2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 4-7 October 2016, University of Alcalá, Madrid, Spain.

An Enhanced WiFi Indoor Localization System Based on Machine Learning

Ahmed H. Salamah, Mohamed Tamazin, Maha A. Sharkas, Mohamed Khedr
Department of Electronics and Communications Engineering
College of Engineering and Technology
Arab Academy for Science, Technology and Maritime Transport
Alexandria, Egypt
AhmedHuss.Salamah@aast.edu, Tamazin@gmail.com, MSharkas@aast.edu, Khedr@aast.edu

***Abstract*—The Global Navigation Satellite Systems (GNSS) suffer from accuracy deterioration and outages in dense urban canyons and are almost unavailable for indoor environments. Nowadays, developing indoor positioning systems has become an attractive research topic due to the increasing demands on ubiquitous positioning. WiFi technology has been studied for many years to provide indoor positioning services. The WiFi indoor localization systems based on machine learning approach are widely used in the literature. These systems attempt to find the perfect match between the user fingerprint and pre-defined set of grid points on the radio map. However, Fingerprints are duplicated from available Access Points (APs) and interference, which increase number of matched patterns with the user's fingerprint. In this research, the Principle Component Analysis (PCA) is utilized to improve the performance and to reduce the computation cost of the WiFi indoor localization systems based on machine learning approach. All proposed methods were developed and physically realized on Android-based smart phone using the IEEE 802.11 WLANs. The experimental setup was conducted in a real indoor environment in both static and dynamic modes. The performance of the proposed method was tested using K-Nearest Neighbors, Decision Tree, Random Forest and Support Vector Machine classifiers. The results show that the performance of the proposed method outperforms other indoor localization reported in the literature. The computation time was reduced by 70% when using Random Forest classifier in the static mode and by 33% when using KNN in the dynamic mode.**

Keywords*—WiFi, Indoor positioning, Machine Learning, PCA, fingerprint.

I. Introduction

In the last few decades, ubiquitous positioning is becoming a significant topic that the researchers community are carrying out today. By using ubiquitous positioning in the mobile navigation system, it encourages ubiquitous location service in mobile phones, as they are recognizable personal electronic devices for many users. There is an increasing need for better development using the proposed ubiquitous positioning techniques. Researchers are impaired with challenging frameworks for understanding different options during building ubiquitous positioning system in several environments.

The users always require alternative context of information on the environment of the mobile phone, not only limited to communication services. One of the information contexts is the location of the mobile user. The location services should be available in outdoor and indoor environments. There are many well-known navigation and positioning techniques such as Global Navigation Satellite Systems (GNSS) based navigation, vision-based, odometer-based, map matching, and others [1]. The location services are required in numerous fields such as the medical field, which facilities the location of doctors needed, for better care service to patients. Marketing field as it assists in displaying the advertisements and offers based on location.

In the case of indoor environments, the GNSS are not available due to the blockage of the satellite's signals and the multipath effects. Therefore, an area of investigation, to create alternative solutions to restrain the localization problems in the indoor environments, is becoming a point of interest by many researchers. One of these innovative areas is based on using Signals of Opportunity (SoOP) for positioning [2]. These signals are being originated from the wireless backbone that already exists in place. Furthermore, SoOP overcomes the main GNSS challenges in indoor environments [3, 4].

Different short-range radio frequency technologies such as Radio Frequency Identification (RFID), Bluetooth, Infrared and Wireless Local Area Network (WLAN) are a source of SoOP. Since WLAN infrastructures are available in common places such as home, shopping mall and offices, therefore, this paper focuses on utilizing the WiFi signals of the WLAN for indoor positioning.

The accuracy of the WiFi-based location is based on estimating the distances between the mobile phone and the available Access Points (APs). The Received Signal Strength Indicator (RSSI) is used to calculate the distance that is required for WiFi-based location estimation [5]. Based on the RSSI, there are many techniques developed for localization that are divided into two major categories [6]: signal propagation model and fingerprinting approaches. Applying the path loss propagation model in indoor environments leads to unfeasibility of distance estimation due to the random propagation effects, which require different models to be applied in different regions [7]. Fingerprint approach is widely

978-1-5090-2425-4/16/$31.00 ©2016 IEEE

used in the literature due to its robustness and cost-effective in the indoor environments [8].

Indoor localization systems based on fingerprint approach can be categorized in two classes of processing bases namely; the client base and the server base [9]. The processing bases are categorized according to who is responsible for the applied positioning procedures. Concerning the client base, all the processing is done in the mobile device. This rapidly consumes the lifetime of the battery and the storage of the radio map, which stores the collected RSSI fingerprints from the selected WLAN infrastructure. As for the server base, it extracts the tested data at the online stage and uploads it to the server in order to have an estimated location of the user. Then, the server can send the user its location or the user can monitor his location through the server.

Fingerprinting approach has been widely used in indoor positioning systems based on WiFi technology. It's accuracy based on the accuracy of estimating the signal strength that is collected form the WiFi APs. Several techniques are used in the literature, which are based on deterministic or probabilistic approaches. The main problem is that shadowing, multipath diffraction and scattering could affect the Received Signal Strength (RSS) in the propagation indoor environments. These effects lead to large position errors and position outliers.

The main goals of this paper are to enhance the accuracy of the WiFi indoor localization systems based on machine learning approach and to reduce the required computational cost and time. This research attempts to find the perfect match between user locations from a pre-defined set of grid points. Comparing between the user fingerprint and stored fingerprint is the principle method for positioning, which find the best matching pattern in the radio map. Fingerprints are duplicated from APs and inference of indoor environment at different positions in the radio map, which increase the number of matched patterns with the user's fingerprint. The Principle Component Analysis (PCA) plays the role of handling the latter part in this research, as it identifies the redundancy structure resulting from multivariate APs, in order to compact description. The duplicated fingerprint and noise between APs is reduced which decrease the dependency on a certain APs.

This paper is organized as the following. Section II discuss the related work. Section III and IV introduce the methodologies used in our proposed method followed by an introduction of the proposed system architecture. Section V describes the experimental setups and discuss the outcomes results.

II. RELATED WORK

The Indoor localization systems based on fingerprint approach can be classified into three categories [10]: the deterministic approach, probabilistic approach and machine learning approach. The widely known WiFi RADAR system is the first RF-based localization system using the deterministic approach of the WLAN fingerprint. The WiFi RADAR system stores the RSSI fingerprints at grid points in the offline stage, which create the radio map. The K-nearest neighborhood method is used to find the estimated location for the user's fingerprint [11].

The probabilistic approach is based on computing the probability of each grid point and estimating the user's position using the Bayesian inference [12]. The latter approach is used in the HORUS system, where it achieves a higher accuracy in comparison to the WiFi RADAR system that is based on the deterministic approach [13]. The main drawback of the probabilistic approach is that it requires a large number of samples from APs to create a distribution. This increases the time required to build the radio map and requires large storage size.

The COMPASS system [14] is another probabilistic indoor positioning system. This system is developed to consider the attenuation caused by human body by adding a compass to the system. In the offline stage, the COMPASS system creates multiple radio maps with several selected orientations (typically each 45° or 90°). In the online stage, the user orientation is provided with a digital compass and only the fingerprints with a similar orientation are utilized to estimate the user's location. The main disadvantage of the COMPASS system is that requires a large storage size to save several radio maps with different orientations.

The aforementioned approaches require significant computation cost, which led researchers to consider the machine learning approach to estimate the indoor position. Therefore, this paper focuses on utilizing the WiFi indoor localization system based on machine learning approach.

III. METHODOLOGY

There are several location fingerprinting-based positioning algorithms using pattern recognition techniques. These techniques are implemented into two stages. The first stage, named training stage, where data is collected and provided to the classifier to build a model to classify and predict data properties. The second stage, called testing stage, where new data is tested against the model that was built during the training stage [11, 13-14]. The machine learning algorithms can be used for classification or regressions [15]. In classification, the machine-learning algorithm learns to classify the data in different classes, while the regressions predicts a continuous variable by learning from the training data.

A. K- Nearest Neighbor (KNN)

K-nearest neighbor is a simple method that tries to perform classification by calculating the distance between features [11]. The values of RSSI fingerprints depend on the physical distance between APs used in radio map and mobile phone. The KNN algorithm considers K calibration points. The selection of these points based on selecting the closest K points in the feature space to approximate the position of the user. In the literature [11], it was mentioned that selecting the nearest point is good indication of closeness in the physical space.

The KNN algorithm starts by calculating the *P*-norm of *M*-dimensions RSSI vector (x_i), where x_i belongs to the fingerprint radio map R^N [11].

$$x_i \in R^N \tag{1}$$

The KNN algorithm calculate the distance between the measured $\overline{y}$ and the RSSI of vector of the radio map $\overline{x}_i$ as shown in the following equation.

$$d\left(\overline{y} - \overline{x}_i\right) = \left(\sum_{j=1}^{|\overline{y}|} \left| \overline{y}_j - \overline{x}_{ij} \right|^p \right)^{1/p} \tag{2}$$

where $d(.)$ is the distance measured, $\overline{x}_{ij}$ is the sample average (j indicates the selected (APs)). In case of p = 1 represents Manhattan norm-distance and in case of p = 2 represents using the Euclidean norm-distance. In this paper, the Euclidean distance is used.

The algorithm selects the minimum Euclidian distance of the K-neighbor points. Let P_K is a list of the calibration points coordinates corresponding to K fingerprints in the radio map. Each L contains RSSI readings of x_i from APs, which satisfy:

$$d\left(\overline{y} - \overline{x}_i\right) \le d\left(\overline{y}, \overline{x}_j\right) \tag{3}$$

$$P_K = \{L_1, ..., L_K\} \tag{4}$$

$$\overline{x}_{1:K} = \left\{\overline{x}_1, ..., \overline{x}_K\right\} \tag{5}$$

The location $\hat{L}$ is estimated by calculating the average of the coordinates of the K-nearest neighbor.

$$\hat{L} = \frac{1}{K} \sum_{i=1}^{K} L_i \tag{6}$$

B. Support Vector Machines (SVMs)

Support Vector Machines [15, 16] are powerful technique used for classification and data regression. They are used as a non-parametric supervised classifier for pattern recognition problems. SVMs are used in the localization system by training the support vectors on radio map that consist of grid points. SVMs analyze the relationship between the trained fingerprints and their grid points by considering each grid point as a class. The tested RSSI fingerprints are taken as an input to SVM that predict the class to which the tested belongs. This technique can be generalized to classify between more than two classes for N training data (x_i, y_i).

Before any classification, the RSSI fingerprint vectors are mapped into higher dimensional space using kernel function. The SVM kernel functions $K(.,.)$ is the dot product of two feature vectors x_i and x_j in some expanded feature space, there are several kernels are proposed by researchers. The four basics kernels as follow: linear, polynomial, sigmoid and radial basis function (RBF). In this research, both the linear and RBF are used in the following two forms:

- Linear:

$$K(x_i, x_j) = x_i^T \cdot x_j \tag{7}$$

- RBF or Gaussian:

$$K(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} \tag{8}$$

where σ^2 is the variance (i.e. width) of the Gaussian kernel.

After representing the training data by mapping the data to the feature space. The SVM algorithms identify hyperplane, which separates the support vector trained with a distance equal $2/\|w\|$ [16]. It is constructed in such a way that they can be divided in two data classes with a maximum distance to the closest vector from the same class. The optimization problem is shown in:

$$y_i(w^T x_i + b) - 1 \ge 0 \tag{9}$$

$$\max \frac{2}{\|w\|} \to \min \frac{\|w\|}{2} \to \min \frac{1}{2}\|w\|^2 \tag{10}$$

$$\min\left\{ L_{pd}\left(w, b, \alpha_i\right) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i [y_i(w^T x_i + b) - 1] \right\} \tag{11}$$

where b and α_i are solution of the constrains and y_i is the output of each class $y_i \in \{1, -1\}$, which achieve the minimize (10) based on lagrangian function, where α_i is the lagrange multipliers . The constrained optimization problem can be expressed in a dual form by searching a solution under the form [17]:

$$w = \sum_{i=1}^{N} \alpha_i y_i x_i \qquad \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{12}$$

maximizing with respect to α :

$$\max\left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j^T) \right\} \tag{13}$$

under constrains:

$$\sum_{i=1}^{N} \alpha_i y_i = 0 \text{ where } \alpha_i \ge 0 \quad \forall i \tag{14}$$

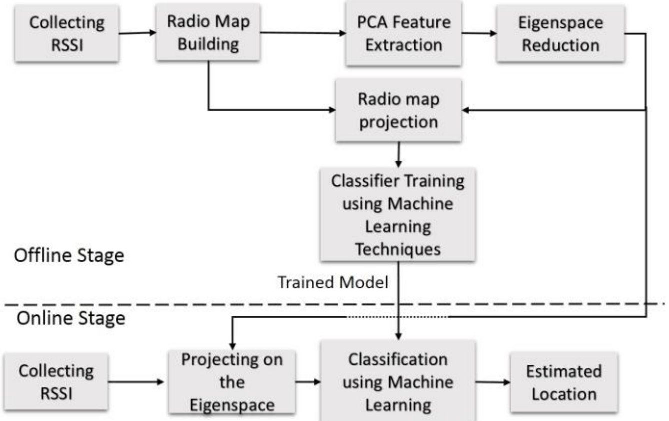

Figure 1: Proposed system architecture

The vectors, that are closest to the maximal margin hyperplane are called, support vectors x_i . The maximal margin hyperplane and support vectors are responsible to identify the tested data class. The measured fingerprint is classified according to the sign of:

$$f(x) = \sum_{i=1}^{M} \alpha_i y_i K(x_i, x) + b \qquad (15)$$

where α_i and b are obtained from the optimization problem and x_i is the class label and fingerprint of the dataset entry i for M AP*s* . The grids in the radio map turns to multiclass classification problems into a combination of two-classes, which are labeled as binary (i.e. pairwise). This can be applied in two ways one-against-all and one-against-one [15].

The one-against-all is constructed by dividing the multiclass problem into a group of pair classification problems. The problem with *n* classes is divided into *n* binary classifier. Each binary classifier is responsible to separate one class from all classes. Each *n* classifier is trained separately, to find the estimated location. The output the outputs of all *n* classifier is first calculated. The largest value of $f(x)$ (15) is the predicted class.

C. Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for regression and classification [18]. The DTs is algorithm is based on a binary decision tree that is constructed from the training data set. The algorithm starts by learning simple decision rules inferred from the data features. It is constructed in three nodes; root node, internal node and leaf or terminal nodes. The root node has no incoming edges and zero or more outgoing edges. The internal has exactly one incoming edge and two or more outgoing edges. The leaf has one incoming edge and no outgoing edges. Every leaf node in the decision tree is assigned as a class label. Each of these nodes is corresponds to a decision made on one of the input parameters. The node is then divided into new subsets, one for each of the node's sub-trees, in such a way that the same target location is in the same subsets [18]. The algorithm stops when there is a pure decision and uncertainty is inefficient. A pure decision means each node's data subset contains one and only one target location.

The node's data subset splitting rule is the Gini diversity index:

$$Gini(t) = \sum_{i=1}^{N} p(i/t)(1 - p(i/t)) = 1 - \sum_{i=1}^{N} [p(i/t)]^2 \qquad (16)$$

where $p(i/t)$ is the probability of class *i* occurring at a certain node in the tree.

D. Random Forest

Random forest (RF) is a classifier in which many decision trees are generated. The RF chooses the tree which has the highest votes after their classification results. The most occurring class number in the output of the decision trees is the final output of the RF classifier. The random forest classifier allows the estimation of the importance of each feature in those classification results [19]. A recursive process in which the input dataset is composed of smaller subsets allows the training of each decision tree. This process continues until all the tree nodes reach to the similar output targets. The random forest classifier takes weights based on the input as parameter that resembles the number of the decision trees. Those weights will be formed in the collaborative forest classifier without the conventional tree pruning process [19].

IV. Proposed Method

The proposed system is composed of two main stages. The offline stage and the online stage, as shown in Figure 1.

In the offline stage, the RSSI fingerprints, which are collected from numerous grid points, is used to build the radio map. The radio map was built by saving the RSSI fingerprint at each grid points. Each grid point defined by 2 dimensional coordinates. In order to reduce the high correlation between the grid points and it's adjacent point, the PCA is proposed to generate uncorrelated space. The PCA [20, 21] is a powerful technique, which is used for feature extractions and as a dimensional reduction method. The main two advantages of proposing the PCA algorithm in the offline stage are that (1) the PCA extracts the important information from the pre-defined radio map; (2) the PCA reduces the multivariate data matrix without losing much information in which observations are described by several inter-correlated quantitative dependent variables.

The PCA algorithm deals with the high dimensionality pre-defined radio map by linearly combining the features into an uncorrelated space (i.e. eigenspace) by using training covariance matrix of the pre-defined radio map of size *N* x *N*. The radio map matrix is projected into the uncorrelated space in the direction of the largest variance. The selection of Principle Components (PCs) is based on the highest variance (i.e. eigenvalue). The eigenvalue represents the information context of PCs.

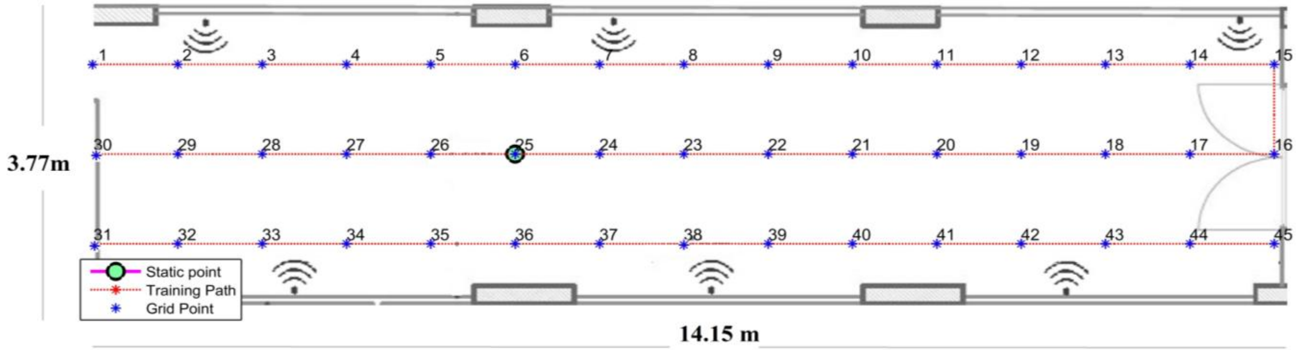

Figure 2. Layout of the residence where the experiments was conducted

The PCA eigenspace is created based on a set of M RSSI readings per location of vector x_i of size M × 1 as column vector, where M is smaller than N.

This Eigenspace is characterized by the corresponding mean, where N is the number of training samples:

$$\bar{x} = \sum_{i=1}^{N} \frac{x_i}{N} \tag{17}$$

$$C_r = \sum_{i=1}^{N} \left(x_i - \bar{x}\right)\left(x_i - \bar{x}\right)^T = X_i X_i^T \tag{18}$$

where C_r is the covariance matrix, computed from the set of N experiments using

Let the eigenvalues $\{\lambda_1, \lambda_2, \ldots\ldots, \lambda_N\}$ of C_r that are arranged in a descending order with corresponding normalized eigenvectors $\{V_1, \ldots\ldots, V_N\}$ as follows:

$$C_r V_i = \lambda_i V_i \tag{19}$$

$$\lambda_1 \geq \lambda_2 \geq \ldots\ldots \geq \lambda_N \tag{20}$$

The eigenvectors $\{V_1, \ldots\ldots, V_N\}$ are geometrically orthonormal and statistically uncorrelated.

Before projecting the radio map on the eigenspace, it will be reduced by selecting PCs. The selection of PCs are based on information context of each PC and the error which each component has achieved. The analysis on selecting the minimum number of PCs will be discussed further more in the following section. Then, the uncorrelated new radio map Z_i is calculated by projecting X_i on the reduced eigenspace W, which represent a linear combination of the eigenvectors, where Z_i will represent the new radio map.

$$W = X \times V \tag{21}$$

$$Z_i = W^T \times X_i \quad \forall i \tag{22}$$

Several classifiers will be trained using the new radio map Z_i generating the training model, which it have been reduced after selecting PCs from eigenspace, where the important information contexts are extracted from the radio map.

While in the online stage, the tested RSSI fingerprints, that are collected, are then projected on the eigenspace W , which was selected on the offline stage. The projected values from the tested RSSI fingerprints are compared with the trained model to find the best match between them using the proposed classifiers. The physical location of the model which has the best match in the new radio map will be labelled as the estimated location.

V. REAL EXPERIMENTS AND RESULTS

A. *Real Experimental Setup*

In this research, an android application was developed on a mobile device – Galaxy Note 10.1. The application was developed to collect RSSI measurements form the available APs. In this paper, a server processing based approach was utilized in both training and testing stages. Thus, a server was established to upload the collected RSSI measurements using the developed application. The collected RSSI measurements are used to build the radio map. The experiment was conducted in a hall of a typical apartment located in the 12th floor of a building in Alexandria, Egypt. The hall consists of typical furniture. There are random interfering motion was added to the indoor environment. The layout of the hall is shown in Figure 2. The 2D-dimension of the hall is length 14.15m × width 3.77m. There were six available 802.11 WiFi APs located in the hall distributed as shown in Figure 2.

The data was collected during the month of February 2016. Forty Five grid points were defined horizontally and vertically. The distance between the horizontal grid points 1 m. However, the distance between the vertical grid points is 1.2 m. All RSSI measurements were collected using the one mobile device, which is running the developed android application.

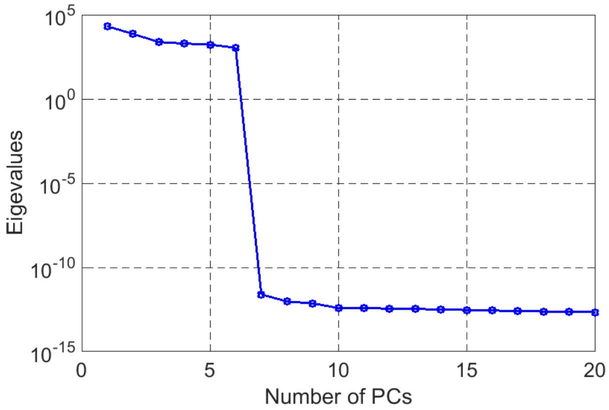

Figure 3. The first 15 eigenvalues (log scale)

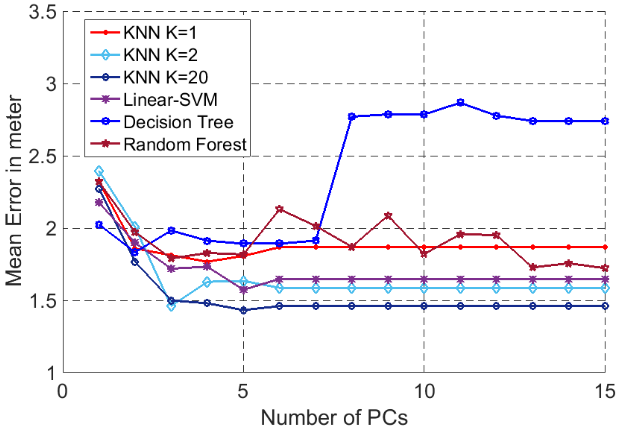

Figure 4. The first 15 principle component with mean error with classifiers

For the offline stage, A 20 seconds RSSI measurements was collected at each grip point's location. In order to build the radio map, three training sets were collected by walking in a regular walk for 2 different days.

For the testing stage, both static and dynamic experiments were made inside the hall. The static experiment duration was 30 minutes. The static data was collected in the location of the grid point labeled by 25 as shown in Figure 2. The RSSI collecting rate was 9 samples every 20 seconds. The dynamic experiment was done inside the hall. The trajectory of the dynamic mode is a regular walk through all 45 grid points, as shown in red color in Figure 2, for a duration of 15 minutes.

B.Results

In this section, it started with important parameter which is choosing minimum number of PCs. The increasing of the number of PCs leads to more information added, whereas more noise and duplicated information were gained. The duplication and noise would confuse the training model and degrade the performance of the classifiers. Then, the performance evaluation of the classifiers on the selected PCs. The performance of the proposed method was tested using the KNN algorithm at K equal 1 and 2, which usually used for comparison [11]. But after iterative experiments, it was found that at K of value equal to 20 the KNN has better performance in return the time complexity was increased. As for the RF, the selected DTs to train are 15, which were also selected after iterative error calculations.

1) Analysis on Selecting Minmum Number of PCs

The number of PCs were studied to minimize the computation complexity and decrease the mean error. The minimum number of PCs in this paper was studied through three methodologies, that all lead to the same selected number of PCs, where (1) the importance of information context of each PC is evaluated through the value of the eigenvalue, which represent the variance of the data around the PC. As the eigenvalue of the corresponding PC increase, it tends to contain more valuable information; (2) the mean error is evaluated at different number of PCs to choose the PCs that have minimum mean error, which will not be highly affected by noise and duplication; (3) the rule of selecting the number of PCs are based on the cumulative percentage of eigenvalues.

After applying the PCA algorithm, the resulting eigenspace from the covariance matrix is *N*×*N*. If the collected radio map was projected on the resulting eigenspace directly, then the computational complexity will be high and hence, the size of the radio map will increase.

The choice of number of PCs ***U*** is an important parameter in the proposed method. Previous work has empirically reported that ***U*** could be around 7 to 10 [21]. On the other hand, the proposed method represents an unbiased mechanism to select the minimum ***U*** required, since after the PCA transformation. The information quantified will be satisfied with the selected number of ***U***.

The first method, which discusses the importance of information in the data is by choosing the PC which has high eigenvalues, where the eigenvalue is an indication of the variance for the corresponding PC. The high eigenvalue (i.e. variance) for PC mean that the PC holds high valuable information for the data, which indicates its importance. As shown in figure 3, the first 6 components contains the highest eigenvalues compared to the other principle components. The eigenvalues of the 4th, 5th and 6th PC are relatively equal, which is an indication of having an equal information context that can be neglected.

The second method is obtained from figure 4, where the classifiers were compared according to the mean error with the first 15 PCs. It was concluded that at the first 3 PCs the mean error was decreasing to minimum. According to previous figures description, it was deduced that using the first 3 PCs will reduce the computational complexity of the system and the mean error.

The last method is the rule of selecting the number of principle components are based on the cumulative percentage of eigenvalues as follows [20]:

$$(\sum_{i=1}^{U}\lambda_i)/(\sum_{i=1}^{N}\lambda_i)>\eta \tag{23}$$

where $\{\lambda_1,\lambda_2,.....,\lambda_N\}$ are the eigenvalues, $(\sum_{i=1}^{U}\lambda_i)/(\sum_{i=1}^{N}\lambda_i)$ represents the percentage which retains

Table 1. Calculating different metrics for the used classifiers at the static mode experiment and the dynamic experiment.

PCs		Static Mode						Absolute Time Complexity	Dynamic Mode						Absolute Time Complexity
Classifier		Without PCA			3rd PC				Without PCA			3rd PC			
		Mean	Variance	RMSE	Mean	Variance	RMSE		Mean	Variance	RMSE	Mean	Variance	RMSE	
KNN K = 1		3.12	≈ 0	2.29	3.12	0.011	3.2	31.6%	2.390	2.45	6	2.17	1.16	3.2	7.77%
KNN K = 2		3.12	≈ 0	2.29	3.12	≈ 0	2.29	65.3%	2.054	1.79	6	1.71	1.43	3.2	33.52%
KNN K = 20		3.12	≈ 0	2.29	3.12	0.014	2.29	20.3%	1.97	2.33	4.58	1.77	1.49	4	31.58%
Decision Tree		2.87	0.2517	3	2.96	0.326	2.29	65%	2.19	1.76	4	2.27	2.81	6.41	27.92%
Random Forest		3.1	0.022	2.29	1	0	1	72%	2.65	3.49	8.08	2	2.02	5.124	31.08%
SVM	Linear Kernel	2	0.091	2.29	3.12	0.091	2.29	-104%	1.77	1.54	5.48	2	1.38	4	-255%
	Gaussian Kernel	3.12	0	2.29	4.12	≈ 0	3.2	48%	4.03	7.23	9.07	6.62	15.57	12.21	8.44%

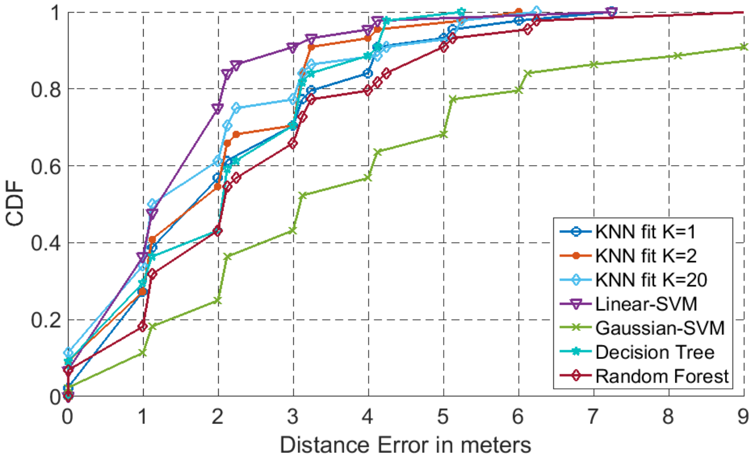

Figure 5. Cumulative error distribution function without PCA for a dynamic mode

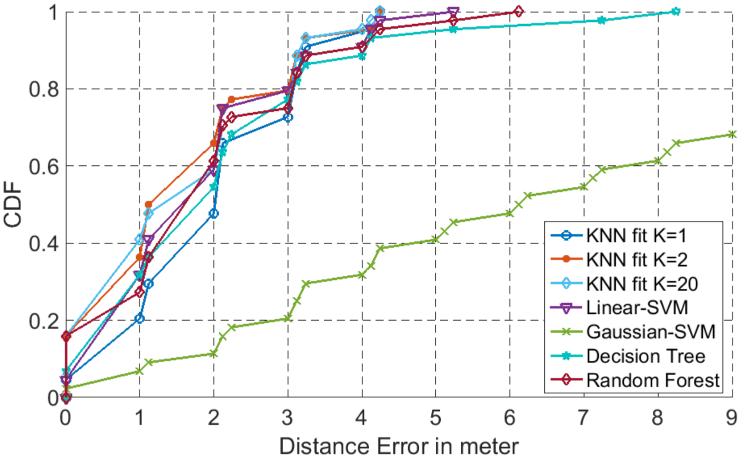

Figure 6. Cumulative error distribution function with PCA for a dynamic mode

information with its' information context, η is the cut-off threshold and $\boldsymbol{U}$ is the number of selected PCs. The selected threshold value is adjusted to be 85% of the total information. When $\boldsymbol{U}$ is calculated at 2, the result is 79.64% and then calculated at 3, with a result of 86.47%. The selected PC is the integer number where the cumulative percentage exceeds the cut-off threshold. Therefore, the selected PC is at $\boldsymbol{U}$ equal to 3.

2) Performance Evaluation

The performance evaluation of the proposed system was tested by calculating different metrics on the estimated location, which are mean, variance and root mean square error (RMSE). In addition, the absolute time complexity was evaluated by calculating the reduced computational time. The reduced time is produced due to the using of the new projected radio map on the selected PCs. The cumulative error distribution function (CDF) was used to compare between the performances of the classifiers.

A comparison between the performances of the proposed method is carried out by using different classifiers, where two real experiments were conducted, which are in static mode and dynamic mode. For further verification, the RMSE as a position accuracy and absolute time complexity were used for testing several classifier. Table 1 shows the results in both static and dynamic modes for different classifiers with and without using the PCA algorithm.

In static mode, It was noticed that adding the PCA with KNN classifier reduces the absolute time complexity (as an example at KNN K=2, it reduces by 65.3%). However, the position accuracy remains the same. In case of using the PCA with decision tree and random forest, both the position accuracy and absolute time complexity are improved (as an example for random forest, it reduces by 1 m and 72%, respectively). In case of using the PCA with linear kernel SVM, it was observed that using the PCA method increases the time complexity. However, the Gaussian kernel SVM reduces the absolute time complexity by 48%.

In dynamic mode, It was noticed that adding the PCA with the KNN classifier improve the position accuracy and absolute time complexity (as an example at KNN K = 2, it is reduces by 3.2 m and 33.52%, respectively). In case of using the PCA with decision tree the absolute time complexity is reduced consequently. As for the random forest both position accuracy and absolute time complexity are improved by 5.12 m and 31%, respectively. By using linear kernel SVM with PCA, the position accuracy increased to 4 m on the expense of the increase of the absolute time complexity. However, case of using in the Gaussian kernel SVM the reduction in the time complexity by 8.44% didn't enhance the position accuracy.

Figures 5 and 6, show the classifiers performance with and without using PCA in term of CDF. The error within 3 meters in dynamic mode of linear SVM, KNN at K = 2, decision tree and random forest with PCA are 79%, 79%, 77% and 75%, respectively, the error without PCA are 90%, 70% , 70% and 65%, respectively. Table 2 shows the error distance equivalent to the percentage of cumulated error. The precision is identified according to the minimum error distance at 100% of cumulated error.

Table 2. The CDF error at 100%

Classifiers	Without PCA	With PCA	Enhanced
K = 1	7.24 m	4.24 m	41.44%
K = 2	6 m	4.24 m	29.33%
K = 20	6.24 m	4.24 m	32.05%
Linear SVM	7.24 m	5.24 m	27.62%
Random Forest	9.12 m	6.12 m	32.9%

In table 2, show numerical results that our approach reduces in terms of CDF when using KNN, linear SVM and random forest classifiers with using PCA. However, the other classifiers did not accept this transformation of the data into uncorrelated space done by PCA, which opposes their assumption of having Gaussian distributed data. The decision tree depends on the probability of occurrence of the data while the PCA removes the duplication in the RSSI fingerprints between the grid points. In addition to, the Gaussian kernel in SVM makes another transformation on the assumed Gaussian distributed data, allowing it to be more fit on the Gaussian space. The PCA removes the Gaussian relation between RSSI fingerprints in the same grid point. This result decreases the performance when using the Gaussian Kernel along with the PCA transformation.

VI. Conculsion

A proposed method is introduced in this paper to enhance the performance of the WiFi indoor localization system. The Principle Components Analysis (PCA) technique is utilized to extract the important information from the pre-defined radio map which reduces the multivariate data matrix without losing important information. Both static and dynamic experiments were conducted in real indoor environments. The performance of the proposed method was evaluated using several machine learning techniques. The results showed that the proposed method reduced the computational complexity by 70% when using Random Forest classifier in the static mode and by 33% when using KNN. It was noticed that the position accuracy was improved in case of using KNN and the Random Forest classifiers.

As for the Gaussian SVM and the Decision tree, they do not perform well in the uncorrelated space and so it is preferred to use the correlated space with them. Furthermore, a graceful balance between positioning accuracy and system cost can be obtained by theoretically choosing a sufficient number of PCs.

VII. References

[1] A. Farrell, "Aided Navigation, GPS with High-Rate Sensors, " McGraw Hill, 2008.

[2] Kenneth A. Fisher, "The Navigation Potential of Signals of Opportunity-Based Time Difference of Arrival Measurements, " Ph.D. Dissertation, Air Force Institute of Technology, 2005.

[3] P. Misra, and P. Enge, "Global Positioning System, Signals, Measurements, and Performance, " Ganga-Jamuna Press, 2011.

[4] K. A. Fisher, "The navigation potential of signals of opportunity-based time difference of arrival measurements, " Ph.D. Dissertation, 2004.

[5] Raida Al Alawi, "RSSI Based Location Estimation in Wireless Sensors Networks, " in Processeding of the 17th IEEE International Conference on Networks, pp.118 122, 2011.

[6] G. V. Zaruba, M. Huber, F. A. Kamangar and I. Chlamtac, "Monte Carlo sampling based in-home location tracking with minimal RF infrastructure requirements, " in Proceeding of the IEEE Global Telecommunications Conference, Piscataway, USA, 2004.

[7] A. Harter, and A. Hopper, "A new location technique for the active office, " in Proceeding of IEEE Personal Communications, 43–47, 1997.

[8] M. B. Kjærgaard, "A Taxonomy for Radio Location Fingerprinting, " in Proceeding of the Third International Symposium on Location and Context Awareness (LoCA 2007), pp. 139-156, 2007.

[9] A. Krishnakumar and P. Krishnan, "The Theory and Practice of Signal Strength-Based Location Estimation, " in Proceeding of Conf. Collaborative Computing: Networking, 2005.

[10] Z. Tian, X. Tang, M. Zhou, and Z. Tan, "Fingerprint indoor positioning algorithm based on affinity propagation clustering, " EURASIP Journal on Wireless Communications and Networking, article 272, 2013.

[11] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system, " in Proceedings of IEEE INFOCOM, pp. 775-784 , 2000.

[12] H. Zang, F. Baccelli, and J. Bolot, "Bayesian inference for localization in cellular networks, " in Proceeding of IEEE INFOCOM (San Diego, CA), pp. 1–9, 2010.

[13] M. Youssef and A. Agrawala, "The Horus location determination system, " Springer Wireless Networks, 2008.

[14] T. King, S. Kopf, T. Haenselmann, C. Lubberger and W. Effelsberg, "COMPASS: A probabilistic indoor positioning system based on 802.11 and Digital Compasses, " in Proceeding of WINTECH, 2006.

[15] Schölkopf B. and Smola A., "A. Learning with Kernels, " MIT Press, Cambridge, MA, 2002.

[16] T Joachims, "Transductive inference for text classification using support vector machines, " in Proceeding of International Conference on Machine Learning (ICML), pp. 200–209, 1999.

[17] M. Brunato and R. Battiti, "Statistical learning theory for location fingerprinting in wireless LANs, " Comput. Netw. 47, 825–845, 2005.

[18] Breiman L., Friedman J., Olshen R., and Stone C., "Classification and Regression Trees, " Wadsworth Int. Group, 1984.

[19] Breiman, L. Random forests. Machine Learning, 45, 5–32, 2001.

[20] H. S. Fang and T. Lin, "Principal component localization in indoor WLAN environments, " IEEE Trans. Mobile Comput., vol. 11, no. 1, pp. 100-110, 2012.

[21] T. King, T. Haenselmann, and W. Effelsberg, "Deployment, Calibration, and Measurement Factors for Position Errors in 802.11 Based Indoor Positioning Systems, " in Proceeding of Conf. Location and Context Awareness, pp. 17-34, 2007.