Hindawi
Mobile Information Systems
Volume 2018, Article ID 3287810, 11 pages
https://doi.org/10.1155/2018/3287810

Research Article

CEnsLoc: Infrastructure-Less Indoor Localization Methodology Using GMM Clustering-Based Classification Ensembles

Beenish Ayesha Akram,[1] **Ali Hammad Akbar**,[1] **and Ki-Hyung Kim**[2]

[1] *Department of Computer Science and Engineering, University of Engineering and Technology, Lahore, Pakistan*
[2] *Department of Computer Engineering, Graduate School, Ajou University, Suwon, Republic of Korea*

Correspondence should be addressed to Beenish Ayesha Akram; beenish.ayesha.akram@uet.edu.pk

Received 7 June 2018; Accepted 19 August 2018; Published 1 October 2018

Academic Editor: Subramaniam Ganesan

Indoor localization has continued to garner interest over the last decade or so, due to the fact that its realization remains a challenge. Fingerprinting-based systems are exciting because these embody signal propagation-related information intrinsically as compared to radio propagation models. Wi-Fi (an RF technology) is best suited for indoor localization because it is so widely deployed that literally, no additional infrastructure is required. Since location-based services depend on the fingerprints acquired through the underlying technology, smart mechanisms such as machine learning are increasingly being incorporated to extract intelligible information. We propose CEnsLoc, a new easy to train-and-deploy Wi-Fi localization methodology established on GMM clustering and Random Forest Ensembles (RFEs). Principal component analysis was applied for dimension reduction of raw data. Conducted experimentation demonstrates that it provides 97% accuracy for room prediction. However, artificial neural networks, k-nearest neighbors, K^*, FURIA, and DeepLearning4J-based localization solutions provided mean 85%, 91%, 90%, 92%, and 73% accuracy on our collected real-world dataset, respectively. It delivers high room-level accuracy with negligible response time, making it viable and befitted for real-time applications.

1. Introduction

Positioning systems aka localization systems both for outside and inside buildings is an ever-exciting area of research and development due to increasing market shares as in smart buildings, assistive and assisted living, safer metropolitans using geographical information systems, and tracking of IoT objects for commercial purposes. User localization is poised to reach 2.6 billion dollars' worth of market share soon [1], specially involving indoor localization solutions. Localization for the outdoors has been successfully commercialized in the form of satellite-based technologies such as GPS, BeiDou, GLONASS, COMPASS, and GALILEO [2]. Indoor positioning cannot be performed using the same technologies because of No-line-of-sight (NLOS) and occlusion. Radio frequency (RF) signals, on the contrary, do not require explicit LOS for operation.

A received signal strength indicator (RSSI) as a measure of RF signal quality for Wi-Fi, RFID, Bluetooth [3], ZigBee [4] and ultrawideband [5] has been used in several indoor positioning systems. Moreover, several kinds of sensory input such as images [6], video, ambient sound [7], accelerometer [8], magnetometer [9], pedometer, gyroscope readings, and their amalgamation with various aforesaid RF signals [10] have also been explored.

Several approaches and their combinations such as time of arrival (TOA), time difference of arrival (TDOA), pedestrian dead reckoning (PDR), and angle of arrival (AOA) [11] have also been utilized for indoors. Each of these has been shown to have serious limitations. For instance, at successive predictions, PDR suffers from error propagation, and precise clock synchronization between sender and receiver is a requirement for TOA- and TDOA-based systems. In addition, specialized antennas are needed for AOA-based systems.

RSSI-based systems are based on two widely adopted mechanisms; location estimation using formalized propagation models or fingerprints. Location systems that are based on the former suffer from low precision at run time because of variability in channel behavior including fading and shadowing, and also due to heterogeneity of device types and form factors [12].

Wi-Fi networks are deployed as in Access Points (APs) that are prevalent everywhere. Utilizing these to capture RSSI fingerprint (FP) is conveniently possible with device as simple as smartphones or phablets. Wi-Fi fingerprint-based localization has the following benefits: no requirement of extra hardware at both sender and receiver sides; utilization of already existent infrastructure; easily implementable; and no essential need of propagation model building which may or may not depict real signal propagation at run time [13].

The infrastructure of APs allows us to collect a dataset comprising FPs on selected Reference Points (RPs) that essentially becomes a cue to the physical layout of the building, like a map. It is then utilized to prepare the localization system as in the training phase. Once trained, the system is ready to be used, i.e., for an unseen FP captured anew, a room number or an associated label is returned by the system to estimate the location.

In this paper, a new localization methodology is presented that uses a combination of data reduction technique as in principal component analysis (PCA), soft clustering technique such as Gaussian mixture model (GMM), and bootstrapped aggregated/bagged ensembles of decision trees commonly referred to as Random Forest Ensembles (RFEs). We aspire to provide infrastructure-less indoor localization methodology which is scalable, easy to deploy, and provides real-time response for high room-level accuracy instead of explicit coordinates. First of all, PCA was employed for raw data dimension reduction. Then, clustering was performed to split the data in similar groups to help classifier better learn the data dynamics. Finally, a separate RFE is trained for every single cluster.

The remainder of the paper has been organized as follows. Section 2 presents related work. Preliminary experimentation results are summarized in Section 3. Section 4 provides details of the localization methodology that we have proposed. Section 5 delves into experimental setup, layout and results for validation. Conclusion along with possible future directions is showcased in Section 6.

1.1. Scope of the Study.

It is important to declare the scope of the study here to give a prelude to the paper that follows. The paper is aimed at proposing a new methodology that is put to application in a practical and particular environment. The resulting constraints and limitations of our experimental regime are quite natural and fairly generalizable in terms of spatiotemporal aspects, specifically for building size and type, fingerprints' collecting device type, and time of the year. The fingerprints were collected at the Software Engineering Centre of our own university. The building is double-storey and has architectural diversity in terms of rooms, corridors, and an inlaying garden. Nonetheless, an extensive dataset spanned over multiple buildings can be obtained to ensure wider spatial diversity. Our dataset was collected using a single Android phone; however, with a large team using a variety of devices, data collection can be performed over different times of the year to obtain data both for training and testing of our approach.

2. Related Work

We summarize here the work on IPS based on Wi-Fi that is typically a WLAN standard IEEE802.11 (b, a, g, ac, or any) or a combination of Wi-Fi with another wireless or sensory input. RADAR [14] from Microsoft® labs is the pioneer research work to employ Wi-Fi signals. Wi-Fi signals received at the base stations (Access Points) from a laptop were used for predicting the user's coordinates using k-NN-based method and triangulation, reporting a median error of 2-3 m. Li et al. [12] combined affinity propagation as a message passing-based clustering algorithm with PSA-based artificial neural network (ANN) for the Cartesian coordinates-based prediction. A mean error as low as 1.89 m was reported by them including 2.9 m for 90% of estimates.

Song et al. [13] analyzed FP collection as an AP relevancy problem. Hidden Naïve Bayes (HNB) was used as a mechanism to infer the most relevant APs and suggested that redundant APs may be obviated for each RP through a variant of ReliefF with the Pearson product-moment correlation coefficient (PPMCC). Moreover, clustering was performed on RPs. One HNB was trained per cluster to approximate user coordinates. Cooper et al. [15] employed combination of Wi-Fi and Bluetooth low energy radio signal FPs based on boosting technique targeting the room-level prediction. They trained one classifier per room based on a variant of AdaBoost that conveniently harnessed decision stumps in one-vs-all notion. Using combination of Wi-Fi + BLE, they acquired 96% accuracy with $4.3E-03$ seconds response time. Wang et al. [16], similar to Li et al., presented training of ANN with back propagation that was based on PSA for RSSI measurements of RFID tags. They performed data preprocessing by normalizing dataset to [0, 1] range along with using Gaussian filter. They estimated x and y coordinates reporting a mean error of 0.34 m.

Xu et al. [17] utilized multilayer neural network (MLNN) for Wi-Fi signals along with network boosting. They trained the MLNN in two stages commonly followed in deep learning, namely, pretraining using autoencoders and fine tuning using back propagation algorithm reporting a mean error of 1.09 m. Zhang et al. [18] proposed a coarse localizer composed of four-layered deep neural network using stacked denoising autoencoders, succeeded by the hidden Markov model-based fine localizer, reporting a mean error of 0.39 m.

Calderoni et al. [19] utilized RFID tags' RSSI values targeting room-level accuracy in a hospital environment. They divided the total area into macroregions using k-means variant, followed by a Random Forest trained per macroregion. Multiple random forests for whom the cluster matching score was greater than a particular threshold determined the final prediction with 83% reported accuracy. Jedari et al. [20] investigated room-level prediction using k-NN, rule-based JRip, and Random Forest classifier based on Wi-Fi signals. They concluded that Random Forest produced much better results than k-NN (77.4%) and JRip (72.2%) with 91.3% accuracy.

Mo et al. [21] proposed the usage of kernel PCA (KPCA) algorithm for the coarse-level prediction of manually labelled cluster using Random Forest. They derived trained matrices from extracted KPCA features and prepared sub-radio maps. For prediction, the features extracted from coarse positioning, refined by the trained KPCA matrices were fed to weighted k-NN (WK-NN) for final coordinates estimation. They reported an accuracy of 93% with an error distance of 2 m. Górak et al. [22] employed Random Forest for finding important APs and applied threshold-based elimination. They determined malfunctioning APs during operation based on important APs. They were able to report error rates as low as 4% for detecting the floor, and as for horizontal detection, 2 m error was reported. The results were compared against 30% and 7 m, respectively, when malfunctioning AP were undetected. They [23] divided FP dataset both into subsets which were either overlapping and/or nonoverlapping as per the presence of RSSI from every AP. Furthermore, one Random Forest was trained for each such subset. They compared the results with base Random Forest, signifying around 5% to 9% betterment in average reported error at the floor level. The performance in terms of detection of floors was unchanged.

Aforementioned are some recent efforts in field of indoor localization using RF signals. k-NN works by storing all the data samples along with marked ground truth labels. An unseen sample is compared with complete dataset based on a similarity measure/distance to determine nearest neighbors, where k is the number of neighbors and neighbors' weightage. The final decision of the sample's class is based on the majority of the labels of the k-nearest neighbors. Several IPS such as in Oussalah et al. [11] and Niu et al. [24] based on k-NN and its variants do not scale well when the dataset grows because they require FP matching with whole dataset. Moreover, recently artificial neural networks and deep learning have gained a great deal of focus for indoor localization. Artificial neural networks try to mimic human brain. They form multiple layers of neurons, namely, a single input layer, a single output layer, and one or more hidden layers. At every layer, several neurons are connected to one another according to a specific configuration and triggering function. Every layer affects and triggers neurons in the next layer following rules of the learning function which eventually evolves into the final output. Heavy resource utilization is required during the training phase; however, their response time is negligible due to minimal required computation. IPS employing ANNs and deep learning such as Li et al. [12], Ding et al. [25], León et al. [26], Zhang et al. [18] and Tuncer and Tuncer [27] faces the challenge of finding several tunable parameters such as the optimal architecture, no. of layers, learning function, and no. of neurons at every layer. Moreover, the convergence rate and the final accuracy of various configurations do not follow any specific trend. Sometimes, a 2-layer simpler configuration takes more time to converge than a 4-layer network, and the accuracy achieved by a 6-layer network is lower than the accuracy obtained by a 3-layer ANN. Hence, heuristics are predominantly used for proposing an architecture based on ANNs because Monte Carlo configuration testing is impossible. An AP location change, or the addition/removal of new APs, will lead to essential retraining of the IPS putting ANN and deep learning at a disadvantage.

Inspired from existing works, we suggest a clustering-based multiclass classifier approach for room-level prediction which is easy to train-and-deploy in terms of computational complexity, provides suitable accuracy, and offers response time appropriate for real-time applications. Clustering follows the divide and conquer approach to help the classifier better learn the group of similar observations instead of the whole dataset. We perform soft classification (clustering) of FPs, not for RPs which has been mostly done in the existing works. PCA-based data dimension reduction helps us to reduce the response time of the system by decreasing the number of predictors.

This approach scales well in terms of response time with an increase in the number of rooms since a single classifier is invoked for each location providing maximum accuracy reported so far to the best of our knowledge.

3. Preliminary Experiments

Preliminary experiments were performed on sample dataset collected from our departmental building to evaluate classifier suitability [28]. The results presented here are on the sample dataset. The detailed experimentation results on the complete dataset of all locations in building are presented in Section 5.

3.1. Dataset Acquisition. A customized app was developed for an Android phone, built to record RSSI as vector data coming from Wi-Fi APs. The Wi-Fi FPs of all observable APs both within 2.4 and 5 GHz bands at a RP were scanned using a commercial off-the-shelf Samsung phone (Version: J5 Galaxy). FPs were obtained at each RP while hovering the smart phone starting at 0° up to right angle (90°) with respect to the floor as shown in Figure 1, so as to make the phone face N, NW, W, SW, S, SE, E, and NE with an effort to keep occlusion as minimum as possible due to the human torso [5, 9]. The user stood at the centre of the RP, held the phone used for FP collection in his hand, and captured multiple FPs in each direction out of the total 8 directions shown in Figure 1. A total of A APs were present in the premises continuously emitting radio signals. These FPs were then stored into DB, each with respective room labels.

3.2. Preprocessing. The resultant dataset was found to be sparsely populated with the identifiers of APs because of the presence/absence of these APs at various RPs labelled in rooms and in the corridors of building. The measured RSSI values varied between −98 dBm and −15 dBm (from being weak to being strong as a result of distance from APs). Being consistent with the well-known practice to keep the missing values slightly weaker than the weakest signal detected in the dataset [12, 14, 18, 19], the missing values were replaced with −100 dBm.

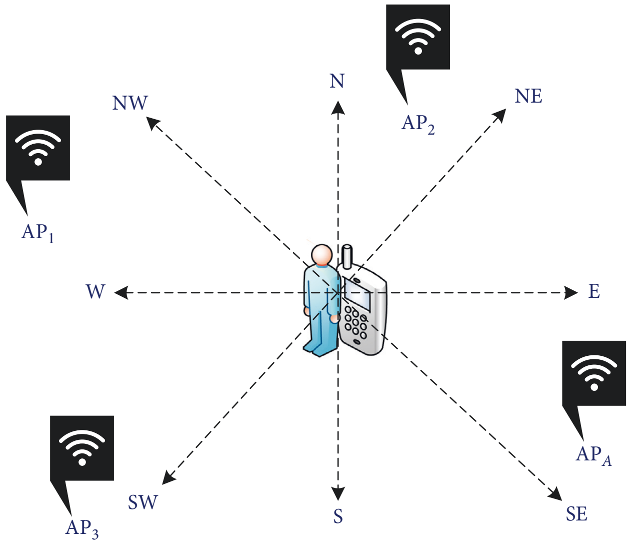


Figure 1: User orientation during data collection.

3.3. Classifier Evaluation. We evaluated performance of 60+ classifiers in WEKA for room-level prediction on sample dataset out of which top ten best performing classifier performances are summarized in Table 1.

Taking into account all the performance measures such as accuracy to receiver operating characteristics (ROC) area, the overall performance best attained (descending order) was by K^*, k-nearest neighbors (k-NN), Random Forest Ensemble (RFE), and algorithm for Fuzzy Unordered Rule Induction Algorithm (FURIA), multilayer perceptron, deep learning for JAVA (Dl4jMlpClassifier), support vector machines, Naive Bayes classifier, and finally AdaBoost that uses stumps for decision-making. $K^*$and k-NN both are instance-based classifiers and produced similar performance results. Followed by RFE, FURIA, and multilayer perceptron (ANN), FURIA and ANN show almost similar performance trend. We selected k-NN and ANN for comparison as many existing works had utilized these which makes comparison with other works easier. Moreover, we chose K^*, FURIA, and DeepLearning4J classifiers for comparison too, as they are state-of-the-art and relatively new machine learning methods. RFE is suited for large datasets to give high accuracy and time efficiency. It is resilient to noise in data and is also capable of dealing with missing values in data. RFE utilizes bootstrapping that reduces variance and keeps the bias in check because creating different subsets of training dataset along with a replacement mechanism ensures that the trees have little or no correlation. Therefore, overfitting is avoided, making it more generalizable. Both training and prediction time of RFE due to parallel computation supported by bagging make it suited for real-time implementation of IPS. Hence, we selected RFE [29] as the suitable classifier module in our proposed methodology.

4. Proposed Localization Methodology

4.1. Problem Formulation. We assume localization/positioning as a combination of clustering and multiclass classification problem where each room is considered to be a class. A two-dimensional indoor area is partitioned into R square grids of dimensions $C \times D\,\mathrm{m}^2$. The centre of each square grid is a reference point (RP). A device equipped with the wireless adapter card can sense wireless signals from a total of A AP-s at a certain RP at a given time, which forms the fingerprint $\mathrm{FP}_i = \{RV_i, L_i\}$. $RV_i = \{\mathrm{rssi}_{i1}, \mathrm{rssi}_{i2}, \mathrm{rssi}_{i3}, \ldots, \mathrm{rssi}_{iA},\}$ where rssi_{ij} symbolizes the RSSI value from jth AP (dBm) in the ith sample of FP collected and L_i is the respective class/room label. Let N such FP constitute the dataset. Localization function, LF, is learnt from the FP dataset to map the observed FP to a certain room label L_x as described by the following equation:

$$L_x = \mathrm{LF}(\mathrm{FP}_i). \tag{1}$$

There are two phases of the proposed methodology (CEnsLoc); namely, training phase and prediction phase as shown in Figure 2. First of all, a sparse Wi-Fi FP dataset with many missing values is collected. In the training phase, the collected FPs reserved for training the system are preprocessed for missing values replacement, followed by PCA application for dimension reduction. Then GMM-based hard clustering is used for nonoverlapping/disjoint data subsets generation. For each such subset, a separate RFE is trained for location prediction and stored in the database. In the location prediction phase, same steps of missing value replacement and PCA computation are performed on the captured FP. Then, FP is matched with a single cluster using the stored GMM. The final prediction L_x is generated by invoking the respective pretrained RFE for the best matched cluster/subset. These phases are formally elaborated in detail in Sections 4.2 and 4.3 respectively.

4.2. Training Phase. During training, the training dataset is fed to the preprocessing module which replaces empty readings with missing value replacement (MV_r). PCA is then performed on the dataset for dimension reduction. Orthogonal transformation is applied by PCA for redundant information removal to decrease the number of predictors. Principal components (PCs) were obtained by applying PCA, which are a set of linearly uncorrelated variables such that maximum variance by some projection of the data is captured by the first principal component and so on. Choosing the smaller of the number of predictors/APs and number of samples minus one, A PCs are generated $\{\mathrm{PC}_1, \mathrm{PC}_2, \ldots, \mathrm{PC}_A\}$. For computation of PCs, first the mean RSSI value of each AP is subtracted from the ith RP using the following equation:

$$X_i = \frac{1}{N}\sum_N \mathrm{FP}_i - \frac{1}{R}\frac{1}{N}\sum_R\sum_N \mathrm{FP}_i, \tag{2}$$

where N is the total no. of rows of samples/dataset and R is the total no. of RPs. The PCs matrix is computed by the following equation:

$$\mathbf{PC}_i = X_i \times \mathbf{E}_A, \tag{3}$$

where E_A is the eigenvector matrix of the average RSSI value of each AP in the ith RP. The resulting dataset is then divided

Table 1: A comparison of performance measures of various classification algorithms for Wi-Fi-based position estimation.

Algorithm	Time to build model (sec)	Accuracy	Kappa statistic	RMSE	Precision	Recall	F1	MCC	ROC area
K^*	**0**	**99.52**	0.98	**0.02**	**0.99**	**0.99**	**0.99**	**0.99**	**1**
k-NN	**0**	99.06	**0.99**	0.03	**0.99**	**0.99**	**0.99**	0.98	0.99
RFE	1.11	98.76	0.98	0.04	0.98	0.98	0.98	0.98	**1**
FURIA	5.92	97.26	0.96	0.05	0.97	0.97	0.97	0.96	0.99
Multilayer perceptron	25.84	97.05	0.96	0.06	0.97	0.97	0.97	0.96	0.99
J48	0.1	95.91	0.95	0.08	0.95	0.95	0.95	0.95	0.98
Dl4jMlp classifier	26.31	94	0.93	0.08	0.94	0.94	0.94	0.93	0.99
SVM	5.82	90.6	0.89	0.12	0.93	0.90	0.90	0.9	0.94
Naive Bayes	**0.02**	89.79	0.88	0.12	0.91	0.89	0.89	0.89	0.99
AdaBoost with decision stump	0.1	36.81	0.22	0.25	0.15	0.36	0.21	0.18	0.76

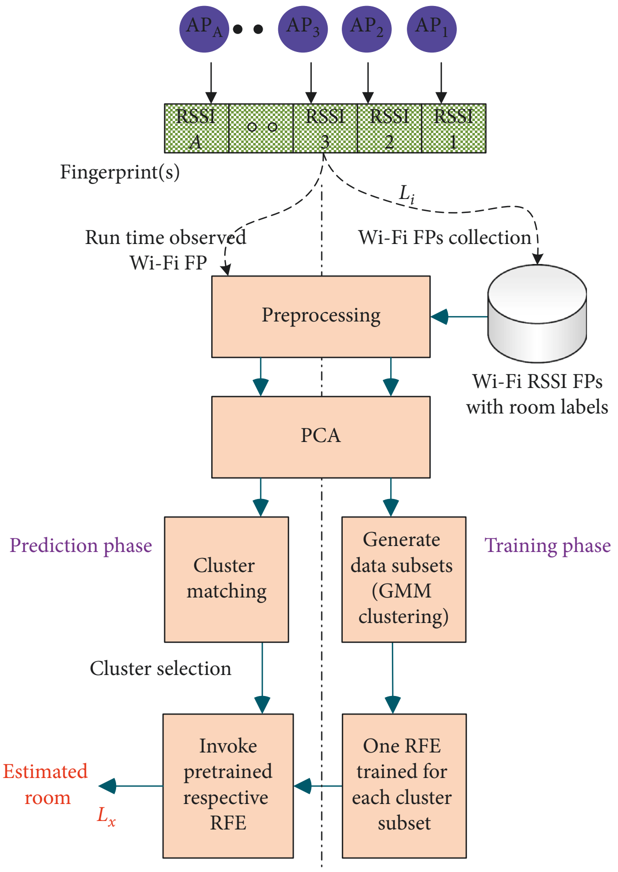


Figure 2: Proposed localization methodology (CEnsLoc).

into subsets using GMM clustering (K data subsets). Equation (4) is a distribution based on 2D Gaussian where mean is represented by μ and the covariance matrix is $\sum$. A GMM with N as the no. of overlapping distributions is described by the following equations:

$$N(x \mid \mu, \Sigma) = \frac{1}{2\pi\sqrt{|\Sigma|}} \exp\left\{-\frac{1}{2}(x-\mu)^{\mathrm{T}}\Sigma^{-1}(x-\mu)\right\}, \tag{4}$$

$$P(x) = \sum_{k=1}^{N} \pi_k N\left(x \mid \mu_k, \Sigma_k\right), \tag{5}$$

$$\sum_{k=1}^{N} \pi_k = 1, \tag{6}$$

where π_k defines the mixing coefficient to express the weight of each mixing element (weighted sum being 1). The resultant shape of 2D Gaussian is the average of distributions individually, in terms of covariance and mixing coefficients. Assuming that a linear mix of weighted coefficients for each of the respective distribution's average and covariance is obtained, and by incorporating sufficient distributions, a final density function may be obtained. The reason behind GMM clustering was the similarity between Gaussian distribution and the radio propagation characteristics of a Wi-Fi AP [30] which makes GMM a highly suitable candidate for clustering Wi-Fi RSSI vectors.

Furthermore, each data subset is used for training a RFE to predict the room label as a multiclass classifier. The trained models of GMM as well as all RFEs are stored for later use in the prediction phase.

The algorithm for training and prediction phases of CEnsLoc is given in Algorithm 1.

Let the total number of samples be N_{sample} in the training set, the number of trees is N_{tree}, the number of maximum splits allowed is $S_{\max}$, the number of predictors for the classifier is A', f is the value specifying no. of input predictors that are utilized to split at a tree node, and tc denotes the total no. of classes in the dataset. For finding best split, RFE uses Gini Index as given in the following equation where P_j is the class j's relative frequency in N_{sample}:

$$\mathrm{Gini}\left(N_{\text{sample}}\right) = 1 - \sum_{j=1}^{\mathrm{tc}} \left(P_j\right)^2. \tag{7}$$

RFE is trained using the method presented in Algorithm 2.

4.3. Prediction Phase.

The average of collected RSSI FP is fed to CEnsLoc, A PCs are computed by the method similar to the training phase. The saved model of GMM is invoked for cluster matching. Matched cluster's trained RFE is invoked where the final decision is computed by the majority vote described in Algorithm 2.

4.4. Time Complexity of Training and Prediction Phases for CEnsLoc.

Ceteris paribus, the time complexity of the training phase and the prediction phase for CEnsLoc is essentially dependent upon the size of the experimental area,

```
Input: training dataset with total $A$ predictors
    Missing value replacement $\mathrm{MV_r}$
    Maximum number of clusters $K_{\max}$
Output: predicted location Lx
For training:
Replace empty values with $\mathrm{MV_r}$
Apply PCA on dataset to generate $A'$ predictors
For $k = 1$ -> $K_{\max}$
    Generate clusters
    Generate and save $k$ data subsets
    For each $p \in k$ data subsets
        Train $p$ RFE using Algorithm 2 (training)
        Calculate performance measures
    End for
End for
Choose optimal configuration
Save respective models for GMM, all RFEs
For prediction at a new point x:
Replace missing values with $\mathrm{MV_r}$
Apply PCA on the FP
Match one cluster $C_{\mathrm{match}}$
Invoke RFE of $C_{\mathrm{match}}$ using Algorithm 2 (prediction)
```

Algorithm 1: CEnsLoc training and prediction algorithm.

our acquisition regimen, the resultant dataset of FPs, and how the dataset is manipulated by the tandem of schemes we employ.

4.4.1. Time Complexity of Training. In the training phase for PCA, time complexity is governed by the following equation:

$$O\left(\min\left(A^3, N^3\right)\right), \tag{8}$$

where $A =$ no. of predictors and $N =$ no. of observations.

For training a decision tree (DT) that has not been pruned, the expression is as follows:

$$O(A \times N \log(N)). \tag{9}$$

As RFE consists of numerous DTs, merely a small no. f is used from total predictors A. Complexity for a single DT in RFE is represented by Equation (10) and the complexity of N_{tree} by Equation (11):

$$O(f \times N \log(N)), \tag{10}$$

$$O(N_{\mathrm{tree}} \times f \times N \log(N)), \tag{11}$$

where $N_{\mathrm{tree}} =$ no. of trees in RFE and $f =$ random features that are chosen to get the best split.

While trying to control trees' depth grown using $S_{\max}$, training complexity of one RFE is as follows:

$$O(N_{\mathrm{tree}} \times f \times N \times S_{\max}). \tag{12}$$

Since K ensembles are grown for predicting room level, time to train complexity is represented by the following equation:

$$O(N_{\mathrm{tree}} \times f \times N \times S_{\max} \times K). \tag{13}$$

For GMM, the complexity is expressed by the following equation:

$$O\left(N \times K \times D^3\right), \tag{14}$$

where $N =$ no. of samples, $K =$ no. of components, and $D =$ no. of dimensions.

Incorporating all, the time complexity to train for CEnsLoc is as follows:

$$\begin{aligned} &O\left(\min\left(A^3, N^3\right)\right) + O\left(N \times K \times D^3\right) \\ &\quad + O(N_{\mathrm{tree}} \times f \times N \times S_{\max} \times K). \end{aligned} \tag{15}$$

4.4.2. Time Complexity of Prediction. For prediction, time complexity for PCA is given by the following equation:

$$O\left(\min\left(A^3, N^3\right)\right). \tag{16}$$

The complexity for a DT and an ensemble in terms of prediction time are shown by Equations (17) and (18), respectively:

$$O(N \log(N)), \tag{17}$$

$$O(N_{\mathrm{tree}} \times N \log(N)). \tag{18}$$

$S_{\max}$ controls trees depth; therefore, complexity for an ensemble is given by the following equation:

$$O(N_{\mathrm{tree}} \times N \times S_{\max}). \tag{19}$$

Only a single RFE out of K ensembles gets invoked for predicting a room.

Therefore, time complexity for GMM is given by the following equation:

$$O\left(K \times D^3\right). \tag{20}$$

Finally, the overall complexity for CEnsLoc in terms of prediction is as follows:

$$O\left(\min\left(A^3, N^3\right)\right) + O\left(K \times D^3\right) + O(N_{\mathrm{tree}} \times N \times S_{\max}). \tag{21}$$

5. Experimental Results and Discussion

This section entails hardware equipment, software used, and particulars of experiments that were used to evaluate the performance of CEnsLoc in light of accuracy, precision, recall, training, and response time.

An Intel machine (64-bit Xeon: X5650) with a master clock at 2.67 GHz with 24 GB RAM, and 64-bit Windows 10 Education was used for experimentation in MATLAB. The real dataset was developed through FP collection at the ground floor of Software Engineering (SE) Centre, University of Engineering and Technology (UET), Lahore Pakistan. The building's dimensions are 39 m × 31 m (1209 m^2) containing offices, class rooms, laboratories, and open corridors. Figures 3 and 4 depict the building's floor plan, room labels (L1–L10 closed rooms, L12 open corridor, and L11 a semiopen room), a total of 180 RPs, and the

```
Input: data subset with total $A'$ predictors for training
    No. of tree $N_{\text{tree}}$
    Allowed maximum no. of splits $S_{\max}$
    Random no. of predictors/features $f$
Output: estimated location $L_x$
For system training:
Step 1: for $l = 1$ to $N_{\text{tree}}$
(i) Choose a bootstrap sample set (SS) of size ($N_{\text{sample}}$) with replacement from the training data subset
(ii) Generate a Random Forest Tree ($T_l$) to SS, via recursively iterating (a-c) for every terminal node of tree, unless the maximum no. of
     splits ($S_{\max}$) is reached
    (a) Randomly select $f$ features/variables from the $A'$ predictors ($f << A'$)
    (b) Choose the best features/split-point from the $f$ employing Gini Index
    (c) Split node forming into two children nodes
        Step 2: Produce the resulting ensemble of trees $\{T_l\}_1^{N\text{tree}}$
        For location prediction at a new point x from RFE $L_{rf}^{N\text{tree}}$:
        Let, $L_m(x)$ be the room/class prediction by the $m^{\text{th}}$ RFE tree
        $L_{\text{rf}}^{N\text{tree}}(x) = \text{maj. vote } \{L_m(x)\}_1^{N\text{tree}}$
```

Algorithm 2: RFE classification algorithm for training and prediction.

number of samples collected per room. RPs that are represented by small coloured dots in rooms in Figures 3 and 4 enlist the total number of samples collected at each location L1–L12. Such notion of rooms was used as walls playing a crucial role in fluctuation of Wi-Fi RSSI values [31, 32]. The area was planned into a grid of cells of $1.5 \times 1.5\,\text{m}^2$. Each cell center was marked as the Reference Point (RP) for FP collection.

The complete dataset consisted of 20087 Wi-Fi RSSI FPs, in which total 40 APs were detected. Figure 4 depicts the number of FPs captured in each room/location marked as L1–L12. It must be noted that all these APs belong to university infrastructure comprising of SE centre and its immediate neighboring buildings. The FPs were preprocessed following the same process described in Sections 3.1 and 3.2. PCA-based dimension reduction was employed with optimal results found with 23 PCs. The resultant dataset was divided with 70 : 30% stratified ratio for training and testing subsets. The experimental results are discussed using both 10-fold cross validation (10-CV) on training subset and on unseen 30% test subset. GMM clustering then partitioned the training data into further subsets, and the optimal configuration was two clusters, shared covariance kept as true and diagonal covariance. Furthermore, one RFE was trained for each subset with 132 trees, 1024 maximum splits, and 8 random features. CEnsLoc was hosted at a machine, and the mean of observed RSSI FPs was used for run time location estimation after applying the same model of PCA and GMM cluster matching finalized from the training phase. Best matched cluster's respective RFE was invoked for room prediction. Various companion apps can query location of subscribed users using the IPS configured as a server. Response time was computed as an average of the difference between localization query time and prediction generation time. The following formulae were used to compute the performance parameters:

$$
\begin{aligned}
\text{accuracy} &= \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \\
\text{precision} &= \frac{\text{TP}}{\text{TP} + \text{FP}}, \\
\text{recall} &= \frac{\text{TP}}{\text{TP} + \text{FN}}.
\end{aligned}
\tag{22}
$$

5.1. Classification Effectiveness and Efficiency. Tables 2 and 3 summarize the 10-CV performance evaluation of CEnsLoc, comparing it with k-NN [33], artificial neural network (ANN) [34], K^* [35], FURIA [36], and DeepLearning4J [36]. The best results are highlighted throughout with boldface. k-NN results were computed averaged over six different configurations based on the number of neighbors, similarity measure employed, and neighbor weightage. Similarly, six different configurations of ANN, namely, 2-, 3-, and 4-layer ANNs each having varying number of neurons per layer, employing two learning algorithms SCG and RBP, were averaged out to obtain the results. For K^*, the entropic blend percentage was varied from 10 to 90 percent. The results obtained for FURIA were obtained by varying the count of folds for growth and pruning as well as through varying minimum instances weight that was for each split. DeepLearning4J results were obtained by varying the number of neurons per layer, number of dense layers in the network, and training algorithm out of many possible variations in the hyper parameters. The results by CEnsLoc were generated by the found optimal configuration of our proposed approach. For all the approaches, the same set of configurations, as used during 10-CV performance evaluation, were used for result generation on 30% unseen stratified test data subset whose results are presented in Table 4. The best performance on both 10-CV results (Table 2) and test dataset (Table 4)

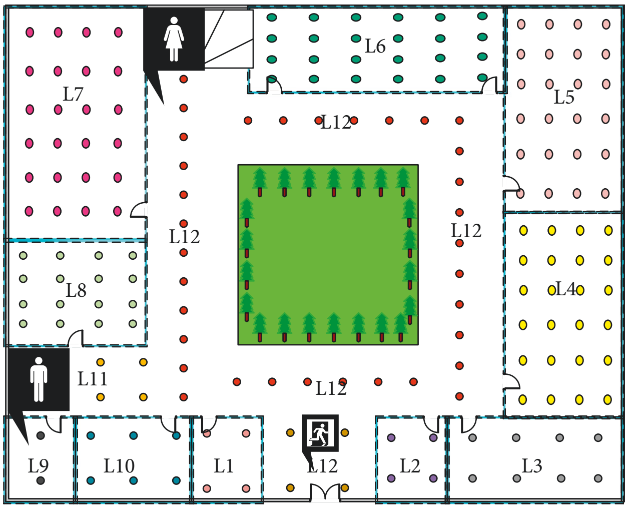


Figure 3: Room labels and marked RPs.

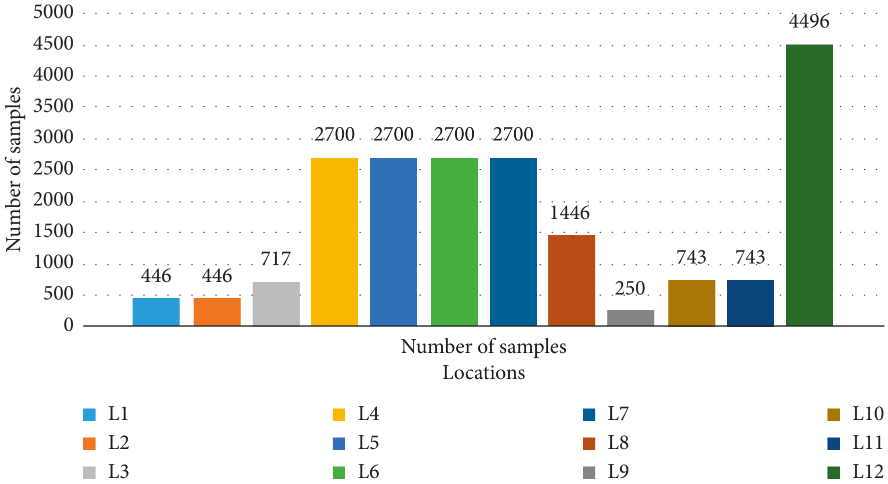


Figure 4: Number of samples per room.

were obtained by CEnsLoc with 97% and 95% accuracy followed by FURIA (92%, 90%), k-NN (91%, 89%), K^* (90%, 87%), ANN (85%, 82%), and DeepLearning4J (73%, 71%), respectively. The results are presented on both 10-CV and unseen test data subset to show that, for a small dataset, 10-CV results can provide a good performance estimate, as results on the test data subset were found to be showing similar trends as depicted by the 10-CV results. Although deep leaning and ANN provide good performance results, but finding optimal configuration for a huge number of tunable parameters is a tricky task. Moreover, during experiments, it was found that despite generic guidelines for parameter tuning, the performance measures and convergence rate of ANN as well as deep learning schemes highly fluctuate with even a slight variation in the number of neurons in layers, training algorithm, and number of hidden layers, which makes it harder to find optimal configuration for ANN and deep learning models. Lazy and instance-based approaches such as k-NN and K^* are also good candidates for indoor localization; however, they have a very limited number of tunable parameters resulting in inability to surpass CEnsLoc performance despite trying different parameter combinations. They do not generalize well as compared to RFE as the end result of prediction is heavily dependent on the majority vote by k closest matched samples in the dataset. They also need template matching with the entire dataset for one location prediction which results in growing response time with increasing number of samples in the dataset which is highly likely in practical real-world scenarios. As in a typical building, there are a quite huge number of visible APs, and a sufficiently large number of samples are also required for classifiers to work properly.

A minimum response time of $2.05E-05$ seconds was obtained by FURIA. DeepLearning4J stood second with $6.82E-05$ seconds response time. ANN, k-NN, and CEnsLoc all had response time on the scale of $E-04$ seconds which is 10 times slower than aforementioned two

Table 2: Tenfold cross-validated performance evaluation and comparison of CEnsLoc.

	Accuracy	Precision	Recall
k-NN	0.91	0.88	0.87
ANN	0.85	0.83	0.78
K^*	0.90	0.96	0.62
FURIA	0.92	0.89	0.82
DeepLearning4J	0.73	0.51	0.41
CEnsLoc	**0.97**	**0.98**	**0.97**

Table 3: Tenfold cross-validated performance comparison of CEnsLoc in terms of training and response time (seconds).

Time (sec)	Training time (10-fold)	Avg. training time (1-fold)	Response time dataset	Response time (1 sample)
k-NN	—	—	1.97	$1.70E-04$
ANN	267.20	26.72	0.18	$1.41E-04$
K^*	—	—	103.09	$8.17E-02$
FURIA	158.7	15.8	**0.03**	$\mathbf{2.05E-05}$
DeepLearning4J	755.59	75.5	0.09	$6.82E-05$
CEnsLoc	**140.07**	**14.007**	2.35	$2.08E-04$

Table 4: Performance evaluation and comparison of CEnsLoc on test subset.

	Accuracy	Precision	Recall
k-NN	0.89	0.87	0.85
ANN	0.82	0.81	0.76
K^*	0.87	0.94	0.60
FURIA	0.90	0.86	0.79
DeepLearning4J	0.71	0.48	0.39
CEnsLoc	**0.95**	**0.96**	**0.94**

approaches, but the difference was trifling, which cannot be detected by any human user of the system; the accuracy, precision, and recall provided by CEnsLoc was much greater than all other approaches.

FURIA stands out as the second best performer regarding indoor localization, which is an upgraded version of the RIPPER algorithm, indicating that rule-based algorithms, specially fuzzified versions with good generalization capabilities perform well for location estimation as well. However, it lagged behind CEnsLoc in terms of accuracy, precision, and recall by 5%, 10%, and 15%, respectively.

The details of both 10-CV and test dataset results for all IPS compared and CEnsLoc are depicted in Figure 5 for side by side visual comparison.

5.2. Out-of-Bag (OOB) Error Results. There is a performance measure called OOB error which is peculiar to RFE. During training of RFE, data subsets are generated with replacement, resulting in some repeated and left out observations (OOB observations) for each tree. That particular tree does not train on these left out OOB observations. Prediction capability of RFE can be measured using "OOB Loss" which is the error made on unseen OOB observations during training. OOB loss measure has been investigated and shown to provide an upper bound on testing error [37], specifically, useful for small-sized datasets. Hence, OOB error can be used just like/instead of unseen test data subset if the available dataset is small or unseen test dataset is unavailable. It provides a very good estimate of the trained classifier's generalization capability. Table 5 summarizes the OOB loss compared with averaged out 10-CV loss indicating that it indeed bounds it.

6. Conclusion and Future Work

Location prediction/estimation provides derivation of meaningful context for a broad range of services and applications. Indoor localization can open altogether a plethora of new opportunities because humans spend most of their time indoors. CEnsLoc offers shorter response time and an overall improvement in accuracy, precision, and recall. With only a few parameters to be tuned, it is suited for FP-based localization which requires frequent recollection of data and retraining. CEnsLoc was able to attain 97% accuracy in comparison with other IPS averaged over 6 different configurations, namely, FURIA, *k*-NN, K^*, ANN, and DeepLearning4J with 92%, 91%, 90%, 85%, and 73% accuracies, respectively. It can be utilized for elderly assistance, navigation, smart buildings, and smart transportation to name a few potential applications.

Our future work includes deployment of CEnsLoc across a wide range of civil infrastructures including different floors and/or buildings to understand its performance in more detail as well as its scalability using crowdsourcing. We also aim to build safety, security, and evacuation guide applications with CEnsLoc at their core for users in offices, universities, and retail. Furthermore, integration with GPS, Bluetooth, and PDR to further enhance accuracy, utilizing

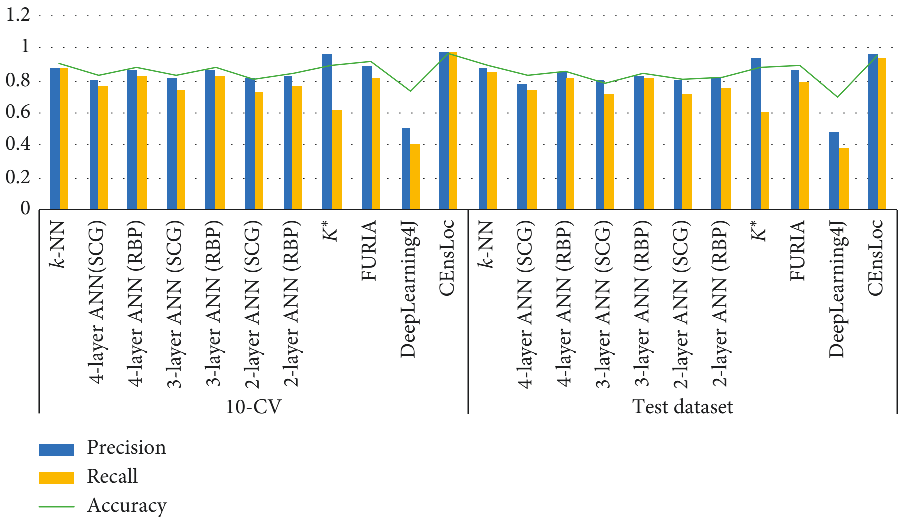


Figure 5: Performance measures on both 10-CV and test dataset.

Table 5: OOB loss.

Average 10-fold loss	OOB loss
0.0292813	0.0300123

available hybrid technologies at a given time, is also part of the plan.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2018R1D1A1B07048697).

References

[1] J. Torres-Sospedra, R. Montoliu, S. Trilles, Ó. Belmonte, and J. Huerta, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.

[2] H. Mehmood and N. K. Tripathi, "Cascading artificial neural networks optimized by genetic algorithms and integrated with global navigation satellite system to offer accurate ubiquitous positioning in urban environment," *Computers, Environment and Urban Systems*, vol. 37, no. 1, pp. 35–44, 2013.

[3] M. M. Soltani, A. Motamedi, and A. Hammad, "Enhancing cluster-based RFID tag localization using artificial neural networks and virtual reference tags," *Automation in Construction*, vol. 54, pp. 93–105, 2015.

[4] M. L. Rodrigues, L. F. M. Vieira, and M. F. M. Campos, "Fingerprinting-based radio localization in indoor environments using multiple wireless technologies," in *Proceedings of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1203–1207, Toronto, ON, Canada, September 2011.

[5] J. Luo and H. Gao, "Deep belief networks for fingerprinting indoor localization using ultrawideband technology," *International Journal of Distributed Sensor Networks*, vol. 12, no. 1, article 5840916, 2016.

[6] S. Saeedi, L. Paull, M. Trentini, and H. Li, "Neural network-based multiple robot simultaneous localization and mapping," *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 880–885, 2011.

[7] M. Azizyan, R. R. Choudhury, and I. Constandache, "SurroundSense: mobile phone localization via ambience fingerprinting," in *Proceedings of the 15th annual international conference on Mobile computing and networking-MobiCom'09*, pp. 261–272, Beijing, China, 2009.

[8] L. Pei, R. Chen, J. Liu et al., "Motion Recognition Assisted Indoor Wireless Navigation on a Mobile Phone," in *Proceedings of the 23rd International Technical Meeting of The Satellite Division of the Institute of Navigation*, pp. 3366–3375, Portland, OR, USA, September 2010.

[9] P. S. Nagpal and R. Rashidzadeh, "Indoor positioning using magnetic compass and accelerometer of smartphones," in *Proceedings of International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT)*, pp. 140–145, Montreal, Canada, 2013.

[10] J. Menke and A. Zakhor, "Multi-modal indoor positioning of mobile devices," in *Proceedings of IEEE International Conference on Indoor Positioning and Indoor Navigation*, pp. 13–16, Alberta, Canada, October 2015.

[11] M. Oussalah, M. Alakhras, and M. I. Hussein, "Multivariable fuzzy inference system for fingerprinting indoor localization," *Fuzzy Sets and Systems*, vol. 269, pp. 65–89, 2015.

[12] N. Li, J. Chen, Y. Yuan, X. Tian, Y. Han, and M. Xia, "A wi-fi indoor localization strategy using particle swarm optimization based artificial neural networks," *International Journal of Distributed Sensor Networks*, vol. 12, no. 3, article 4583147, 2016.

[13] C. Song, J. Wang, and G. Yuan, "Hidden naive bayes indoor fingerprinting localization based on best-discriminating AP selection," *ISPRS International Journal of Geo-Information*, vol. 5, no. 10, p. 189, 2016.

[14] P. Bahl and V. N. Padmanabhan, "RADAR: an in-building RF based user location and tracking system," in *Proceedings of IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No. 00CH37064)*, vol. 2, pp. 775–784, Tel Aviv, Israel, March 2000.

[15] M. Cooper, J. Biehl, G. Filby, and S. Kratz, "LoCo: boosting for indoor location classification combining Wi-Fi and BLE," *Personal and Ubiquitous Computing*, vol. 20, no. 1, pp. 1–14, 2016.

[16] C. Wang, F. Wu, Z. Shi, and D. Zhang, "Indoor positioning technique by combining RFID and particle swarm optimization-based back propagation neural network," *Optik (Stuttg).*, vol. 127, no. 17, pp. 6839–6849, 2016.

[17] J. Xu, H. Dai, and W. Ying, "Multi-layer neural network for received signal strength-based indoor localisation," *IET Communications*, vol. 10, no. 6, pp. 717–723, 2016.

[18] W. Zhang, K. Liu, W. Zhang, Y. Zhang, and J. Gu, "Deep neural networks for wireless localization in indoor and outdoor environments," *Neurocomputing*, vol. 194, pp. 279–287, 2016.

[19] L. Calderoni, M. Ferrara, A. Franco, and D. Maio, "Indoor localization in a hospital environment using Random Forest classifiers," *Expert Systems with Applications*, vol. 42, no. 1, pp. 125–134, 2015.

[20] E. Jedari, Z. Wu, R. Rashidzadeh, and M. Saif, "Wi-Fi based indoor location positioning employing random forest classifier," in *Proceedings of 2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 13–16, Banff, Albeta, Canada, October 2015.

[21] Y. Mo, Z. Zhang, Y. Lu, W. Meng, and G. Agha, "Random forest based coarse locating and KPCA feature extraction for indoor positioning system," *Mathematical Problems in Engineering*, vol. 2014, Article ID 850926, 8 pages, 2014.

[22] R. Górak and M. Luckner, "Malfunction immune Wi–Fi localisation method," in *Computational Collective Intelligence*, pp. 328–337, Springer, Cham, Switzerland, 2015.

[23] R. Górak and M. Luckner, "Modified random forest algorithm for Wi–Fi indoor localization system," in *Proceedings of International Conference on Computational Collective Intelligence*, pp. 147–157, Cham, Switzerland, September 2016.

[24] J. Niu, B. Wang, L. Cheng, and J. J. P. C. Rodrigues, "WicLoc: an indoor localization system based on WiFi fingerprints and crowdsourcing," in *Proceedings of 2015 IEEE International Conference on Communications (ICC)*, pp. 3008–3013, London, UK, June 2015.

[25] G. Ding, Z. Tan, J. Zhang, and L. Zhang, "Fingerprinting localization based on affinity propagation clustering and artificial neural networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 2317–2322, Shanghai, China, April 2013.

[26] O. León, J. Hernández-Serrano, and M. Soriano, "Securing cognitive radio networks," *International Journal of Communication Systems*, vol. 23, no. 5, pp. 633–652, 2010.

[27] S. Tuncer and T. Tuncer, "Indoor localization with bluetooth technology using artificial neural networks," in *Proceedings of the IEEE 19th International Conference on Intelligent Engineering Systems*, pp. 213–217, Bratislava, Slovakia, September 2015.

[28] B. A. Akram, A. H. Akbar, B. Wajid, O. Shafiq, and A. Zafar, "LocSwayamwar: finding a suitable ML algorithm for wi-fi fingerprinting based indoor positioning system," in *Lecture Notes in Electrical Engineering*, A. Boyaci, A. Ekti, M. Aydin, and S. Yarkan, Eds., vol. 504, Springer, Singapore, 2018.

[29] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[30] K. Kaji and N. Kawaguchi, "Design and implementation of wifi indoor localization based on Gaussian mixture model and particle filter," in *Proceedings of 2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–9, Sydney, Australia, November 2012.

[31] C. Wu, Z. Yang, Y. Liu, and W. Xi, "WILL: wireless indoor localization without site survey," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 839–848, 2013.

[32] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: calibration-free crowdsourced indoor localization," in *Proceedings of IEEE INFOCOM*, pp. 2481–2489, Turin, Italy, April 2013.

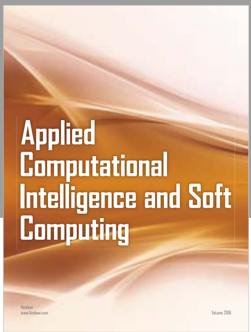
[33] T. Seidl and H.-P. Kriegel, "Optimal multi-step k-nearest neighbor search," *ACM SIGMOD Record*, vol. 27, no. 2, pp. 154–165, 1998.

[34] W. S. Mcculloch and W. Pitts, "A logical calculus nervous activity," *Bulletin of Mathematical Biology*, vol. 52, no. l-2, pp. 99–115, 1990.

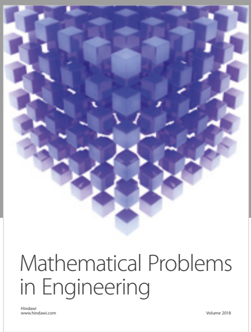
[35] J. G. Cleary and L. E. Trigg, "K∗: an instance-based learner using an entropic distance measure," in *Proceedings of Twelfth International Conference on Machine Learning*, vol. 5, pp. 1–14, Tahoe City, CA, USA, July 1995.

[36] J. Hühn and E. Hüllermeier, "FURIA: an algorithm for unordered fuzzy rule induction," *Data Mining and Knowledge Discovery*, vol. 19, no. 3, pp. 293–319, 2009.

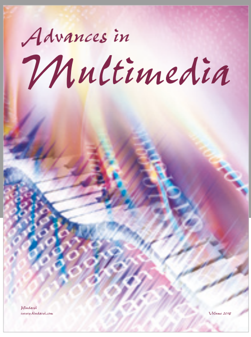
[37] S. Ciss, *Generalization Error and Out-of-bag Bounds in Random (Uniform) Forests*, Ph.D. thesis, French University, Paris, France, 2015.

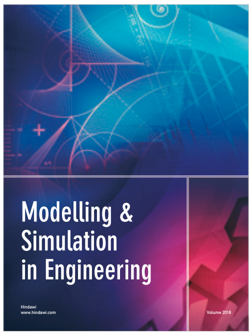

Applied Computational Intelligence and Soft Computing

The Scientific World Journal


Mathematical Problems in Engineering

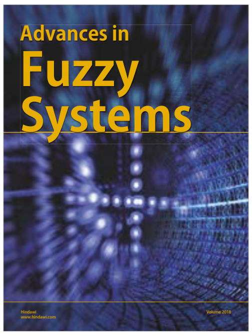
Journal of Engineering

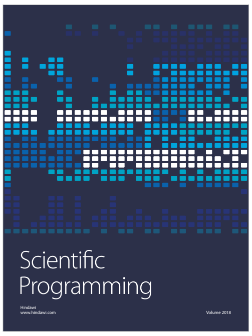

Advances in Multimedia

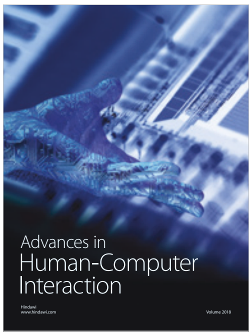

Modelling & Simulation in Engineering

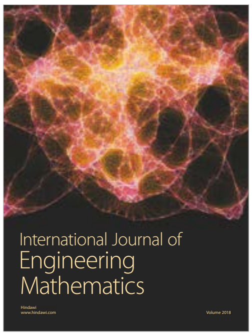
Advances in Artificial Intelligence

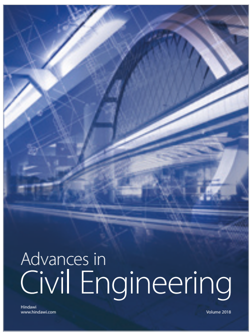
International Journal of Reconfigurable Computing

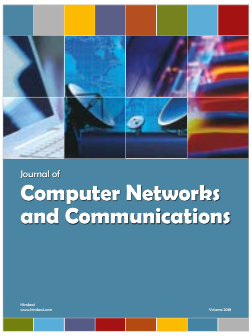

Advances in Fuzzy Systems


Scientific Programming


Advances in Human-Computer Interaction

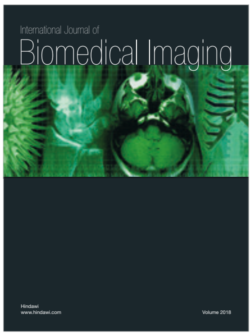

International Journal of Engineering Mathematics


Advances in Civil Engineering

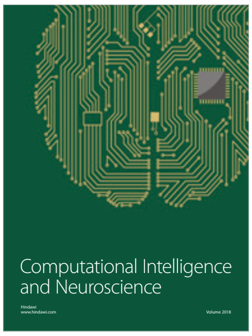

Journal of Computer Networks and Communications

Journal of Robotics

International Journal of Computer Games Technology


International Journal of Biomedical Imaging

Journal of Electrical and Computer Engineering


Computational Intelligence and Neuroscience