.6 meters	1.9 meters
Hard Case	19 meters	16 meters	**14.3 meters**

To calculate the mean error in the machine learning model, we assume that every correct prediction has an error of $\frac{l}{2}$ meters where l represents the edge size of a region and that every incorrect prediction has an error of $\frac{l}{2} + dis$ meters where dis represents the distance from the prediction region to the real target region. For example, the distance between region1 and region2 in the easy case is eight meters, which is the distance between the center points of two regions. According to this information, Table 3 summarizes the mean errors of three models for the three cases. The experiment's results indicate that, if gateways are placed densely throughout a noise-free area, triangulation alone is enough to calculate a target's position. Likewise, when noise is added to an area with densely placed gateways, triangulation is still sufficient to calculate the position. In this scenario, the proposed model obtained similar results to the triangulation model. However, for noisy environments with sparsely placed gateways, the machine learning model was more accurate than triangulation, and the proposed model obtained the best results.

5 Conclusion

In this paper, we propose a novel approach for BLE-based indoor localization that uses a combination of triangulation and machine learning. Five machine learning models (Naïve Bayes, k-nearest neighbor, logistic regression, support vector machines, and artificial neural networks) were optimized and used for prediction. The results show that k-nn performs the best among the machine learning models, but it is the slowest model when using real-time streaming data. Therefore, LR is used in the proposed model. When gateways are placed close together, triangulation obtains the smallest degree of error. However, as the distance between gateways increases, the degree of error with triangulation increases dramatically, and the proposed model yields better results. For a future study, we plan to develop a scalable horizontally ready application of the proposed model and we will test it in different environments. We also plan to increase the sample size and apply deep learning models.

6 Acknowledgement

This study is an output of studies conducted in Detaysoft research and development center. We appreciate their support. The numerical calculations reported in this paper were fully/partially performed at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources)

7 Author contribution statement

In this study, Yasin GÖRMEZ created the idea, designed the system, created the implementation steps and wrote the analysis codes; Halil ARSLAN evaluated the results, made improvements in the system design, supported the parameter optimization processes; Yunus Emre IŞIK supported the article writing and analysis coding stages; Sercan TOMAÇ assisted in the data collection processes and coded the necessary Broker system for data collection.

8 Ethics committee approval and conflict of interest statement

Ethics committee approval is not required for the prepared article.

There is no conflict of interest with any person/institution in the prepared article.

9 References

[1] Calderoni L, Ferrara M, Franco A, Maio D. "Indoor localization in a hospital environment using Random Forest classifiers". *Expert Systems with Application*, 42(1), 125-134, 2015.

[2] Álvarez-Díaz N, Caballero-Gil P. "Decision support system based on indoor location for personnel management". *Remote Sensors*, 13(2), 248-257, 2021.

[3] Beaconstac. "10 Airports Using Beacons to Take Passenger Experience to the Next Level". https://blog.beaconstac.com/2016/03/10-airports-using-beacons-to-take-passenger-experience-to-the-next-level (2023).

[4] Jianyong Z, Haiyong L, C. Zili, Zhaohui L. "RSSI based bluetooth low energy indoor positioning". *International Conference on Indoor Positioning and Indoor Navigation,* Buson, Korea, 27-30 October 2014.

[5] Mussina A, Aubakirov S. "RSSI based bluetooth low energy indoor positioning". *IEEE 12th International Conference on Application of Information and Communication Technologies (AICT)*, Almaty, Kazkhstan, 17-19 October 2018.

[6] Yoon PK, Zihajehzadeh S, Kang BS, Park EJ. "Adaptive Kalman filter for indoor localization using Bluetooth Low Energy and inertial measurement unit". *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Milan, Italy, 25-29 August 2015.

[7] Xu B, Zhu X, Zhu H. "An efficient indoor localization method based on the long short-term memory recurrent neuron network". *IEEE Access*, 7(1), 123912-123921, 2019.

[8] Dinh TMT, Duong NS, Sandrasegaran K. "Smartphone-based indoor positioning using BLE iBeacon and reliable lightweight fingerprint MAP". *IEEE Sensors Journal*, 20(17), 10283-10294, 2020.

[9] Iqbal Z, Luo D, Henry P, Kazemifar S, Rozario T, Yan Y, Westover K, Lu W, Nguyen D, Long T, Wang J, Choy H, Jiang S. "Accurate real time localization tracking in a clinical environment using bluetooth low energy and deep learning". *Plos One*, 13(10), 205392-205393, 2018.

[10] Giuliano R, Cardarilli GC, Ceserani C, Di Nunzio L, Fallucchi F, Fazzolari R, Mazzenga F, Re M, Vizzarri A. "Indoor localization system based on bluetooth low energy for museum applications". *Electronics*, 9(6), 1055-1056, 2020.

[11] Peng Y, Fan W, Dong X, Zhang X. "An iterative weighted KNN (IW-KNN) based indoor localization method in bluetooth low energy (BLE) environment". *International IEEE Conferences on Ubiquitous Intelligence Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress,* Toulouse, France, 18-21 July 2016.

[12] Terán M, Aranda J, Carrillo H, Mendez D, Parra C. "IoT-Based system for indoor location using bluetooth low energy". *IEEE Colombian Conference on Communications and Computing (COLCOM)*, Cartagena, Colombia, 16-18 August 2017.

[13] Baronti P, Barsocchi P, Chessa S, Mavilia F, Palumbo F. "Indoor bluetooth low energy dataset for localization, tracking, occupancy, and social interaction". *Sensors*, 18(12), 4462-4464, 2018.

[14] Sadowski S, Spachos P. "RSSI-Based indoor localization with the internet of things". *IEEE Access*, 6(1), 30149-30161, 2018.

[15] Hou X, Arslan T. "Monte Carlo localization algorithm for indoor positioning using Bluetooth low energy devices". *International Conference on Localization and GNSS*, Nottingham, United Kingdom, 27-29 July 2017.

[16] Röbesaat J, Zhang P, Abdelaal M, Theel O. "An improved BLE indoor localization with kalman-based fusion: an experimental study". *Sensors*, 17(5), 951-977, 2017.

[17] Mackey A, Spachos P. "Performance evaluation of beacons for indoor localization in smart buildings". *IEEE Global Conference on Signal and Information Processing*, Montreal, Canada, 14-16 November 2017.

[18] Ji M, Kim J, Jeon J, Cho Y. "Analysis of positioning accuracy corresponding to the number of BLE beacons in indoor positioning system". *17th International Conference on Advanced Communication Technology*, PyeongChang, South Korea, 1-3 July 2015.

[19] Qureshi UM, Umair Z, Duan Y, Hancke GP. "Analysis of Bluetooth Low Energy (BLE) based indoor localization system with multiple transmission power levels". *IEEE 27th International Symposium on Industrial Electronics*, Cairns, Australia,13-15 June 2018.

[20] Kayış O, Çakmak Y, Utku S. "Indoor navigation system with using mobile devices". *Pamukkale University Journal of Engineering Sciences*, 24(2), 238-245, 2018.

[21] Hou X, Arslan T, Juri A, Wang F. "Indoor localization for bluetooth low energy devices using weighted off-set triangulation algorithm". *Proceedings of the 29th International Technical Meeting of the Satellite Division of The Institute of Navigation*, Portland, USA, 12-16 September 2016.

[22] Yu X, Wang H, Wu J. "A method of fingerprint indoor localization based on received signal strength difference by using compressive sensing". *EURASIP Journal on Wireless Communications and Networking*, 2020(1), 1-13, 2020.

[23] Peterson LE. "K-nearest Neighbor". http://www.scholarpedia.org/article/K-nearest_neighbor (21.02.2009).

[24] Murphy KP. "Naive Bayes Classifiers". University of British Columbia, Vancouver , Canada, 18, 2006.

[25] Amazon Web Services. "Lojistik Regresyon Nedir?-Lojistik Regresyon Modeline Ayrıntılı Bakış-AWS". https://aws.amazon.com/tr/what-is/logistic-regression (24.02.2023).

[26] Wright RE. "Logistic regression". *American Psychological Association*, 10(2), 217-244, 1995.

[27] Jain AK, Mao J, Mohiuddin KM. "Artificial neural networks: a tutorial". *Computer*, 29(3), 31-44, 1996.

[28] Noble WS. "What is a support vector machine?". *Nature Biotechnology*, 24(12), 1565-1567, 2006.

[29] Mesut A, Öztürk E. "A method to improve full-text search performance of MongoDB". *Pamukkale University Journal of Engineering Sciences*, 28(5), 720-729, 2022.

[30] This reference is hidden Yasin G, Halil A, Ömer Faruk K. "Efficient and Scalable Broker Design for the Internet of Things Environments". *28th Signal Processing and Communications Applications Conference (SIU)*, Gaziantep, Turkey, 05-07 October 2020.

[31] Gwon Y, Jain R, Kawahara T. "Robust indoor location estimation of stationary and mobile users". *IEEE INFOCOM*, Hong Kong, China, 07-11 March 2004.

[32] scikit-learn: machine learning in Python. "Scikit-Learn 0.23.2 Documentation". https://scikit-learn.org/stable/ (20.10.2020).

[33] Wikipedia. "Precision and Recall". https://en.wikipedia.org/w/index.php?title=Precision_and_recall&oldid=1122267443 (24.02.2023).