Enhanced Indoor Localization Based BLE Using Gaussian Process Regression and Improved Weighted kNN

DAI DUONG NGUYEN AND MINH THUY LE, (Member, IEEE)
School of Electrical Engineering, Hanoi University of Science and Technology, Hanoi 10000, Vietnam
Corresponding author: Minh Thuy Le (thuy.leminh@hust.edu.vn)

This research was funded by Hanoi University of Science and Technology under grant number T2021-SAHEP-007.

ABSTRACT Indoor positioning has attracted commercial developers and researchers in the last few decades. Global positioning system (GPS) cannot well localize in indoor environment. For clinical applications such as workflow analysis, Bluetooth Low Energy (BLE) beacons have been employed to map the positions of individuals in indoor environments. The received signal strength indicator (RSSI) fingerprinting plays a key role in the access point performance services. This paper deals with the issue of indoor localization based RSSI fingerprint using only RSSI vectors without any prior knowledge of the pose. We proposed to use machine learning (ML) combined with modified kNN algorithm to enhance the real-time performance and propose method to detect uncertainty of estimated pose. The mentioned ML algorithm is Gaussian Process Regression (GPR). In the online phase of our system, GPR gives a prediction for the location of the pose using RSSI vector. This prediction helps kNN algorithm therefore limits the searching region leading to reduce the computational cost. An analysis on the distribution of k nearest points is also presented which aims to evaluate the confidence of the estimated pose for extracting a list of trustable points. Furthermore, we also present an extrapolation process using this trust-list to have an optimized trajectory. The accuracy and timing analysis of our proposal was realized on a challenging BBIL dataset which contains very noisy RSSI signals due to a fast moving of the object. The experimental results show that our system exhibits a much better performance than traditional kNN or WkNN algorithms. The RMSE of our optimal trajectory is 1.78 m in the room with a dimension of 10 m x 25 m, which is competitive in comparison to other methods where initial pose is known.

INDEX TERMS Indoor localization, Bluetooth low energy, RSSI fingerprint, Gaussian process regression, machine learning.
Machine learning (ML) is now a potential solution for indoor localization research. One of the most important advantages of ML is the ability to effectively and rapidly make decisions in online-phase from observed data without time-consuming mathematical formulation. The widely use of machine learning is due to the growing computational advantages of high security, low power consumption, and low cost. However, the comparison between Wi-Fi and BLE is still debatable because Wi-Fi has a longer range than BLE.

In this paper, we focus on the accuracy of indoor localization issue based on only RSSI vector and no-prior pose is known. Our distributions are: 1) To avoid the disadvantages of previous work, our system makes a fusion of Gaussian process regression (GPR) and kNN methods. In practice, we propose to apply GPR to predict the searching area, then only perform kNN calculation in this area to help reduce the number of calculations in pose estimation. 2) We propose an evaluation model to determine whether estimated pose of kNN method is reliable or not. 3) We propose an optimization processing based on reliable points to have a more accurate trajectory.

The rest of this paper is structured as follows: Section II presents previous studies related to the research of our work. Section III introduces BBIL dataset used for experiments. Our localization system is then described in Section IV. Section V discusses the performance of our proposal on BBIL dataset. Finally, Section VI consists of the conclusions and the future directions for our research.

II. RELATED WORKS

In 2010, Altini et al. [26] proposed to use a combination of neural networks to locate users in the corridor based on Bluetooth with an average error of 0.5 m. However, the author does not mention the size of the corridor and the number of broadcasting stations. Another study on BLE wave based positioning by Sthapit et al. [27] used a classifier with two popular machine learning algorithms, Support Vector and Logistic Regression and fingerprint technique to help determine the location of the object using the BLE wave RSSI parameter. The study tested the results in the university corridor, the size of the space is 4.5 m x 15 m, including 14 BLE broadcasting modules. The average error is about 0.5 m. The idea of the method is to divide the space into 0.45m x 0.45m squares. A large number of samples will be collected in each plot to form a set to train the classifier. Thus, for each signal to be positioned, the classifier calculates the probabilities that the signal belongs to the square regions. The position of the point will be calculated as the average of the positions of the squares with the weighted to the calculated probabilities. This method of dividing cells is suitable for empty spaces, free from large objects in the middle of the room. In contrast, for spaces with many objects, this method of evenly dividing cells is no longer suitable. Another study by Alexander and Kusuma [28] in 2019 also used navigation proposed by Sun et al. [22] was another example of applying ML technique. Besides, ML also exhibits as an effective way to make a fusion of multi-dimensional data collected from multiple positioning sensors, technologies and methods. For example, Guo et al. [23] and Alhajri et al. [24] present the applying of both supervised and unsupervised learning for fusion weight generation. Furthermore, when there is a lack of labeled data, Liu et al. [25] proposed to use transfer learning in fingerprint-based localization to enhance system scalability without excessive site surveys while ensuring the accuracy.

In this paper, we focus on the accuracy of indoor localization based on only RSSI vector and no-prior pose is known. Our distributions are: 1) To avoid the disadvantages of previous work, our system makes a fusion of Gaussian process regression (GPR) and kNN methods. In practice, we propose to apply GPR to predict the searching area, then only perform kNN calculation in this area to help reduce the number of calculations in pose estimation. 2) We propose an evaluation model to determine whether estimated pose of kNN method is reliable or not. 3) We propose an optimization processing based on reliable points to have a more accurate trajectory.

The rest of this paper is structured as follows: Section II presents previous studies related to the research of our work. Section III introduces BBIL dataset used for experiments. Our localization system is then described in Section IV. Section V discusses the performance of our proposal on BBIL dataset. Finally, Section VI consists of the conclusions and the future directions for our research.

II. RELATED WORKS

In 2010, Altini et al. [26] proposed to use a combination of neural networks to locate users in the corridor based on Bluetooth with an average error of 0.5 m. However, the author does not mention the size of the corridor and the number of broadcasting stations. Another study on BLE wave based positioning by Sthapit et al. [27] used a classifier with two popular machine learning algorithms, Support Vector and Logistic Regression and fingerprint technique to help determine the location of the object using the BLE wave RSSI parameter. The study tested the results in the university corridor, the size of the space is 4.5 m x 15 m, including 14 BLE broadcasting modules. The average error is about 0.5 m. The idea of the method is to divide the space into 0.45m x 0.45m squares. A large number of samples will be collected in each plot to form a set to train the classifier. Thus, for each signal to be positioned, the classifier calculates the probabilities that the signal belongs to the square regions. The position of the point will be calculated as the average of the positions of the squares with the weighted to the calculated probabilities. This method of dividing cells is suitable for empty spaces, free from large objects in the middle of the room. In contrast, for spaces with many objects, this method of evenly dividing cells is no longer suitable. Another study by Alexander and Kusuma [28] in 2019 also used navigation proposed by Sun et al. [22] was another example of applying ML technique. Besides, ML also exhibits as an effective way to make a fusion of multi-dimensional data collected from multiple positioning sensors, technologies and methods. For example, Guo et al. [23] and Alhajri et al. [24] present the applying of both supervised and unsupervised learning for fusion weight generation. Furthermore, when there is a lack of labeled data, Liu et al. [25] proposed to use transfer learning in fingerprint-based localization to enhance system scalability without excessive site surveys while ensuring the accuracy.

In this paper, we focus on the accuracy of indoor localization based on only RSSI vector and no-prior pose is known. Our distributions are: 1) To avoid the disadvantages of previous work, our system makes a fusion of Gaussian process regression (GPR) and kNN methods. In practice, we propose to apply GPR to predict the searching area, then only perform kNN calculation in this area to help reduce the number of calculations in pose estimation. 2) We propose an evaluation model to determine whether estimated pose of kNN method is reliable or not. 3) We propose an optimization processing based on reliable points to have a more accurate trajectory.

The rest of this paper is structured as follows: Section II presents previous studies related to the research of our work. Section III introduces BBIL dataset used for experiments. Our localization system is then described in Section IV. Section V discusses the performance of our proposal on BBIL dataset. Finally, Section VI consists of the conclusions and the future directions for our research.

II. RELATED WORKS

In 2010, Altini et al. [26] proposed to use a combination of neural networks to locate users in the corridor based on Bluetooth with an average error of 0.5 m. However, the author does not mention the size of the corridor and the number of broadcasting stations. Another study on BLE wave based positioning by Sthapit et al. [27] used a classifier with two popular machine learning algorithms, Support Vector and Logistic Regression and fingerprint technique to help determine the location of the object using the BLE wave RSSI parameter. The study tested the results in the university corridor, the size of the space is 4.5 m x 15 m, including 14 BLE broadcasting modules. The average error is about 0.5 m. The idea of the method is to divide the space into 0.45m x 0.45m squares. A large number of samples will be collected in each plot to form a set to train the classifier. Thus, for each signal to be positioned, the classifier calculates the probabilities that the signal belongs to the square regions. The position of the point will be calculated as the average of the positions of the squares with the weighted to the calculated probabilities. This method of dividing cells is suitable for empty spaces, free from large objects in the middle of the room. In contrast, for spaces with many objects, this method of evenly dividing cells is no longer suitable. Another study by Alexander and Kusuma [28] in 2019 also used
the fingerprint method and combined with machine learning for BLE wave based positioning. In that study, the author uses two different neural networks to individually predict for each X and Y coordinate. The author tested with 4 machine learning algorithms: Artificial Neural Network Regression (ANN), Multiple Linear Regression (MLR), Random Forest Regression (RF), Support Vector Regression (SVR). The database was collected in a room with the size of 4m x 6m and the average positioning error was from 1.34m to 1.46m. Most recently, Hoang et al. [29] used the Recurrent neural network (RNN) also with the fingerprint method to achieve an average positioning error of 0.75m in a space with a ruler of 16m x 21m. However, the RNN in the study assumes that the object’s movement is smooth and the signal is not interrupted. The reason is that the RNN uses the current RSSI signal along with the entire distance the object has traveled in the past to make predictions. The paper [30] proposed the accuracy analysis of indoor Wi-Fi signal based on localization using two approach: Gaussian process regression and K-nearest neighbor (KNN). According to the results, the author argues that the analysis of corridors using KNN could provide more accuracy than GPR for testing environment. However, in terms of processing time, we note that GPR works much faster than KNN, especially for the dataset containing huge number of RSSI reference points. This is the reason for our research in this paper in combining the GPR and KNN to have a low computation cost why ensure the accuracy.

III. DATASET

In order to compare our proposal to other methods with a fair assessment, we use a standard dataset with ground truth points measured accurately of other laboratory. The selected dataset is BBIL (BLE Beacon Indoor Localization Dataset) [31].

This database was collected from September 2018 to May 2019. The size of the office space is roughly 11m x 25m. In the experiment, the author group of the database uses the model of fixed receiver and mobile transmitting station. This means that the device that needs to be located will broadcast, and the fixed device will receive. This role change does not affect the positioning algorithm, only one

problem is that when receiving waves, the fixed devices have to sync the time with each other. The red circles represent room-mounted BLE receivers for a total of 9 receivers. Devices used in receiving stations are Raspberry Pi kits and transmitters are BLE beacons. The exact locations of the equipment were measured based on the landmarks attached to the room. The average distance between landmarks is about 1.33m.

To get sample points, transmitters were moved in the room on random days and 3 times a day at any time. Figure 4 shows a summary of the RSSI sampling reference points after all. These points are spread across the positioning area and each reference point gets an RSSI sample. The reference sampled data set has 9211 points. In addition, the authors also built a test data set of 2107 points obtained after 11 moves with different orbits in the same room with the same equipment. However, in one trajectory, we found that there is a problem in connection of Raspberry Pi kits with 3 receivers so that the RSSI values do not change while moving so that we considered only the 10 remaining trajectories in evaluations.

IV. LOCALIZATION ALGORITHM BASED MACHINE LEARNING AND FINGERPRINT

A. PROPOSED ALGORITHM

For the earlier algorithms, where machine learning is not used, the most common method for locating is kNN (k nearest neighbor). The idea is to find k reference points that are closest to the point to be located in terms of RSSI vector, then the position is determined to be the weighted average of the above points. A problem arises with this method for each location to be located, we have to scan all the reference points so the calculation time will be high when the reference set is large. Alternatively, many authors recently use machine learning such as Recurrent Neural Network (RNN) [29] or Convolution Neural Network (CNN) with Gaussian Process Regression [32] to improve localization accuracy. RNN focus on the tracking problem so that it requires the first position to be known. Otherwise, CNN with Gaussian regression requires a large training dataset with the reference

FIGURE 1. BBIL dataset environment.

FIGURE 2. BBIL reference points of the environment constructed by 43 training trajectories.
point spread in the location environment to have a high accuracy.

In our research, we focus on the case when no prior pose is known. This is the more popular case in practice. Our system tries to attain two milestones: the first is to track current point as fast as possible to have a real-time response and the second work is to reconstruct a optimal trajectory. The offline phase and online phase of our algorithm are shown in figure 3. In offline phase, the RSSI fingerprint collected from training trajectories are divided into training set and validation set. Training set is used to train GPR in order to have an optimal model which exhibits the smallest error on validation set. In online phase, there are two threads executed in parallel. Tracking thread takes RSSI vector of unknown position (UP) to estimate current pose. In order to have real-time performance, the core processing of tracking thread is a fusion of GPR and kNN method. RSSI vector will be passed the optimal trained GPR to have a predicted pose. Then, instead of searching in the whole reference points, we only apply our kNN method to search $k$ nearest neighbor located in the circle area centered in the predicted pose with radius $r$. The output tracking pose is the weighted average of these $k$ nearest neighbor. Concurrently, Optimization thread will evaluate all tracked poses with stricter condition to reconstruct an optimal trajectory. Due to the fact that RSSI is noisy so that the result of kNN method is not reliable at some points, the block “Reliable point detection” examines the confidence at each pose based on the distribution of the $k$ nearest neighbors. If the pose is trustable, it will be added to a trust-list. The block “trajectory reconstruction” will use this trust-list to re-optimize trajectory.

**B. FINGERPRINT FILTERING**

The receivers station in the environment are denoted $AP_1, AP_2, \ldots, AP_n$ where $n$ is number of station. Information gathered from the $AP$s are the RSSI value and MAC address of that device. Since at each point, the vector RSSI is the set of values obtained from all the $AP$s, it is easy to see that the larger the number of $n$, the higher the distinction between the vectors is obtained. To get high positioning accuracy, we need to have a sufficient number of $AP$s covering that space.
Given that RSSI is noisy and it is collected while mobile device is moving, the raw values of RSSI have a huge variation. The figure 4 show the variation of RSSI values of one receiver station along one of the training trajectories. We can see that those values have some critical changes even though two positions are very close to each other. In many applications, to have a reliable RSSI value at one location, the mobile device must be set at that position for along time before being located. However, it is not suitable for application where mobile devices move continuously. In our implementation, to avoid critical changes of RSSI, we propose the step of filtering. It means that RSSI set of each training trajectory will be passed a low-pass filter. We evaluated two kinds of filter: mean and median.

\[
\text{RSSI}_{\text{APk}}^{t} = \text{mean}(\text{RSSI}_{\text{APk}}^{t-w}, \ldots, \text{RSSI}_{\text{APk}}^{t}, \ldots, \text{RSSI}_{\text{APk}}^{t+w}) \quad (1)
\]

\[
\text{RSSI}_{\text{APk}}^{t} = \text{median}(\text{RSSI}_{\text{APk}}^{t-w}, \ldots, \text{RSSI}_{\text{APk}}^{t}, \ldots, \text{RSSI}_{\text{APk}}^{t+w}) \quad (2)
\]

Formula (1) and (2) illustrate the filtering applied for one element of RSSI vector and it is similar to other elements. \(\text{RSSI}_{\text{APk}}^{t-w}, \text{RSSI}_{\text{APk}}^{t}, \text{RSSI}_{\text{APk}}^{t+w}\) corresponds to the value received by receiver station \(\text{APk}\) at time \(t, t - w\) and \(t + w\) respectively. We also call \(w\) as “window size” of filtering. The value of \(w\) must be selected so that all points in the filter are located geographically close to each other. It is obviously that the value of \(w\) depends on the moving speed of mobile device. In BBIL dataset, with maximum speed is around 2m/s and sampling time-step is 0.5 second, we choose the value of \(w\) equal to 2. Figure 4b and 4c show the result of median and mean filtering for the raw RSSI of figure 4a. We can note that critical changes are decreased and RSSI values are smoother. In practice, during the experiments, mean filter exhibits slightly more precise than median filter but we found that localization performances are not much different.

### C. GAUSSIAN PROCESS REGRESSION

Generally, a Gaussian process (GP) is usually specified by its mean function and covariance function [33]. The mean function \(\mu(x)\) and the covariance function \(k(x, x')\) of a real process \(f(x)\) and target value \(y\) were defined as in formulas (3)-(6).

\[
f(x) \sim \text{GP}(\mu(x), k(x, x')) \quad (3)
\]

\[
\mu(x) = E[f(x)] \quad (4)
\]

\[
k(x, x') = \text{cov}(f(x), f(x')) \quad (5)
\]

\[
y = f(x) + \omega \quad (6)
\]

A GP is defined as a collection of random variables. In this paper, \(x\) is RSSI vector and \(y\) is the location coordinates in \(\text{R}\). \(\omega\) is additive Gaussian noise with zero mean and a known variance denoted as \(\sigma^2_\omega\). In BBIL dataset, we have \(n = 9211\) pairs of training data, where each input data \(x\) is a signal vector and each output data \(y\) is 2D-localization pose. Noting that \(x\) is multi-dimensions but the value of \(y\) in formula 6 is scalar so that we apply GPR separately for x-axis and y-axis of 2D-pose. The GP defines posterior distributions over function from input data to output data \(f(x_1), f(x_2), f(x_3), \ldots, f(x_n)\) obeys a multivariate Gaussian distribution shown in formulas (7)-(9).

\[
\begin{align*}
[f(x_1), f(x_2), f(x_3), \ldots, f(x_n)] & \sim N(M, K) \quad (7) \\
M &= E[f(x_1), f(x_2), \ldots, f(x_n)] \quad (8) \\
K_{ij} &= k(x_i, x_j) \quad (9)
\end{align*}
\]

where \(M\) is the mean and \(K\) is the kernel function of covariance. In the concept of kernel, there are three kinds of kernel that could be suitable for RSSI vectors such as Squared Exponential (SE), Periodic (PER) and Matern. These kernel functions are shown in formulas (10)-(13).

\[
k_{\text{se}}(x, x') = \sigma^2_\text{e} \exp(-\frac{||x - x'||^2}{2l^2}) \quad (10)
\]

\[
k_{\text{per}}(x, x') = \sigma^2_\text{e} \exp\left(-\frac{2\sin^2(\pi \frac{||x - x'||}{p})}{l^2}\right) \quad (11)
\]

\[
k_{\text{matern}3/2}(x, x') = \left(1 + \frac{\sqrt{3}d(x, x')}{l}\right) \exp\left(-\frac{\sqrt{3}d(x, x')}{l}\right) \quad (12)
\]

\[
k_{\text{matern}5/2} = \left(1 + \frac{\sqrt{5}d(x, x')}{l}\right) \exp\left(-\frac{\sqrt{5}d(x, x')}{l}\right) + \frac{5d(x, x')^2}{3l^2} \exp\left(-\frac{\sqrt{5}d(x, x')}{l}\right) \quad (13)
\]

where \(\sigma^2_\text{e}\) is the signal variance and is also considered as an output-scale amplitude and parameter \(l\) is the input-length-scale that determines the strength of the correlation between inputs. Before training the GPR model, the signal variance is set to 1.0, period is set to 1.0 and the length-scale is set to 0.2. Since our system uses RSSI vector to give the first predicted 2D-pose, the training set for regression was built:

- **Input data**: \(N\) elements of \(m\)-dimensional RSSI vectors. \(N\) is number of reference points. In BBIL dataset, \(N\) gets the value of 9211 and \(m = 9\).
- **Output data**: \(N\) elements of 2D-pose corresponding respectively to RSSI vectors.

The input data is high dimensional vectors leading to a complex nonlinear relationship with the output data. In previous researches, GPR proved the natural advantage of solving this problem [32], [34], [35]. In practice, we also tried many other regression methods such as Linear and Support vector machine to make a comparison. The training error shown in figure 5 proved that GPR algorithm is a better choice when the problem is a non-linear model. We can note that on the filtered training dataset, the GPR with Rational Quadratic kernel function has the best performance for X-axis. Otherwise, for Y-axis, GPR with exponential kernel function is the best. Therefore, we use these two models to predict the corresponding coordinate axis. The result on figure 5 is also a proof of the fingerprint filtering mentioned in IV-B when all...
the training errors applied on filtered RSSI dataset are lower than that applied on raw RSSI dataset.

**D. TRAJECTORY RECONSTRUCTION**

In localization based fingerprint, to locate unknown point (UP), the traditional kNN algorithm computes the similarity distance of RSSI vector between UP and reference point (RP). The similarity distance can be calculated by formula (14).

\[
D^{R \rightarrow U} = \sqrt{\sum_{i=1}^{m} (RSSI_{i}^{RP} - RSSI_{i}^{UP})^2}
\]  

(14)

where \(RSSI_{i}^{RP}\) is the \(i^{th}\) component of the RSSI vector of the reference point and \(RSSI_{i}^{UP}\) is the \(i^{th}\) element of the RSSI vector of the point to be located. Then \(k\) RPs with the smallest similarity distances are chosen and the UP coordinates is calculated by the average of these RPs’ coordinates shown in formula (15):

\[
(x_{UP}, y_{UP}) = \frac{\sum_{i=1}^{k} (x_{RPi}, y_{RPi})}{k}
\]  

(15)

In the WkNN algorithm, \(K\) RPs with the smallest spatial distances are selected as well, but with weighted average. Considering that the distances from each RP to the UP are different, the weights assigned to the reference points are different. The weights of the selected reference points can be calculated by formula (16); the weight is inversely proportional to the distance \(D^{R \rightarrow U}\). The coordinate of the UP is calculated by formula (17). The WkNN algorithm normally gives a better accuracy than the KNN so that it is frequently used in matching algorithm.

\[
\omega_{i}^{R \rightarrow U} = \frac{1}{D_{i}^{R \rightarrow U}}
\]

\[
(x_{UP}, y_{UP}) = \frac{\sum_{i=1}^{k} \omega_{i}^{R \rightarrow U} (x_{RPi}, y_{RPi})}{\sum_{i=1}^{k} \omega_{i}^{R \rightarrow U}}
\]  

(16)  

(17)

This traditional weighting is based on the similarity of RSSI vector. Lately, Peng et al. [36] proposed an enhanced method by adding the physical distance to weighting. This physical distance is also computed from RSSI by the signal attenuation model. However, with noisy signal due to fast moving, physical distance does not work. It is even worst than traditional weighting in our experiments.

Our work uses traditional WkNN algorithm but we propose novel processing that taking into account the distribution of \(K\) nearest neighbours to evaluate the estimated UP pose. The reason is that in some case, some points of \(K\) nearest points are located quite far from weighted position which can be called outliers but some others (inliers) are located closely. Let’s consider two distributions example illustrated on figure 6, we found that the weighted pose will have high possibility to be a good estimation if the majority of \(K\) nearest-points located close to it as shown in figure 6a. Otherwise, when \(K\) nearest-points are in widespread distribution as shown in figure 6b, weighted pose is not reliable. The block “Reliable point detection” realizes this evaluation at each tracking point. A nearest-point is considered as inlier when the distance from it to weighted pose is smaller than a threshold \(r_c\). If the number of inliers is bigger than \(\sigma K\) (\(\sigma = 0.5\) in our experiments), weighted pose is added to trustable-list.

The trajectory is then optimized based on this trustable-list. The idea is that when a reliable-point \(j\) is detected, the sub-trajectory from it to reliable-point \(j - 1\) will be re-computed. We suppose that there are \(h\) poses from reliable-point \(j - 1\) to reliable-point \(j\) denoted as \(P_{j-1,j}^{0}, P_{j-1,j}^{1}, P_{j-1,j}^{2}, ..., P_{j-1,j}^{h-1}\) where \(P_{j-1,j}^{0}\) is the pose of reliable-point \(j - 1\) and \(P_{j-1,j}^{h-1}\) is the pose of reliable-point \(j\). Noting that all poses of \(h\) point are estimated from WkNN methods by searching in the local reference-point set of GPR prediction (radius \(r\)). We call it the “large WkNN”. To make the extrapolation in sub-trajectory, we apply two-directions “narrow WkNN”. For forward direction, we take the initial point at \(P_{j-1,j}^{0}\) then re-compute \(P_{j-1,j}^{1}\) by searching \(k\) nearest-points located around \(P_{j-1,j}^{0}\) within a radius \(r_c\) (smaller than \(r\)) using RSSI vector. The new value of \(P_{j-1,j}^{1}\) is denoted as \(P_{j-1,j}^{f}\). The following pose \(P_{j-1,j}^{2}\) is then computed by applying narrow WkNN based on \(P_{j-1,j}^{f}\). This forward extrapolation process is done until \(P_{j-1,j}^{h-1}\). On the other side, we do the same work for backward direction but from the initial point considered at \(P_{j-1,j}^{0}\) to last point \(P_{j-1,j}^{h-1}\). The optimal pose at each point is the mean of three estimations as shown in formula (18).

\[
P_{j-1,j}^{\hat{P}} = \text{mean}(P_{j-1,j}^{0}, P_{j-1,j}^{f}, P_{j-1,j}^{h})
\]  

(18)

where \(P_{j-1,j}^{0}, P_{j-1,j}^{f}, P_{j-1,j}^{h}\) are respectively the result of “large WkNN”, “forward narrow WkNN” and “backward narrow WkNN”.

**V. EXPERIMENT**

We evaluate our system on 10 test trajectories of BBIL dataset. All parameters are not changed for all trajectories. We use root mean square error (RMSE) and its 90th percentile for comparison because it is commonly used by many other researches in the state of the art. In order to validate presented proposals, different test scenarios are:

- kNN(raw signal): Applying traditional kNN algorithm on raw RSSI signal available in dataset.
- WkNN(raw signal): Implementation of Weighted kNN algorithm also on raw RSSI signals.
- kNN(filtered signal): Applying traditional kNN algorithm on RSSI signals after filtering proposed in IV-B.
- WkNN(filtered signal): Weighted kNN on filtered RSSI signals.
- GPR+WkNN(filtered signal): The fusion of Gaussian Process Regression and WkNN on filtered RSSI signals without reconstruction.
• GPR+WkNN+Reconstruct: trajectory reconstruction after localization by GPR+WkNN.

Table 1 shows a RMSE comparison of different scenarios. We can note that WkNN is slightly more precise than kNN on all raw and filtered signals. Otherwise, filtering method brings an obviously enhancement in accuracy when RMSE in all trajectories are significantly reduced. The fusion of GPR and WkNN does not help to get a lower RMSE but it helps to accelerate the processing. As shown in table 2, execution time for estimating location of one pose is reduced by half from 350ms to 180ms. The reason is that GPR gives a location prediction and WkNN is realized only on reference points located in a limited region around this prediction instead of searching in the whole reference set. We do not mention the kNN in this table because it has a same algorithm complexity with WkNN so it has a similar timing performance.

When trajectory reconstruction is applied, RMSE is even better. The optimization helps RMSE to be decreased between 10% and up to 38% in almost all the trajectories except that in trajectory 6, RMSE of the reconstruction is slightly bigger but it is not too severe (2.14m compared to 2.08m). Figure 7 shows the number of trustable points and untrustable points in each trajectory based on the evaluating condition presented in IV-D. Following this, table 3 presents a RMSE of two sub-sets: Trustable-points set and Untrustable-points set in each trajectory. By observing table 1 and figure 7, the issue of accuracy degeneracy after reconstruction in trajectory 6 can be explained by the reasons that: 1) the result of WkNN is good enough in this trajectory before reconstruction and 2) the number of trustable points attains a high value of 75%. Therefore, doing optimization on an already good result could lead to a small degeneracy. On the other side, for other 9 trajectories, we can easily see the advantage of optimization process. We can note that trajectory reconstruction is done in parallel thread as illustrated in figure 3 hence it does not increase the execution time comparing to the version of GPR+WkNN only.

In table 4, we group the result of all 10 trajectories and figure 8 shows the CDF of Euclidean distance error. By observing the CDF, we can notice that ~ 90% of poses have the error smaller than 3m, 70% of poses having error within 2m with the version of GPR+WkNN+Reconstruct. Other versions have lower performance. In table 4, we make a com-
TABLE 1. RMSE in meter of different scenarios using raw signal and filtered signal for each test trajectories.

| Algorithm               | Traject.1 | Traject.2 | Traject.3 | Traject.4 | Traject.5 | Traject.6 | Traject.7 | Traject.8 | Traject.9 | Traject.10 |
|-------------------------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|
| kNN(raw signal)         | 3.69      | 3.50      | 4.30      | 4.11      | 3.30      | 3.30      | 3.58      | 3.81      | 3.42      | 4.40      |
| WkNN(raw signal)        | 3.57      | 3.38      | 4.12      | 4.00      | 3.24      | 3.18      | 3.43      | 3.74      | 3.40      | 4.26      |
| kNN(filtered signal)    | 2.64      | 2.62      | 3.22      | 2.90      | 2.20      | 2.23      | 2.70      | 2.91      | 2.25      | 3.80      |
| WkNN(filtered signal)   | 2.50      | 2.39      | 3.10      | 2.68      | 2.08      | 2.13      | 2.40      | 2.65      | 2.09      | 3.68      |
| GPR=WkNN(filtered signal)| 2.51      | 2.34      | 3.07      | 2.68      | 2.09      | 2.08      | 2.36      | 2.63      | 2.07      | 3.62      |
| GPR+WkNN+Reconstruct    | 1.97      | 1.91      | 1.95      | 2.02      | 1.51      | 2.14      | 2.17      | 2.05      | 1.81      | 2.54      |

TABLE 2. Mean of estimation time for each pose.

| Algorithm   | Pose estimation time |
|-------------|----------------------|
| WkNN        | 350 (ms)             |
| GPR+WkNN    | 180 (ms)             |

FIGURE 7. Report of the number of trustable-points and untrustable-points in each trajectory.

TABLE 3. Mean RMSE comparison between trustable-points set and untrustable-point set for each trajectory.

| Traject. | Mean of RMSE (m) | Trustable points | Untrustable points |
|----------|-----------------|-----------------|-------------------|
| 1        | 1.86            | 2.90            |
| 2        | 1.51            | 2.89            |
| 3        | 1.87            | 3.37            |
| 4        | 1.57            | 3.27            |
| 5        | 1.46            | 2.90            |
| 6        | 1.37            | 2.63            |
| 7        | 1.81            | 2.55            |
| 8        | 2.01            | 2.89            |
| 9        | 1.42            | 2.34            |
| 10       | 2.25            | 4.05            |

comparison to 3 others algorithm: Trilateration, Elman network (EN) and Long Short Term Memory (LSTM). The result of these 3 algorithms is presented by Kennedy et al. [31]-author of BBIL dataset. Trilateration method used path loss model to estimate distance from unknown pose to each BLE beacon and make a multi-trilateration to obtain its location. This method is strongly influenced by noise so that RMSE has a very high value at 6.32m. Otherwise, EN and LSTM are two algorithms using machine learning applied on RSSI fingerprint. The RMSE of these 2 latters are low at respectively 1.62m and 1.54m. However, the problem of these 2 methods is that for each trajectory, the truth location for the initial pose must be known which is not a popular case.

TABLE 4. RMSE and 90% percentile for whole dataset. The result of Trilateration, EN and LSTM methods (initial pose must be known) are collected from author of BBIL dataset [31] on website.

| Algorithm               | RMSE(m) | 90% percentile(m) |
|-------------------------|---------|--------------------|
| kNN(raw signal)         | 3.74    | 5.93               |
| WkNN(raw signal)        | 3.64    | 5.83               |
| kNN(filtered signal)    | 2.75    | 4.61               |
| WkNN(filtered signal)   | 2.56    | 4.20               |
| GPR+WkNN(filtered signal)| 2.54    | 4.10               |
| GPR+WkNN+Reconstruct    | 1.78    | 3.48               |
| Trilateration           | 6.32    | 7.24               |
| EN                      | 1.62    | 2.55               |
| LSTM                    | 1.54    | 2.30               |
FIGURE 9. Trajectories comparison.
FIGURE 9. (Continued.) Trajectories comparison.
in reality. Moreover, the result of EN and LSTM depends strongly on hyper parameters of network model so it is not stable for each launching.

Finally, figure 9 (at the end of the paper) shows result of trajectory built by our algorithm compared to ground-truth (orange curve). The result of our whole system is in blue while the implementation of WkNN with raw signals is in red and the combination of GPR+WkNN on filtered signals (without optimization) is in green. From left-side to right-side, we can observe easily the enhancement of our proposal by using signal filtering and then optimization process. The trajectory after optimization is closer to the ground-truth.

VI. CONCLUSION

Experiments on the challenging BBIL database (room dimension is 10m x 25m) show that the implementation of machine learning in positioning helps us to reduces the computation time of the kNN algorithm when the searching region is narrowed. The computation time is hence decreased by half in estimating pose location from RSSI vector. Furthermore, the analysis in the distribution of k-nearest points will give an effective evaluation for pose confidence. When mobile device is moving continuously, the proposal of signal filtering is also a good method to reduce noise effect. The results indicated on 10 different test sequences of BBIL showed that even before trajectory reconstruction, the accuracy is significantly enhanced where RMSE is reduced from 3.64m to 2.54m when using filtered RSSI signals instead of the raw RSSI signals. Moreover, our proposal of optimizing trajectory based on high confident poses gives an attractive enhancement in accuracy. The RMSE after optimization attains even a lower value at 1.78m. This error is competitive with other tracking algorithms (EN, LSTM or GRU) which require the initial pose of trajectory to be known.

In the future work, we intend to validate our system in other environments with other configurations of RSSI fingerprint. Moreover, we will research the possibility of using Recurrent Neural Network algorithm such as EN or LSTM in the extrapolation step to improve the timing performance of trajectory reconstruction.

REFERENCES

[1] C.-H. Huang, L.-H. Lee, C. C. Ho, L.-L. Wu, and Z.-H. Lai, “Real-time RFID indoor positioning system based on Kalman-filter drift removal and Heron-bilateration location estimation,” IEEE Trans. Instrum. Meas., vol. 64, no. 3, pp. 728–739, Mar. 2015.

[2] T.-H. Dao, M.-T. Le, and Q.-C. Nguyen, “Indoor localization system using passive UHF RFID tag and multi-antennas,” in Proc. Int. Conf. Adv. Technol. Commun. (ATC), Oct. 2014, pp. 405–410.

[3] X. Yu, Q. Li, J. Peña Queralt, J. Heikkonen, and T. Westerlund, “Applications of UWB networks and positioning to autonomous robots and industrial systems,” 2021, arXiv:2103.13488. [Online]. Available: http://arxiv.org/abs/2103.13488.

[4] Ø. Volden, A. Stahl, and T. I. Fossen, “Vision-based positioning system for auto-docking of unmanned surface vehicles (USVs),” Int. J. Intell. Robot. Appl., vol. 2021, pp. 1–18, Aug. 2021.

[5] H. Q. Tran and C. Ha, “Fingerprint-based indoor positioning system using visible light communication—A novel method for multipath reflections,” Electronics, vol. 8, no. 1, p. 63, Jan. 2019.

[6] Z. Y. Dong, W. M. Xu, and H. Zhang, “Research on ZigBee indoor technology positioning based on RSSI,” Proc. Comput. Sci., vol. 154, pp. 424–429, Jun. 2019.

[7] H-H. Lam, C.-C. Cheung, and W.-C. Lee, “RSSI-based Lora localization systems for large-scale indoor and outdoor environments,” IEEE Trans. Veh. Technol., vol. 68, no. 12, pp. 11778–11791, Dec. 2019.

[8] F. Liu, J. Liu, Y. Yin, W. Wang, D. Hu, P. Chen, and Q. Niu, “Survey on WiFi-based indoor positioning techniques,” IET Commun., vol. 14, no. 9, pp. 1372–1383, Jun. 2020.

[9] L. Ruan, L. Zhang, T. Zhou, and Y. Long, “An improved Bluetooth indoor positioning method using dynamic fingerprint window,” Sensors, vol. 20, no. 24, p. 7269, Dec. 2020.

[10] K. Shahala, B. Seetharamulu, and M. A. Jabb, “Machine learning in healthcare: A review,” in Proc. 2nd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA), Mar. 2018, pp. 910–914.

[11] H. Yin and N. K. Jha, “A health decision support system for disease diagnosis based on wearable medical sensors and machine learning ensembles,” IEEE Trans. Multi-Scale Comput. Syst., vol. 3, no. 4, pp. 228–241, Oct./Dec. 2017.

[12] D. E. Ruiz-Guirola, C. A. Rodríguez-López, S. Montejo-Sánchez, R. D. Souza, and M. A. Imran, “DRX-based energy-efficient supervised machine learning algorithm for mobile communication networks,” IET Commun., vol. 15, no. 7, pp. 1000–1013, Apr. 2021.

[13] G. Yang, Y. Zhang, Z. He, J. Wen, Z. Ji, and Y. Li, “Machine-learning-based prediction methods for path loss and delay spread in air-to-ground millimetre-wave channels,” IET Microw., Antennas Propag., vol. 13, no. 8, pp. 1113–1121, Jul. 2019.

[14] A. Sharma, A. Jain, P. Gupta, and V. Chowdary, “Machine learning applications for precision agriculture: A comprehensive review,” IEEE Access, vol. 9, pp. 4843–4873, 2021.

[15] M. N. Akhter, S. Mekhifel, H. Mokhlis, and N. M. Shah, “Review on forecasting of photovoltaic power generation based on machine learning and Metaheuristic techniques,” IET Renew. Power Gener., vol. 13, no. 7, pp. 1009–1023, May 2019.

[16] D. Schweizer, M. Zehnder, H. Wache, H.-F. Witschel, D. Zanatta, and M. Rodriguez, “Using consumer behavior data to reduce energy consumption in smart homes: Applying machine learning to save energy without lowering comfort of inhabitants,” in Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA), Dec. 2015, pp. 1123–1129.

[17] M. A. Bhatti, R. Riaz, S. S. Rizvi, S. Shokat, F. Riaz, and S. J. Kwon, “Outlier detection in indoor localization and Internet of Things (IoT) using machine learning,” J. Commun. Netw., vol. 22, no. 3, pp. 236–243, Jun. 2020.

[18] C.-H. Hsieh, J.-Y. Chen, and B.-H. Nien, “Deep learning-based indoor localization using received signal strength and channel state information,” IEEE Access, vol. 7, pp. 33256–33267, 2019.

[19] B. Jia, B. Huang, H. Gao, and W. Li, “On the dimension reduction of radio maps with a supervised approach,” in Proc. IEEE 42nd Conf. Local Comput. Netw. (LCN), Oct. 2017, pp. 199–202.

[20] A. H. Salamah, M. Tamazin, M. A. Sharaks, and M. Khedr, “An enhanced WiFi indoor localization system based on machine learning,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Oct. 2016, pp. 1–8.

[21] D. V. Le, N. Meratnia, and P. J. M. Havinga, “Unsupervised deep feature learning to reduce the collection of fingerprints for indoor localization using deep belief networks,” in Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN), Sep. 2018, pp. 1–7.

[22] D. Sun, E. Wei, Z. Ma, C. Wu, and S. Xu, “Optimized CNNs to indoor localization through BLE sensors using improved PSO,” Sensors, vol. 21, no. 6, p. 1995, Mar. 2021.

[23] X. Guo, L. Li, N. Ansari, and B. Liao, “Knowledge aided adaptive localization via global fusion profile,” IEEE Internet Things J., vol. 5, no. 2, pp. 1081–1089, Apr. 2018.

[24] M. AlHajri, N. Ali, and R. Shubair, “Indoor localization for IoT using adaptive feature selection: A cascaded machine learning approach,” IEEE Antennas Wireless Propag. Lett., vol. 18, pp. 2306–2310, 2019.

[25] K. Liu, H. Zhang, J. K.-Y. Ng, Y. Xia, L. Feng, V. C. Lee, and A. S. H. Son, “Toward low-overhead fingerprint-based indoor localization via transfer learning: Design, implementation, and evaluation,” IEEE Trans. Ind. Informat., vol. 14, no. 3, pp. 898–908, Mar. 2018.
[26] M. Altini, D. Brunelli, E. Farella, and L. Benini, “Bluetooth indoor localization with multiple neural networks,” in Proc. IEEE 5th Int. Symp. Wireless Pervasive Comput., May 2010, pp. 295–300.
[27] P. Sthapit, H.-S. Gang, and J.-Y. Pyun, “Bluetooth based indoor positioning using machine learning algorithms,” in Proc. IEEE Int. Conf. Consum. Electron., Jun. 2018, pp. 206–212.
[28] I. Alexander and G. P. Kusuma, “Predicting indoor position using Bluetooth low energy and machine learning,” Int. J. Sci. Technol. Res., vol. 8, pp. 1661–1667, Jun. 2019.
[29] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, “Recurrent neural networks for accurate RSSI indoor localization,” IEEE Internet Things J., vol. 6, no. 6, pp. 10639–10651, Sep. 2019.
[30] M. M. Maw, H. M. N. M. Tint, and S. Duangsuwan, “Analysis of indoor wi-fi localization using Gaussian process regression and K-nearest neighbor algorithms,” UTK Res. J., vol. 14, no. 1, pp. 30–39, 2020.
[31] M. Kennedy, P. Spachos, and W. Graham Taylor, “BLE beacon indoor localization dataset,” Univ. Guelph, Guelph, ON, Canada, Tech. Rep., 2019. [Online]. Available: https://dataverse.scholarsportal.info/dataset.xhtml?persistentId=doi:10.5683/SP2/UTZTFT
[32] G. Zhang, P. Wang, H. Chen, and L. Zhang, “Wireless indoor localization using convolutional neural network and Gaussian process regression,” Sensors, vol. 19, no. 11, p. 2508, May 2019.
[33] C. K. Williams and C. E. Rasmussen, Gaussian Processes for Machine Learning, vol. 2. Cambridge, MA, USA: MIT Press, 2006.
[34] Y. Zhao, C. Liu, L. S. Mihaylova, and F. Gunnarsson, “Gaussian processes for RSS fingerprints construction in indoor localization,” in Proc. 21st Int. Conf. Inf. Fusion (FUSION), Jul. 2018, pp. 1377–1384.
[35] X. Wang, X. Wang, S. Mao, J. Zhang, S. C. G. Periaswamy, and J. Patton, “Indoor radio map construction and localization with deep Gaussian processes,” IEEE Internet Things J., vol. 7, no. 11, pp. 11238–11249, Nov. 2020.
[36] X. Peng, R. Chen, K. Yu, F. Ye, and W. Xue, “An improved weighted K-nearest neighbor algorithm for indoor localization,” Electronics, vol. 9, no. 12, p. 2117, Dec. 2020.

**DAI DUONG NGUYEN** received the Electrical Engineering degree in industrial information from the Hanoi University of Science and Technology, Vietnam, in 2014, and the M.S. degree in information system and technology and the Ph.D. degree in robotics from Paris-Sud University, France, in 2015 and 2018, respectively. He is currently a Lecturer with the Department of Instrumentation and Industrial Informatics (3I), School of Electrical Engineering (SEE), Hanoi University of Science and Technology (HUST). His research activities include localization-based RF, visual SLAM, and real-time applications on embedded systems.

**MINH THUY LE** (Member, IEEE) received the Engineer and M.S. degrees in electrical engineering from the Hanoi University of Science and Technology (HUST), 2006 and 2008, respectively, and the Ph.D. degree in optics and radio frequency from the Grenoble Institute of Technology, France, in 2013. She is currently a Lecturer and also a Group Leader of the Radio Frequency Group, Department of Instrumentation and Industrial Informatics (3I), School of Electrical Engineering (SEE), HUST. Her current research interests include built-in antenna, antenna array, beamforming, metamaterials, indoor localization and RF energy harvesting, wireless power transfer, and autonomous wireless sensor.