Expert Systems with Applications 164 (2021) 113758

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Novel weighted ensemble classifier for smartphone based indoor localization

Priya Roy [a,1], Chandreyee Chowdhury [a,*,2], Mausam Kundu [b], Dip Ghosh [c], Sanghamitra Bandyopadhyay [c]

[a] Computer Science and Engineering Department, Jadavpur University, Kolkata, India
[b] Information Technology Department, Jadavpur University, Kolkata, India
[c] Indian Statistical Institute, Kolkata, India

ARTICLE INFO

Article history:
Received 6 May 2019
Revised 15 April 2020
Accepted 12 July 2020
Available online 3 August 2020

Keywords:
Indoor localization
Machine learning
Ensemble
Dempster–Shafer belief theory
WiFi
RSSI

ABSTRACT

Indoor localization systems have the capability to change the way of providing location-based services in a closed environment. Though there is no agreed-upon technology that works best in indoor, WiFi signal is an important alternative as most of such places are covered by WiFi Access Points (APs). In this paper, the problem of indoor localization is investigated from the perspective of expert systems through applying machine learning techniques. The significant variation of WiFi signal strength with ambient conditions as well as device configuration badly affects the localization accuracy. Thus, the fingerprinting effort required to train a localization system subject to context heterogeneity is huge. The uncertainty in localization performance due to varying contexts is hardly investigated in the literature. Consequently, the main contribution of this paper is to propose a weighted ensemble classifier based on *Dempster–Shafer belief theory* to efficiently handle context heterogeneity. Here, the context is defined in terms of different smartphone configurations used for training and testing the system as well as temporal variation of signals. The method presented here utilizes the *Dempster–Shafer theory of belief* functions to calculate the weights of the base learners in the decision of the ensemble. Belief theory is applied here to handle the inherent uncertainty in WiFi signal variations due to heterogeneous context. Real life experiments are conducted for two datasets, JUIndoorLoc and UJIIndoorLoc at different granularity levels. For JUIndoorLoc, with state-of-the-art classifiers, 86–97% accuracy can be achieved for 10-fold cross-validation. However, when the training context differs from the test conditions, accuracy drops to 62–87%. In such a scenario, the proposed weighted ensemble technique is found to achieve almost 98% localization accuracy when RSSIs, mean and variance of RSSIs are considered as features. The technique can lead to an effective expert system for indoor localization at varying granularity levels. Such systems would be beneficial for pervasive indoor positioning applications as no dedicated infrastructure is needed for positioning.

1. Introduction

The proliferation of mobile devices such as smartphones, tablets have raised the enormous interest in location-based services. The well-known outdoor localization system GPS (Global Positioning System) is unable to locate a user with a smart device in a indoor region due to the signal attenuation and multi-path effect. To date, indoor localization techniques are studied by researchers for academic and commercial interests. Localization based on WiFi signals is a popular approach to obtain room level accuracy (Torres-Sospedra et al., 2014). Such localization systems behave similarly to the knowledge-based expert systems. The knowledge-base, that is, the RSSI radio map is constructed through offline site-survey. However, it is difficult to obtain more precise localization as the key parameter, i.e. RSSI of WiFi APs for estimating an unknown location is not stable with time, indoor ambience such as the movement of people, interference of other devices, and indoor multipath effects like diffraction, reflection, and scattering. Moreover, building structure, device configuration also affect the signal strength (Shih, Chen, Chen, Wu, & Jin, 2012). Early WLAN-based localization system like Radar (Bahl & Padmanabhan, 2000) and Horus (Youssef & Agrawala, 2005) used deterministic and

* Corresponding author
E-mail addresses: priyaroy.rs@jadavpuruniversity.in (P. Roy), chandreyee.chowdhury@jadavpuruniversity.in (C. Chowdhury), mausam.kundu.07@gmail.com (M. Kundu), dipgi2005@gmail.com (D. Ghosh), sanghami@isical.ac.in (S. Bandyopadhyay).
[1] ORCID: 0000-0001-8790-0017
[2] ORCID: 0000-0003-2949-9573

https://doi.org/10.1016/j.eswa.2020.113758
0957-4174/

probabilistic methods respectively to mitigate some of these challenges. In one of the most recent works, a sensor fusion framework (Chen et al., 2015) was proposed to combine WiFi signal strength with pedestrian dead reckoning and landmarks to accurately locate a user. In order to achieve localization accuracy of 1m, this framework needs certain prior knowledge such as landmarks, an initial position to restart the algorithm. Besides, these traditional approaches, the machine learning algorithms such as K-Nearest Neighbor (KNN) (Kim, Shin, & Cha, 2012), K* (Mascharka & Manley, 2015), Support Vector Machine (SVM) (YuFeng, JiangMinghua, LiangJing, QinXiao, & HuMing, 2014) etc., have been adopted for solving the indoor localization problem. Thus, indoor localization problem can be presented in the expert systems framework where the method adopts machine learning techniques for training classifiers based on RSSI based features. However, for better localization accuracy, rather than considering the decision of an individual weak classifier, a combined decision of a group of weak classifiers, i.e. ensemble classifier is preferred. Generally, the context like indoor ambience and device configuration may vary at the time of acquiring train and test data. As a result, the classification accuracy of an individual classifier or base learner is also degraded. An ensemble model consisting of multiple context/condition-specific base learners has a generalization capability, which is usually much powerful than individual base learners.

In this regard, the ensemble approaches like bagging and boosting have been proposed in Menéndez, Campomanes, Trawiński, and Alonso (2011) and Taniuchi and Maekawa (2014) where Random Forest and Decision Tree are used as base learner respectively. Belmonte-Fernández, Montoliu, Torres-Sospedra, Sansano-Sansano, and Chia-Aguilar (2018) have introduced an ensemble classifier based on the sum of the probability estimates of different state-of-the-art classifiers. In most of the cases, the reported localization accuracy of their proposed ensemble model is better than individual classifiers for similar train and test conditions. In JUIndoorLoc (Roy, Chowdhury, Ghosh, & Bandyopadhyay, 2019), an ensemble of condition-specific classifiers based on KNN is reported that performs simple majority voting for localization. However, the uncertainty of test conditions and the need for identifying representative ambient conditions were not taken care-of.

These prior works motivated us to propose a weighted ensemble of classifiers for developing a stable expert system for indoor positioning that can inherently handle the uncertainty of test conditions with respect to training. Thus, an expert system for indoor positioning could be developed that can handle various user contexts. Consequently, in this paper, different context-specific individual machine learning classifiers are ensembled according to *Dempster–Shafer belief theory* (Shafer, 1976). This theory was developed by Dempster (1967) and Glenn Shafer (1976) for obtaining degrees of belief for one question from subjective probabilities for a related question. The theory enables us to take decisions in the presence of uncertainties. In indoor localization, the test conditions or test device configurations cannot be predicted earlier. Thus, simple majority voting of all base classifiers (giving equal importance to all conditions) may not be the best solution. Hence, in our work, a trust model, based on the belief theory, evaluates degrees of belief for each condition-specific base learner according to their prediction results during the training phase. The ratio of trust of the base learners w.r.t. the total trust value indicates the importance of the learner in overall decision making in the presence of uncertainty. The proposed mechanism can lead to an effective expert system for indoor localization with a relatively stable performance at different granularity levels.

Accordingly, our main contribution in this paper is noted below:

- Our proposed weighted ensemble method utilized *Dempster–Shafer belief theory* for effective decision making in the presence of context heterogeneity caused by temporal and device variation.
- Our proposed method has been implemented and thorough experiments have been conducted using *JUIndoorLoc* dataset (Roy et al., 2019) collected at a granularity level of $1m \times 1m$ and another benchmark dataset, *UJIIndoorLoc* (Torres-Sospedra et al., 2014) at a different granularity level.

The remainder of this paper is organized as follows: The prior works are studied in the Section 2. Section 3 presents problem definition followed by detailed discussion of our proposed work in Section 4. The experimental setup and performance of our proposed weighted ensemble classifier on JUIndoorLoc dataset (Roy et al., 2019) are discussed in Section 5. Moreover, the performance of our proposed method on *UJIIndoorLoc* dataset (Torres-Sospedra et al., 2014) is illustrated in Section 6. Finally, Section 7 concludes the paper.

2. Related work

The indoor localization domain is enriched by many researchers effort from the last two decades though a commercially scalable and ubiquitous solution is yet to be developed. Some of the relevant state-of-the-art works are summarized in Table 1. The first significant work is RADAR (Bahl & Padmanabhan, 2000) which was proposed in early 2000. This system utilizes the signal strength and signal-to-noise ratio along with the triangulation method. Another popular system is Horus (Youssef & Agrawala, 2005) where various modules have been proposed to address different causes of wireless channel variations. This system is implemented by many researchers as it requires less computational resources. In addition, the crowdsourcing approach is also proposed in many research works (Rai, Chintalapudi, Padmanabhan, & Sen, 2012; Yang, Wu, & Liu, 2012) to reduce the fingerprinting effort which is extremely time-consuming and labor intensive. Users may use various devices and they may partially label the acquired fingerprints. Thus, the recorded coarse-grained fingerprints typically reduce localization accuracy.

Along with these approaches, the researchers also applied machine learning approaches for solving indoor localization problem. This approach generally requires a training model which is constructed by surveying a site in the offline phase. Accordingly, the current location of a device is predicted by matching the fingerprints collected at the current instance with the previously collected fingerprints. Several research efforts are found to use supervised machine learning algorithms or classifiers like SVM (YuFeng et al., 2014; Rossi, Seiter, Amft, Buchmeier, & Tröster, 2013), Decision Tree (Yim, 2008), KNN (Kim et al., 2012), K* (Mascharka & Manley, 2015) as well as semi-supervised (Zhou, Tang, Tian, Xie, & Nie, 2017) and unsupervised (Wang et al., 2012) learning algorithms. Interestingly, transfer learning is a new paradigm for improving the learning performance when the train and test data of various devices and indoor ambience are considered. In this case, transferring the knowledge gained from the source indoor environment may effectively improve the learning performance of the new environment or target domain (Liu et al., 2018). Hence, the overhead of site survey for target domain gets reduced and scalability of the system is enhanced. In this context, the source environment should have sufficient labeled fingerprints to achieve satisfactory localization performance. However, getting a sufficient amount of labeled fingerprints for all representative WiFi signal features is difficult. The dependency relation between the features may also vary with time, context and device.

Table 1
Comparison between state-of-the-art works.

Existing work, year	Technique	Challenges considered	Remarks
Horus (Youssef & Agrawala, 2005) in 2005	Probabilistic method	Temporal, spatial variation	Achieved accuracy within 2.1 meters 90% of the time. Able to cover large areas through clustering techniques. Through its implementation on the mobile devices and low energy requirements, a large number of users use this system.
LuMA (Sun et al., 2008) in 2008	Transfer Learning	Temporal, device heterogeneity	It is an adaptive localization approach irrespective of time and device factors. Hence, the calibration efforts for a new environment can be reduced significantly.
(Oussar et al., 2011) in 2011	Supervised learning (SVM one-vs-one/all), Semi-supervised learning (Transductive SVM)	Use GSM signal due to the fluctuation of WiFi signals	Achieved room level localization accuracy using both supervised and semi-supervised approaches. However, fine-grain localization is hardly possible while considering the GSM signal.
Zee (Rai et al., 2012) in 2012	Placement-independent step counting and orientation estimation, augmented particle filtering, back propagation, WiFi-based particle initialization	Reduce fingerprint effort, no user-specific prior knowledge	Introduces a zero-effort crowdsourced system that requires WiFi signals and inertial sensor data. Existing fingerprint and model-based schemes achieved significant accuracy with these crowdsourced data.
SmartPDR (Kang & Han, 2015) in 2015	Step event detection, heading direction and step length estimation	Avoids the costs of deploying and managing WiFi APs	Introduces a pedestrian tracking system based on inertial sensor readings that completely runs on a smartphone. Achieved less than 2 meters of localization error in the entire experimental period. However, diverse types of smartphones having various inertial chipsets may influence performance.
(Sánchez-Rodriguez et al., 2015) in 2015	Multiple weighted decision trees (C4.5 algorithm) for the base model, Adaptive boosting (AdaBoostM1) for ensemble model	Reduces computational complexity	Introduces a system based on the WiFi RSSI and embedded digital compass of a mobile device. The elapsed time to predict an unknown location is considerably less than other well-known positioning systems. Achieved an average localization accuracy of 2.1 m.
PhaseFi (Wang et al., 2016) in 2016	Deep network with 3 hidden layers, greedy learning algorithm, Bayes method (RBF)	Use Chanel State Information (CSI) to avoid randomness of WiFi RSSI	First indoor positioning system that use calibrated phase information of CSI. Provides low computational overhead for real-time localization and outperforms three CSI/RSSI based benchmark schemes.
(Belmonte-Fernández et al., 2018) in 2018	Radiosity signal propagation model, ensemble classifier based on the sum of probability estimate of all base classifiers	Reduces calibration effort	Proposed a radiosity signal propagation model, using which the WiFi radio map is directly calculated instead of manual acquisition of data. However, the radiosity model requires an accurate floor plan of the scenario and locations of WiFi APs.

In the recent past, multi-classifiers or classifier fusion or an ensemble of many weak learners has been raised as a very powerful approach of machine learning while performing high dimensional, complex classification problems. Generally, an individual classifier provides different patterns of generalization. By combining those results, an ensemble method can be able to yield better performance than any of its individual classifiers (Kuncheva, 2014).

There are a certain classical group of approaches that consider data re-sampling process for generating different training sets to train each individual weak learner or base classifier. Among them in the bagging approach (Breiman, 1996), the weak learners learn independently from certain previously resampled training sets namely bags. These bags are selected randomly with replacement from the original train set. In this regard, certain multi-classification systems for indoor localization are designed using bagging and those are found to give better results than other fingerprint matching algorithm such as the nearest neighbor. Menéndez et al. (2011) have used J48G (Webb, 1999) to derive the component classifiers whereas the final multi-classification system has designed using bagging. Finally, aggregation of the confidence degrees of all the instance for each classifier is done by one of algebraic functions such as mean, median, max, min. Trawiński, Alonso, and Hernández (2013) have proposed another multi-classifier based indoor localization system where two standard methodologies i.e bagging and bagging combined with random subspace (Panov & Džeroski, 2007) have been used. Considering the temporal variation of RSSI, the train and test sets have been collected at different times. In the online phase, the final output is calculated based on the aggregated results provided by the set of component classifiers. This aggregation is done by some algebraic functions such as mean, median. Singh, Aggarwal, and Ujwal (2018) have introduced a real-time indoor localization system using bagging and decision tree based ensemble approaches like Random Forest. WiFi signal strengths have been captured by multiple devices in various timestamps for analyzing the variation of RSSI. Among the various classifiers, random forest gives the best results in terms of the metrics like accuracy, F-measure, etc. However, in bagging some training samples may get repeatedly misclassified in every bag. In addition, the decision tree often takes higher time to train a model. Sometimes the calculation in a decision tree can go far and hence, it becomes more complex than other algorithms. The decision tree often gets unstable as a small variation in the data can cause a huge change in its structure.

In boosting approach (Schapire, 1990) sequentially generated weak learners classify a subset of train data according to the performance of the previous learner(s) in the series. Opposed to bagging, in the train subset selection process, the incorrectly predicted instances by the previous learners have a higher selection probability. Taniuchi and Maekawa (2014) have designed a WiFi-based indoor positioning system where boosted position estimator is used for finding the precise location of the user. The final position is determined by the aggregated estimation of weak estimators. The weak estimators have different weights, depending on the importance of randomly selected APs. An AP's importance is determined based on the signal strength, observation frequency and variance of signal strengths. The weights are computed from the three indicators, where mean, variance and observation frequency of the APs are taken into account during the computation. Weak estimators compute the weighted average of the output coordinates and the average of the coordinates is taken as the final estimated coordinate. Sánchez-Rodríguez, Hernández-Morera, Quinteiro, and Alonso-González (2015) have proposed a model to predict an unknown location of a device using an ensemble model based on AdaBoostM1 algorithm having different multiple weighted decision trees (C4.5 algorithm) as weak learners. However, boosting increases the overall complexity of any system. Besides, AdaBoost is very sensitive to noisy data and also gets highly affected by the outliers.

Apart from these, some other research efforts are also found to improve the positioning accuracy using the ensemble of various machine learning classifiers. Belmonte-Fernández et al. (2018) have designed an ensemble classifier based on the sum of probability estimates of six classifiers namely, Bayesian Network (BN), KNN, Multi-Layer Perceptron (MLP), Random Forest (RF), SVM, and Sequential Minimal Optimization (SMO) to estimate user's position. Bergeron, Bouchard, Gaboury, and Giroux (2018) have proposed an indoor tracking system based on radio frequency identification (RFID) tags readings. This system predicts a location using a random forest algorithm which is basically an ensemble of 100 decision trees. Most of these works do not specifically address the problem of context heterogeneity in terms of device and ambient conditions through ensemble technique. Ghosh, Roy, Chowdhury, and Bandyopadhyay (2016) have captured RSSI data for different conditions such as door open/close and for different devices. Localization accuracies of individual classifiers are reported to vary between 58% and 85% considering only RSSI values for a small dataset collected by them. However, the localization accuracy is found to increase to almost 96% by applying an ensemble method which is based on the majority voting among the decision of all individual base classifiers. However, conditions like door open/close, etc. does not affect the RSSIs significantly unless the indoor environment has many spacious doors and windows to allow signal noise to be introduced. However, the device configurations play a major role in RSSI variation for a given AP at a particular location point.

Consequently, in our research work, context heterogeneity in terms of time and device is addressed. Different time and device specific WiFi signal strengths are used to form condition-specific train sets. Those train sets are further taken as input to the base classifiers. The importance of each base classifier in the decision of the ensemble is decided by applying *Dempster–Shafer belief theory* as discussed in the subsequent sections of this paper.

3. Problem definition

Different notations that are used in our paper are described in Table 2.

Given, a train dataset, having RSSIs along with labeled location points (LP) and a test dataset, having only RSSIs (without location labeling), determine the LPs for the instances of test set.

Let us consider, the i^{th} fingerprint, r_i, is collected at location point, l_k, where r_i consists of RSSI signals received from n APs along with location labeling. Thus, r_i is represented as $\{rssi_{i,1}, rssi_{i,2}, \dots, rssi_{i,n} : l_k\}$, where, $rssi_{i,j}$ represents the *RSSI* value of j^{th} AP, a_j, at location point, l_k. The set of all the LPs is represented by $L = \{l_1, l_2, \dots, l_o\}$. Depending upon context heterogeneity in terms of time(t) and device(d), multiple fingerprints are collected, which is represented as $R^{(t,d)} = \left\{r_1^{(t,d)}, r_2^{(t,d)}, \dots, r_m^{(t,d)}\right\}^T$. Similarly, the test dataset, collected from o' number of LPs, is represented by $R' = \{r'_1, r'_2, \dots, r'_{m'}\}^T$, where $r'_i = \left\{rssi'_{i,1}, rssi'_{i,2}, \dots, rssi'_{i,n}\right\}$ and $|R^{(t,d)}| > |R'|$. These heterogeneous conditions such as, when fingerprint is collected and using which device it is collected for train set are known whereas the conditions of test set are unknown. Consequently, indoor localization problem is to predict $l_k \in L$ corresponding to each $r'_i \in R'$ given labeled dataset $R^{(t,d)}$.

4. Design of the proposed weighted ensemble method

This section discusses the general phases of an indoor localization system as depicted in Fig. 1. We first present how datasets have been collected from the experimental region for varying contexts followed by a brief description of the data preprocessing methods. Finally, the working principle of the proposed weighted ensemble classifier is elaborated.

4.0.1. Data collection

The *JUIndoorLoc* dataset[3] have been used in this work. Description about this dataset can be found in Roy et al. (2019). From this dataset, the RSSI data from 4*th* floor, covering 882 sq.m area of a departmental building in our University campus, is considered here. These data have been acquired over a period of 2 months. Moreover, the fingerprints from 96 APs are collected from the entire floor at different times in a day using 4 handheld devices. The region is divided into $1m \times 1m$ LPs or cells for precise positioning. However, RSSI values have not been collected from all $1m \times 1m$ LP for the presence of obstacles. An Android application as in Roy et al. (2019) is used to collect the RSSI of all available APs from LPs and sends the data to a server to analyze. The recorded RSSI values are stored along with the identifier of LP, collection time, device identifier and detailed features of APs like Basic Service Set Identifiers (*BSSIDs*) and Service Set Identifiers (*SSIDs*). The RSSI scan is repeated 2 to 3 times at a cell from each device to collect around 15 fingerprints and among them, 7 fingerprints are selected randomly.

Accordingly, Fig. 2 shows the signal strengths of a particular AP in each LP received by a device. This radio map helps us to identify the characteristics of signal strengths and also allows to identify the approximate range of an AP in indoors. The signal strengths vary from -100dBm to -40dBm as illustrated using the radio map. -100dBm refers to the weak signal strength while -40dBm refers to the strongest signal strength received. As the signal strengths were collected during the various time of a day, a number of signal strength values having different timestamps can be observed at a particular position with respect to a device and an AP. Hence, to illustrate the characteristics of signal strength received during the entire time period, the mean signal strength value is calculated. This mean signal strength value at each location point is shown in the radio map using various colors depicting the change in mean signal strength over the whole area of coverage. The red color in the radio map indicates the particular LPs which are not considered during the collection of signal strengths.

4.0.2. Data preprocessing

All APs are not heard from every LP due to limited coverage range of WiFi, nearby interfering devices, the presence of obstacles and so on. These, missing entries need to be filled before analysis. In JUIndoorLoc dataset (Roy et al., 2019), the RSSI values vary between -11dBm to -100dBm. Hence, these missing features are filled with -110dBm. Generally, WiFi hotspots or even the APs of nearby buildings are also heard during data collection. Those APs having very weak signal strength exhibits poor distance sensitivity. Moreover, hotspots are movable and alive for a short dura-

[3] https://drive.google.com/drive/folders/1_z1qhoRIcpineP9AHkfVGCfB2Fd_e-fD.

Table 2
Terminology used in our proposed work.

Notations	Description	Notations	Description
n	Number of APs	bl	Belief (how much trustworthy the classifier is)
o	Number of LPs in train set	dl	Disbelief (how much imprecise the classifier is)
m	Number of fingerprints present in train set	un	Uncertainty (When both bl and dl are lacking)
o'	Number of LPs in test set	T_q	Trust of a classifier
m'	Number of fingerprints present in test set	w_q	Weight of a base classifier
b	Number of train subsets	W	Weight of all classifiers i.e. $W = \{w_1, w_2, \ldots, w_b\}$
f	Number of devices	$Tr(q)$	q^{th} train subset
a_j	j^{th} AP	TR	Whole train set, i.e. $TR = \{Tr_1, Tr_2, \ldots, Tr_b\}$
A	Set of all APs, i.e. $A = \{a_1, a_2, \ldots, a_n\}$	Te	Test set
l_k	k^{th} LP	D	Set of all devices, i.e., $D = \{d_1, d_2, \ldots, d_f\}$
L	Set of all LPs, i.e. $L = \{l_1, l_2, \ldots, l_o\}$	$\hat{op}_q^{CV}$	*10-fold-CV* results of classifier and $tr(q)$
$rssi_{i,j}$	RSSI value received from AP a_j in fingerprint r_i of train set	$\hat{op}_q^{Te}$	Classification results of classifier when $Tr(q)$ and Te are taken as train and test sets respectively
r_i	i^{th} fingerprint of train set, i.e. r_i= $\{rssi_{i,1}, rssi_{i,2}, \ldots, rssi_{i,n} : l_k\}$	χ	Decision of weighted ensemble classifier
$R^{(t,d)}$	Train set collected in different times *(t)* and device *(d)* represented by $R^{(t,d)} = \left\{r_1^{(t,d)}, r_2^{(t,d)}, \ldots, r_m^{(t,d)}\right\}^T$	$P(l_k)$	Probability of predicting a location l_k
$rssi'_{i,j}$	RSSI value received from AP a_j in fingerprint r'_i of test set	μ	Mean of RSSIs
r'_i	i^{th} fingerprint of test set, i.e. $r'_i = \left\{rssi'_{i,1}, rssi'_{i,2}, \ldots, rssi'_{i,n}\right\}$	σ^2	Variance of RSSIs
R'	Test set, i.e. $R' = \{r'_1, r'_2, \ldots, r'_{m'}\}^T$	e	Distance between actual location and predicted location
c	Base classifier	E^{avg}	Average localization error of each classifier

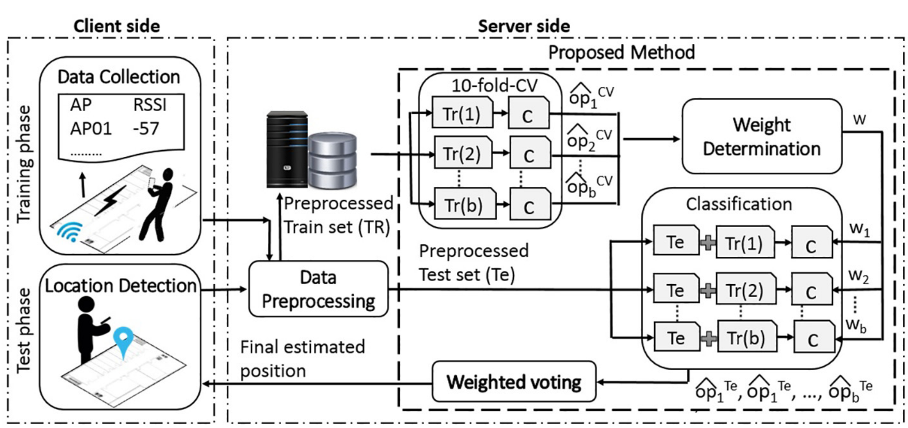


Fig. 1. Block diagram of the proposed framework.

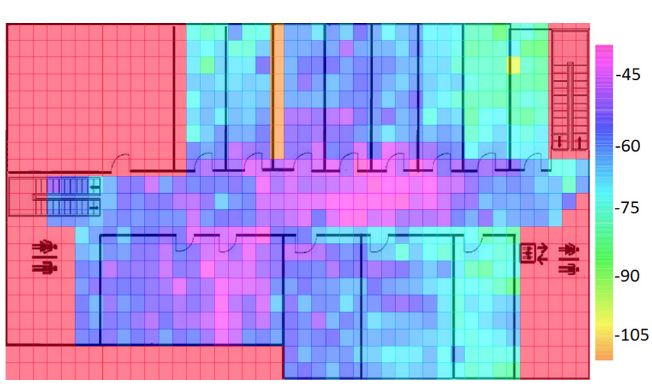


Fig. 2. RSSI map of AP03 according to our experimental zone. Colors represent signal strength in dBm. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

tion. Hence, these fingerprints are filtered out during pre-processing.

4.1. Proposed method

Our proposed method is summarized in Algorithm 1. Initially, the weights of all base classifiers, say W are not assigned as the importance of the classifier in classifying the datasets and hence addressing the contextual uncertainty is unknown. Hence, the whole train dataset (TR) is divided into b number of subsets say $TR = \{Tr(1), Tr(2), \ldots Tr(b)\}$ such that each train subset, $Tr(q)$, contain RSSI data of every AP collected by the devices d_g and d_h (where $g, h = 1 \ldots f$ and $g \neq h$) in different times in a day from all LPs. In the following step, 10-fold-cross-validation(CV) of each train set is performed using a state-of-the-art base classifier. In our proposed work, different train sets, $Tr(q)$s, are cross-validated using same base classifier, c. However, in Table 8 same train sets are cross-validated using different classifiers. The W is determined in Section 4.1.1 according to the individual performance, $\hat{op}_q^{CV}$ of the base classifiers on the train set for cross-validation. We have used a combination of identical and diverse classifiers that are trained to classify different training conditions. Hence, this step indicates which classifiers are better suited for the training environment.

In the testing phase, the base classification models developed by applying the base classifiers on respective train sets are used to classify the given test set. A weighted voting method which is discussed in Section 4.1.2 decides the output of the weighted ensemble classifier according to the $\hat{op}^{Te}$ and W.

Algorithm 1 WeightedEnsembleClassifier

```
input  : L: Set of all LPs
         A: Set of all APs
         D: Set of all devices
         W: Weights of all classifiers
         Te: Preprocessed Test set
output: χ: Decision of weighted ensemble
         classifier
b ← 0
f ← number of devices
if W = null then
    for g ← 1 ··· f do
        for h ← 1 ··· f and h ≠ g do
            Prepare train set Tr(d_g, d_h)
            b ← b + 1
            ôp_b^CV ← 10-fold-CV (c, Tr(d_g, d_h))
    W ← WeightDetermination
      (< ôp_1^CV, ôp_2^CV, ··· , ôp_b^CV >)
for q ← 1 ··· b do
    ôp_q^Te ← Classification(c, Tr(d_g, d_h), Te)
χ ← WeightedVoting (< ôp_1^Te, ôp_2^Te, ··· , ôp_b^Te >, W)
return χ
```

4.1.1. Weight determination

A trust model is built to decide whether decision of a classifier is correct or not; an opinion can be formed about it, which translates into degrees of belief (how much trustworthy the classifier is) or disbelief (how much imprecise the classifier is) as well as uncertainty (Li & Das, 2013) in case both belief and disbelief are lacking. This can be expressed mathematically as

$$bl + dl + un = 1 \tag{1}$$

Here bl, dl and un designate belief, disbelief and uncertainty respectively. Uncertainty is inherent in the system as the context in which the test data is collected is not known in advance; secondly, the context may change while collecting the test data. Hence, to quantify trust, that is, suitability of a base classifier or the decision making about a test set, parameters (bl, dl and un) are updated using Beta (α, β) distribution as in Li and Das (2013). Beta(1,1) represents the initial scenario where correct and incorrect predictions by a classifier are equally likely. To deal with uncertainty (as mentioned above), an approach proposed, leveraging on the *Dempster–Shafer belief theory* (Shafer, 1976) is adopted here to quantify the uncertainty of a classifier and is represented by,

$$un = 12\alpha\beta/(\alpha+\beta)^2(1+\alpha+\beta) \tag{2}$$

Furthermore, belief and disbelief can be calculated as follows,

$$bl = \alpha(1-un)/(\alpha+\beta) \tag{3}$$

$$dl = \beta(1-un)/(\alpha+\beta) \tag{4}$$

Whenever a classifier gives correct prediction for cross validation setting, α is incremented using Eq. (5) and bl is again calculated using Eq. (3). Otherwise, β is incremented and dl is updated using Eq. (4).

$$\alpha_{new} = \alpha_{old} + \delta \tag{5}$$

As in Li and Das (2013) the trust of a classifier is obtained by,

$$T_q = bl + \sigma \times un \tag{6}$$

Here, σ represents the relative atomicity. According to the principle of insufficient reasoning, the uncertainty of the atomic states can be split equally among these states. Since, we don't have any prior knowledge of the classifier's performance on the dataset, to provide an unbiased view, σ is set to 0.5 by default.

Algorithm 2 WeightDetermination

```
input  : < ôp_1^CV, ôp_2^CV, ··· , ôp_b^CV >: Set of
         predicted outputs of cross-validation
output: W: Weights of all classifiers
1  for q ← 1 ··· b do
2      α ← 1
3      β ← 1
4      δ ← 0.2
5      foreach instances in ôp_q^CV do
6          if l_actual = l_predict then
7              α ← α + δ
8          else
9              β ← β + δ
10     un ← 12αβ/(α + β)^2(1 + α + β)
11     bl ← α(1 − un)/(α + β)
12     T_q ← bl + 0.5 × un
13 for q ← 1 ··· b do
14     w_q ← T_q / Σ^q T_q
15 return W
```

The weight determination process of the classifiers is elaborated in Algorithm 2. α, β and δ are initialized with 1, 1 and 0.2 respectively following Beta distribution for obtaining weights of every classifier. The value of α is incremented by δ if the actual location and predicted location is same. Otherwise, β is incremented by δ. These incrementing process is repeated for every instances of $\hat{op}_q^{CV}$. After that, un, bl and trust of a classifier, T_q, are calculated using the Eqs. (2), (3) and (6) respectively. This process is repeated for every classifier. The weight of a classifier, w_q, is obtained by the ratio of the trust of that classifier and the total trust as shown in step 14 of Algorithm 2. Hence, the weights represent the relative importance of a classifier. Since, each classifier is trained for different context or combination of contexts, the classifier that can well classify instances for that context during cross-validation has eventually got better weight value.

4.1.2. Weighted voting

The weighted voting process is depicted in Algorithm 3. This algorithm takes the set of predicted outputs, $<\hat{op}_1^{Te}, \hat{op}_2^{Te}, \ldots, \hat{op}_b^{Te}>$, which contains the predicted locations of all classifiers for every instance of the test set and W as inputs. Suppose, the classifiers of b number of train sets predicted some locations say

$l_1, l_2, \ldots l_o$ (where $o \geqslant b$). Then, the probability of predicting a location, say, l_k is obtained by,

$$P(l_k) = \sum_{q=1}^{b} (w_q \times z_{q,k}), \tag{7}$$

where,

$$z_{q,k} = \begin{cases} 1 & \textit{if classifier of } q^{th} \textit{ train set predicts } l_k \\ 0 & \textit{otherwise} \end{cases}$$

Algorithm 3 WeightedVoting

input : $< \hat{op}_1^{Te}, \hat{op}_2^{Te}, \cdots, \hat{op}_b^{Te} >$: Set of predicted outputs of classification
W: Weights of all classifiers
output: χ: Voting decision

$o \leftarrow$ total number of LPs
for *every instances*, r'_i *in* Te **do**
  $P(l_k) \leftarrow 0 \; \forall \; k \in 1 \cdots o$
  for $q \leftarrow 1 \cdots b$ **do**
    for $k \leftarrow 1 \cdots o$ **do**
      if *predicted location is* l_k *for instance* r'_i *in* $\hat{op}_q^{Te}$ **then**
        $z_{q,k} \leftarrow 1$
      else
        $z_{q,k} \leftarrow 0$
      $P(l_k) \leftarrow P(l_k) + w_q \times z_{q,k}$
  $\chi \leftarrow l_k$ having $max(P(l_k))$
return χ

Further, the location having the maximum probability say $max(P(l_k))$ is chosen as the decision of our proposed ensemble classifier. Thus, not all classifiers trained for different sets of contexts are equally important for the ensemble. Rather, the classifiers that are tuned to the representative conditions (context) are given more importance in decision making. In this regard, the experimental results for evaluating the performance of proposed weighted ensemble classifier are shown in the following section.

5. Performance evaluation on JUIndoorLoc dataset

In this section, the performance of the proposed weighted ensemble classifier is evaluated using RSSI data along with its different features (such as, mean and variance). The train and test datasets are collected from four smart handheld devices ($d_1 \ldots d_4$). The experimental setup is further detailed in Table 3.

Four widely used supervised learning classifiers that are based on different classification techniques are used to analyze the performance of JUIndoorLoc dataset (Roy et al., 2019). We have written the ensemble implementation in Java language interfacing with *Weka 3.9* toolkit to execute the classifiers described as follows.

- *BayesNet*: This classification algorithm is based on the Bayes theorem for representing a probabilistic relationship between the variables and the conditional dependencies among those variables.
- *SVM*: The SVM (Support Vector Machine) classifier is based on statistical learning theory. As an output, it generates an optimal hyperplane with maximal margins to divide the samples into various categories. This classifiers helps to remove the samples over-fitting nature.
- *KNN*: This instance-based lazy learner search K elements in the train set which is nearest to a test sample. The class of the test sample is estimated based on the minimum distance.
- *K**: This is also an instance-based learner which uses entropy as a distance measure. This measure has several benefits for handling real-valued/symbolic attributes and missing values.

Accordingly, these individual base classifiers and their default parameters that are tuned for better accuracy are depicted in Table 4. As the proposed work is outlined corresponding to supervised learning, all the dataset instances are labeled.

5.1. Combination of same base classifier with different train sets

In the following experiments, the performance of the proposed weighted ensemble classifier is evaluated considering the same classifier with different train sets.

5.1.1. Considering RSSI as features

In the following experiments, RSSI values of available APs that are collected from every LPs of our experimental region are considered as features of train and test sets. Accordingly, the experiments are conducted first to verify whether the classifiers can accurately identify locations when training and testing are done for similar conditions as shown in Table 5. A 10-fold-cross-validation is performed on each dataset $Tr(d_g, d_h)$. The cross-validation procedure is iterated ten times and the dataset is partitioned into ten subsets. In each iteration or fold, classification is performed using one subset of data for testing and the remaining nine subsets for training. Finally, the mean of the classification accuracy which is obtained from each iteration is computed and termed as a cross-validation accuracy. The parameters of the classifiers are tuned according to Table 4. In Table 5, the cross-validation accuracy of each dataset is reported along with an estimation of its standard deviation. These accuracy values are found to be significant enough for identifying a location.

In the next experiment, the suitability of the base classifier is explored when train and test condition differs. In Table 6, $Te(D)$ has been used as a test set that consists of the instances collected

Table 3
Summary of experimental setup.

Device	Samsung Galaxy Tab 2 (d_1), Samsung Galaxy Tab E (d_2), Samsung Galaxy Tab 10 (d_3), Motorola Moto E 2nd Generation (d_4)
Train set	$Tr(d_1, d_2), Tr(d_1, d_3), Tr(d_1, d_4), Tr(d_2, d_3), Tr(d_2, d_4), Tr(d_3, d_4)$
Train set specification	Each train set $Tr(d_g, d_h)$ contains samples of all LP from device d_g and d_h
Test set	$Te(D)$ and $Te(d_g)$
Test set specification	$Te(D)$: samples for all LP collected from d_1, d_2, d_3 and d_4 (not present in any train set) $Te(d_g)$: samples for all LP collected either from d_1 or d_2 or d_3 or d_4 (not present in any train set)
No. of LP	96
Features considered	RSSI, mean of RSSI, variance of RSSI

Table 4
Base Classifiers (c_q) with their tuning parameters.

Classifier	Parameter	Parameter Details	Tuned Value
BayesNet	estimator	Finds conditional probability tables	SimpleEstimator
	searchAlgorithm	Searches the structure of bayes network	K2
SVM	SVMType	Set the type of SVM	C-SVC
	KernelType	Set the type of kernel function	linear
KNN	K	The number of nearest neighbour	5
	nearestNeighbour-SearchAlgorithm	The algorithm used for searching nearest neighbour	LinearNNSearch
K*	globalBlend	Determines the entropic auto blending	20

by all the four devices. Unfortunately, in many cases, the accuracies of the base classifiers reported in Table 6 are not better than the accuracy values shown in Table 5 as the timestamps and device configurations of the train and test sets are different. Interestingly, an ensemble of classifiers in this case is found to be effective. In all the cases ensemble shows improvement in estimating a location than individual classifiers through majority voting and weighted ensemble as shown in Table 6. Though, the results of the weighted ensemble are found to be better than majority voting which indicates the effectiveness of our proposed weighted ensemble classifier for indoor localization. Here, the weighted ensemble method and majority voting are applied to the six condition-specific classifiers formed using either BayesNet or SVM or KNN or K*. As the representative conditions are assigned more weight by the proposed algorithm and all conditions are not treated to be equally likely, the proposed algorithm gives a better result than simple majority voting.

5.1.2. Considering RSSI along with its mean and variance as features

Mean and variance of RSSI are more stable features than RSSI itself as it covers the short term temporal variation and multipath effect of radio signals. In the following experiments shown in Table 7a, RSSI values and its mean are considered as features of train sets, $Tr(d_g, d_h)$ and test set, $Te(d_1)$. Similarly, in Table 7b and c RSSI value along with it's variance and RSSI value along with its mean, variance are considered as features of train sets and test set respectively. A train set contains multiple samples that are collected by each device at different times. The mean and variance of RSSIs for each AP at a LP are calculated considering the RSSI values collected at different times using each device. Accordingly, in the train set, mean μ is calculated as follows:

$$\mu = \frac{\sum_{i=1}^{s} rssi_{i,j}}{s} \tag{8}$$

where, $rssi_{i,j}$ is the RSSIs of an AP, a_j, received by a device, d_g, at location, l_k, at different times and s is the number of instances received by d_g at l_k.

Similarly, in the train set, variance σ^2 is calculated as follows:

$$\sigma^2 = \frac{\sum_{i=1}^{s} \left(rssi_{i,j} - \mu\right)^2}{s} \tag{9}$$

Besides, the test set, $Te(d_1)$ contains RSSI fingerprints of available APs that are collected by the device, d_1. Considering the standard average walking speed, the RSSI capture time duration is chosen as 2 s. So, the mean and variance are calculated for every 2 s of RSSI fingerprint.

Hence, in the test set, mean μ' is calculated as follows:

$$\mu' = \frac{\sum_{i=1}^{s'} rssi'_{i,j}}{s'} \tag{10}$$

where, $rssi'_{i,j}$ is the RSSIs of an AP, a_j and s' is the number of instances received during the 2 s time interval.

Similarly, in the test set, variance σ'^2 is calculated as follows:

$$\sigma'^2 = \frac{\sum_{i=1}^{s'} \left(rssi'_{i,j} - \mu\right)^2}{s'} \tag{11}$$

Table 7 clearly depict that our weighted ensemble classifier performs better than individual classifiers when mean and/or variance of RSSI are considered as features.

5.2. Combination of different base classifiers with same train set

This experiment is conducted to investigate effectiveness of the proposed algorithm when a combination of different classifiers is applied. The results are shown in Table 8. The test set, $Te(d_g)$, of Table 8 contains RSSI data collected from either d_1 or d_2 or d_3 or d_4 while the corresponding train sets are formed by samples collected from a different set of devices. That is, the training and test devices are disjoint in each case. All combinations of train and test devices are avoided here for better clarity. This table shows that our proposed weighted ensemble classifier also achieved better localization accuracy than individual base classifiers when the test device is unknown with respect to the train set. Device d_4 is found to be a good training device that may handle device heterogeneity to a large extent. Devices d_1, d_3 and d_4 are found to correlate to one another well and hence better localization accuracy could be obtained when one of them is put as a test device while the train set is formed with the data collected using other two devices.

5.3. Performance of weighted ensemble classifier in terms of the error metric

Localization accuracy indicates the proportion of correct classification to misclassification but it does not differentiate between the extent of misclassification. However, in indoor localization, it is important to determine the distance between the predicted location from the actual location, that is, the extent of misclassification in order to assess the effectiveness of a scheme. Hence, the performance of our proposed weighted ensemble method is evaluated

Table 5
The mean of 10-fold cross-validation accuracy in % ($\pm$ Standard deviation) for each dataset that consider RSSIs as features.

Dataset	BayesNet	SVM	KNN	K*
$Tr(d_1, d_2)$	95.21($\pm$1.35)	97.26($\pm$0.96)	92.94($\pm$1.63)	97.53($\pm$0.87)
$Tr(d_1, d_3)$	90.40($\pm$2.09)	94.49($\pm$1.30)	86.30($\pm$1.99)	94.80($\pm$1.17)
$Tr(d_1, d_4)$	93.65($\pm$1.49)	94.16($\pm$1.52)	92.90($\pm$1.54)	96.39($\pm$1.20)
$Tr(d_2, d_3)$	94.23($\pm$1.36)	97.63($\pm$0.85)	91.79($\pm$1.74)	97.76($\pm$0.88)
$Tr(d_2, d_4)$	96.33($\pm$1.15)	98.72($\pm$0.69)	95.47($\pm$1.28)	98.88($\pm$0.65)
$Tr(d_3, d_4)$	93.84($\pm$1.46)	97.12($\pm$1.15)	92.73($\pm$1.56)	97.74($\pm$1.15)

Table 6
Performance (in terms of accuracy in %) of various base classifiers, majority voting and weighted ensemble considering RSSI as feature and test set, $Te(D)$.

Classifiers	Train sets						Majority voting	Weighted ensemble
	$Tr(d_1, d_2)$	$Tr(d_1, d_3)$	$Tr(d_1, d_4)$	$Tr(d_2, d_3)$	$Tr(d_2, d_4)$	$Tr(d_3, d_4)$		
BayesNet	72.98	63.51	82.95	78.41	82.45	83.43	92.63	94.87
SVM	77.86	67.27	86.40	84.54	86.91	86.49	89.18	93.76
KNN	74.23	62.53	81.68	76.19	80.92	82.17	92.53	93.53
K*	78.97	69.64	86.45	84.96	87.46	87.19	91.51	92.38

Table 7
Performance of various classifiers considering RSSI data along with its mean, variance as features of train sets and test set, $Te(d_1)$.

(a) Accuracies of different classifiers considering RSSI data along with its mean as features.

Classifiers	Train sets						Weighted ensemble
	$Tr(d_1, d_2)$	$Tr(d_1, d_3)$	$Tr(d_1, d_4)$	$Tr(d_2, d_3)$	$Tr(d_2, d_4)$	$Tr(d_3, d_4)$	
BayesNet	84.78	73.92	92.10	81.52	83.15	90.21	94.62
SVM	85.86	57.06	92.93	75.54	77.71	79.89	94.79
KNN	78.26	70.10	93.47	83.15	88.04	89.13	96.27
K*	88.58	67.39	93.19	82.06	84.23	88.58	96.82

(b) Accuracies of different classifiers considering RSSI data along with its variance as features.

Classifiers	Train sets						Weighted ensemble
	$Tr(d_1, d_2)$	$Tr(d_1, d_3)$	$Tr(d_1, d_4)$	$Tr(d_2, d_3)$	$Tr(d_2, d_4)$	$Tr(d_3, d_4)$	
BayesNet	78.80	75.54	91.30	80.97	82.06	85.32	95.88
SVM	76.08	45.10	85.86	79.34	83.69	81.52	90.35
KNN	76.63	71.19	91.84	82.06	88.04	91.30	95.48
K*	86.63	42.39	92.93	75.00	80.43	77.17	95.91

(c) Accuracies of different classifiers considering RSSI data along with its mean and variance as features.

Classifiers	Train sets						Weighted ensemble
	$Tr(d_1, d_2)$	$Tr(d_1, d_3)$	$Tr(d_1, d_4)$	$Tr(d_2, d_3)$	$Tr(d_2, d_4)$	$Tr(d_3, d_4)$	
BayesNet	85.86	71.19	93.10	86.41	88.58	93.47	97.10
SVM	77.17	45.65	86.95	79.34	83.69	82.60	90.36
KNN	79.89	66.84	94.56	87.50	93.47	94.56	97.82
K*	75.54	41.30	93.82	80.97	86.41	87.50	97.22

Table 8
Localization accuracies of various classifiers and weighted ensemble.

Train set	Test set	BayesNet	SVM	KNN	K*	Weighted ensemble
$Tr(d_1, d_2)$	$Te(d_4)$	75.05	75.00	77.60	77.42	84.42
$Tr(d_1, d_3)$	$Te(d_2)$	61.97	63.03	65.10	65.62	73.58
$Tr(d_1, d_4)$	$Te(d_3)$	89.08	89.15	83.33	88.97	93.78
$Tr(d_2, d_3)$	$Te(d_4)$	85.41	82.18	84.37	83.57	91.42
$Tr(d_2, d_4)$	$Te(d_1)$	85.58	85.66	86.89	86.96	92.64
$Tr(d_3, d_4)$	$Te(d_1)$	84.84	85.58	84.35	86.84	93.17

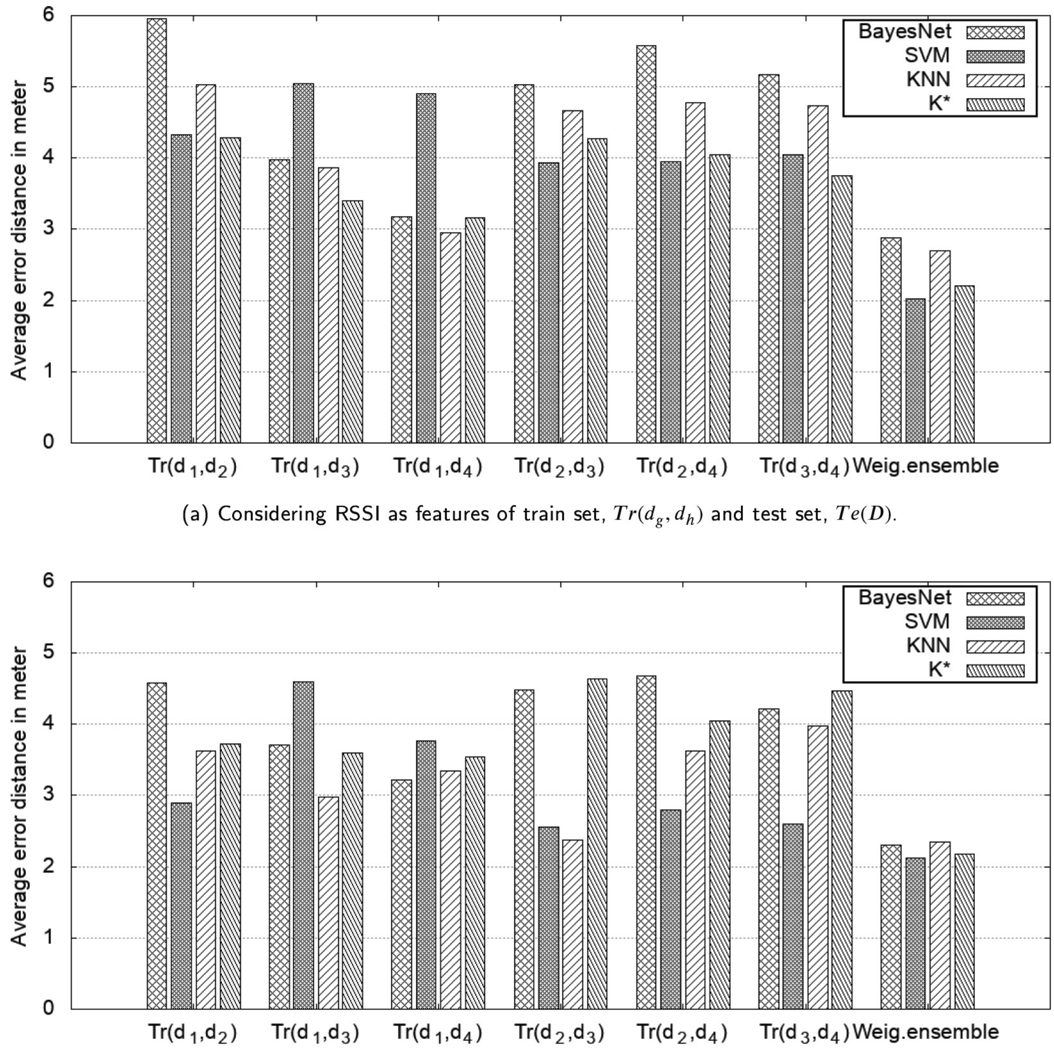
using an error metric. Here, the error e is the average distance between actual location (l^{act}) and predicted location (l^{pre}) for each instances of our test set. It has already been mentioned that each LP having a fixed size $1m \times 1m$ is represented by a 2-dimensional coordinate. So, l^{act} and l^{pre} are represented by (x^{act}, y^{act}) and (x^{pre}, y^{pre}) respectively. Accordingly, the average localization error, E^{avg}, of each classifier for every train test pair is obtained by

$$E^{avg} = \frac{1}{m'} \sum_{i=1}^{m'} e_i \tag{12}$$

where, m' is the total number of instances in test set and

$e_i = \sqrt{(x^{pre} - x^{act})^2 + (y^{pre} - y^{act})^2}$ if $l^{pre} \neq l^{act}$.

When, l^{act} and l^{pre} are same e_i is zero. Considering the different train sets and test set of Table 6, the average error distances for BayesNet, SVM, KNN, K* are shown in Fig. 3a that are calculated using Eq. (12) for each train and test pair. Similarly, the average localization errors of the weighted ensemble classifiers of Table 6 considering identical base classifier with different train sets are obtained. The corresponding localization accuracies are depicted in Table 6. Moreover, in Fig. 3b the train sets and test set of Table 7c are used for evaluating localization error. These train sets and test set contain RSSIs, mean and variance of every APs as features. Analogously, the localization errors of the weighted ensemble classifiers of Table 7c are depicted in Fig. 3b. Our proposed weighted ensemble method produces errors lower than any classifier as depicted in Fig. 3 as the localization accuracy gets improved. But, localization accuracy and error metric may not always follow the same trend as the accuracy calculation treats all misclassified instances equally. Thus, though accuracy of the weighted ensemble is found to be maximum when BayesNet classifier is used (Table 6) as the base classifier, the lowest localization error of 2 m is obtained when SVM is considered as the base classifier for the proposed weighted ensemble classifier as shown in Fig. 3a.



(a) Considering RSSI as features of train set, $Tr(d_g, d_h)$ and test set, $Te(D)$.

(b) Considering RSSI, mean and variance as features of train set, $Tr(d_g, d_h)$ and test set, $Te(d_1)$.

Fig. 3. Average error (in meter) for different classifiers and weighted ensemble method.

Furthermore, considering the different train sets and the test set of Table 6, the Cumulative Distribution Function (CDF) for BayesNet, SVM, KNN and K* are depicted in Fig. 4. Here, the weighted ensemble classifiers (as shown in Table 6) are the combination of identical base classifier with different train sets. The corresponding localization accuracies are depicted in Table 6. According to the Fig. 4, the performance of our proposed weighted ensemble classifier is better than the base classifiers with different train sets, $Tr(d_g, d_h)$ and test set, $Te(D)$. The 80% of the localization errors of the weighted ensemble classifiers considering BayesNet and KNN are within 5 m as depicted in Fig. 4a and c respectively. Besides, in case of the weighted ensemble classifiers considering SVM and K*, the 80% of the localization errors are within 3 m and 3.6 m respectively as shown in Fig. 4b and d.

5.4. Evaluating correlation between train and test set

We have investigated the extent of device heterogeneity that the proposed algorithm can handle and still achieve considerable accuracy. Correlation method is used here to identify how the RSSI vectors of individual APs in train set correlate to the test set.

The correlation between the RSSI vectors from a_j for train and test sets is calculated as follows:

$$\rho_{Tr(a_j),Te(a_j)} = \frac{\sum_{i=1}^{p}\left(rssi_{i,j} - \overline{rssi}\right)\left(rssi'_{i,j} - \overline{rssi'}\right)}{\sqrt{\sum_{i=1}^{p}\left(rssi_{i,j} - \overline{rssi}\right)^2 \sum_{i=1}^{p}\left(rssi'_{i,j} - \overline{rssi'}\right)^2}} \tag{13}$$

Here, $Tr(a_j)$ and $Te(a_j)$ represent the RSSI vectors of an AP, a_j, present in the train and test set respectively. The $rssi_{i,j}$ and $rssi'_{i,j}$ represent i^{th} fingerprint of a_j per LP of train and test set respectively. p is the total number of instances present in both sets. $\overline{rssi}= \frac{1}{p}\sum_{i=1}^{p} rssi_{i,j}$, represents the mean of all signal strengths of train set and analogously for $\overline{rssi'}$. This experiment is conducted for AP03 as most of the experimental region is covered by this AP (as shown in Fig. 2). Generally, the correlation value lies between -1 to +1.

Accordingly, the correlation of RSSI values from AP03 between each train set, $Tr(d_g, d_h)$, and test set, $Te(D)$, is depicted using Fig. 5. The localization accuracies of every classifiers when $Tr(d_g, d_h)$ and $Te(D)$ are taken as train and test set respectively (as described in Table 6) are also plotted in the same figure. Note that, the accuracies of weighted ensemble classifiers are better than any individual classifiers as reported in Table 6. It can be observed that for a classifier showing a good localization accuracy, the corresponding fingerprints from AP03 exhibit good positive

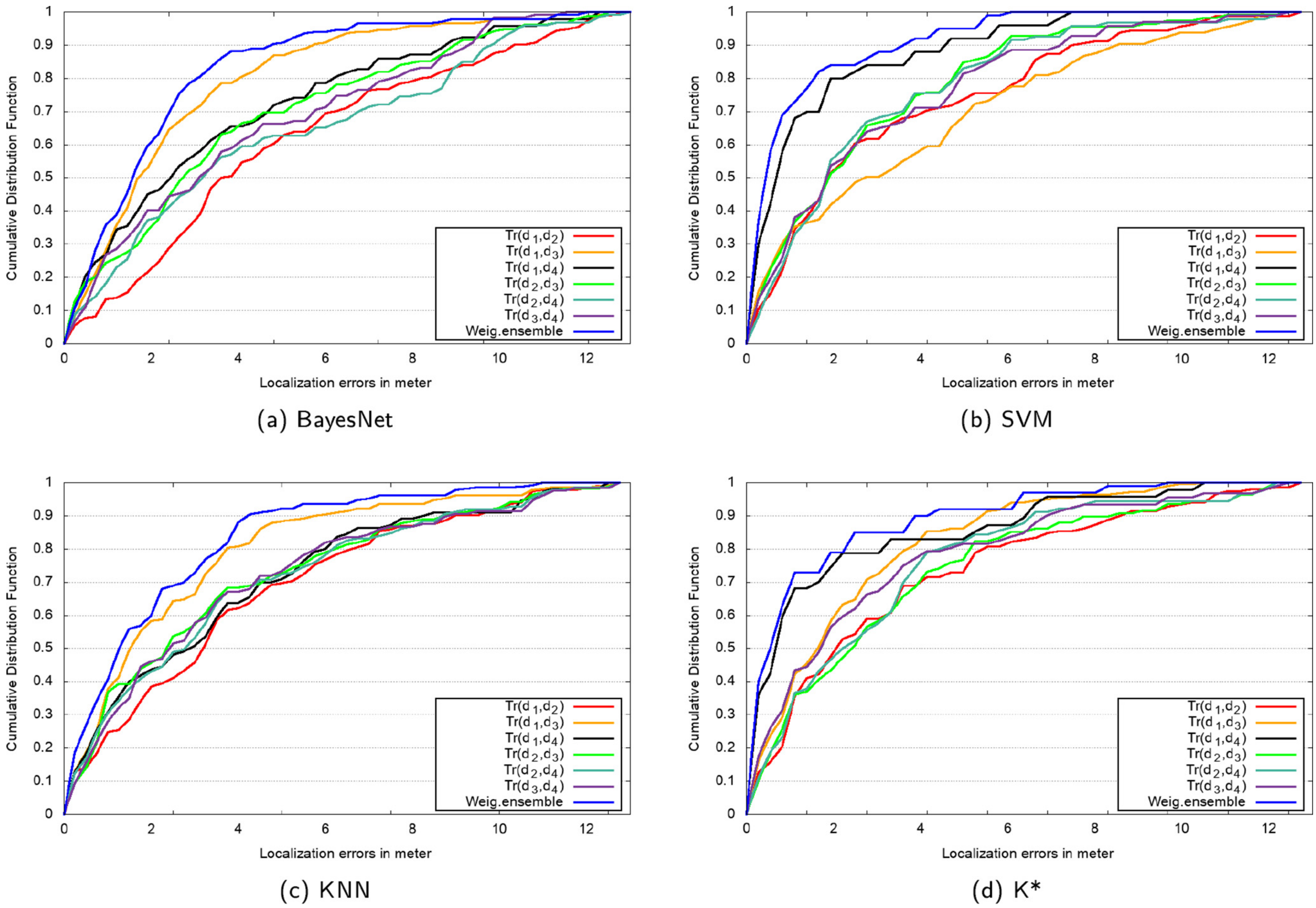


Fig. 4. CDFs of location estimation errors (in meter) for different base classifiers and weighted ensemble method considering RSSI as features of train sets, $Tr(d_g, d_h)$ and test set, $Te(D)$.

correlation. This validates the fact that if the RSSI fingerprint from train set and the test data of an unknown device show positive correlation, then the location of an unknown device (not used as a training device) can be predicted with considerable accuracy. Thus, the training devices should be chosen in such a way that they represent a diverse set and exhibit a good positive correlation with data collected from several test device configuration w.r.t. the infrastructure APs covering a region.

The next section highlights a short description of a benchmark dataset *UJIIndoorLoc* (Torres-Sospedra et al., 2014) and evaluates the performance of our proposed weighted ensemble classifier with the same dataset.

6. Performance evaluation on benchmark dataset

Among all the available databases in the literature of indoor localization, *UJIIndoorLoc* (Torres-Sospedra et al., 2014) is found to be the most comprehensive one and easily accessible from the UCI Machine Learning Repository. This database includes RSSI data of 520 APs captured from 3 buildings namely Building ID 0, 1 and 2 having 4, 4 and 5 floors respectively of the campus of Universitat Jaume I. The other features are user identifier, device identifier, time, GPS coordinates where the fingerprints are taken and positions in the indoor environment such as building, floor, space (lab, office, etc.), relative position (inside of a room or at corridor). RSSI measurements are taken from 933 reference points by 25 different kinds of smart hand-helds and 20 users. The entire database is divided into training and validation sets such that the two data-sets contain 19937 records and 1111 records respectively. Considering 1NN in conjunction to the Euclidean distance, they have reported 89.92% success rate (localization accuracy) and 7.92 m average localization error in their provided baseline.

In our work, the *UJIIndoorLoc* training dataset (Torres-Sospedra et al., 2014) including the fingerprints from 520 APs is used for analysis. We considered Building ID 2 from this training set where maximum number of phone models are used to collect data. Moreover, RSSI data from 4 phone models having device ID 6, 7, 11, 13 are considered for our experiments as these 4 devices are found to cover approximately similar location areas among all the 15 devices. Samples collected from each pair of devices are considered as the train set. On the contrary, a test set is generated with the samples of every said device in such a way that the train and test sets are disjoint. The performance of classifiers are reported in Table 9. Applying our weighted ensemble classifier on *UJIIndoorLoc* dataset (Torres-Sospedra et al., 2014) improves the localization accuracy subject to device heterogeneity as depicted in Table 9. The proposed weighted ensemble classifier is found to achieve better prediction than individual classifiers even for the benchmark dataset.

7. Conclusion and discussion

7.1. Conclusion

The objective of this paper is to propose a weighted ensemble classifier to address the problem of indoor localization subject to temporal and device heterogeneity. Brief discussions about the

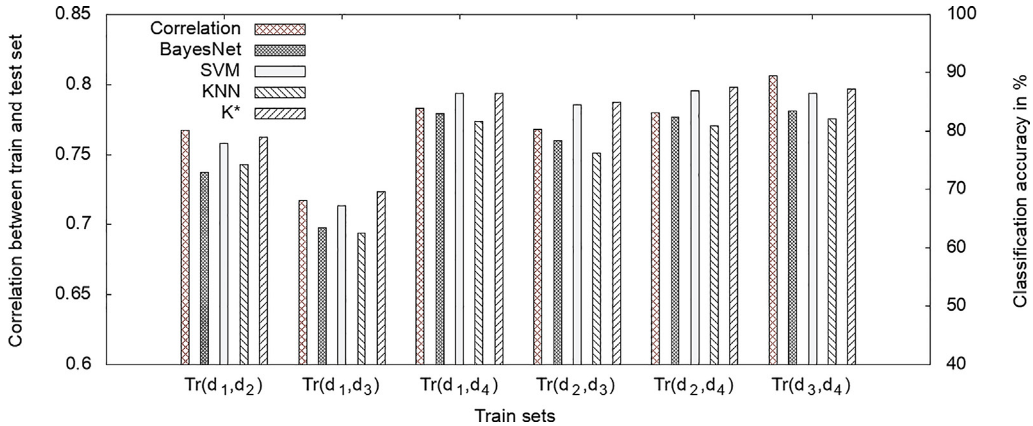


Fig. 5. Positive correlation between the fingerprint collected from AP03 for train and test set.

Table 9
Performance of proposed weighted ensemble classifier on *UJIIndoorLoc* dataset (Torres-Sospedra et al., 2014).

Classifiers	Train sets						Weighted ensemble
	$Tr(6,7)$	$Tr(6,11)$	$Tr(6,13)$	$Tr(7,11)$	$Tr(7,13)$	$Tr(11,13)$	
BayesNet	65.96	54.52	63.55	61.44	59.94	58.13	73.64
KNN	65.06	46.69	55.72	59.04	48.19	50.00	70.48

problem definition, data collection, data preprocessing and our proposed method are presented. *Dempster–Shafer belief theory* (Shafer, 1976) is used to determine the weights of the base classifiers according to their prediction capability and weighted voting is conducted to estimate an unknown location. The performance of our proposed weighted ensemble classifier is evaluated using our collected datasets from various perspectives. Mean and standard deviation of RSSI signals are also included as features in order to handle some temporal variation of WiFi signals. Besides, two different test cases are used for the analysis- test set containing instances captured by (i) all devices ($Te(D)$) and (ii) either d_1 or d_2 or d_3 or d_4 ($Te(d_g)$). Furthermore, *UJIIndoorLoc* dataset (Torres-Sospedra et al., 2014) is also considered for performance analysis. Our proposed weighted ensemble classifier is found to achieve almost 98% localization accuracy with 2 m localization error subject to the ground truth error of 1.41 m for 1×1 sq.m. LPs.

7.2. Discussion

We would like to discuss certain limitations of our proposed ensemble method. First, the proposed method is not adaptive to the change in feature space, thus, the significant change in the existing WiFi-infrastructure like addition, replacement, drop-off or shifting of WiFi *AP*s may affect the system performance. During emergency conditions, such as fire outbreaks, some WiFi *AP*s of one region may fail while some hotspots can be activated temporarily. In such cases, the training model of the system needs to be updated with the aforementioned changes. Thus, in future, a feature space mapping mechanism will be required to adapt the old training model to the new feature space.

Second, the proposed method requires sufficiently labeled data. Collection of a large number of fingerprints is labor-intensive process, so, crowd-sourcing come to rescue. However, the proposed mechanism cannot directly support crowd-sourced datasets where labeling is not generally precise. So, mechanisms should be devised to deduce the labeling of crowd-sourced fingerprints using existing well-labeled training dataset for the same experimental region. Then, the crowd-sourced dataset can be incorporated in the training model of the proposed ensemble classifier.

Third, the proposed method is implemented for the WiFi fingerprint dataset. However, the mechanism can be extended for other sensor datasets, such as Bluetooth beacons. Even, the mechanism can be extended to support a combination of different sensing data. The trust calculation of the classifiers would be revisited to handle such experiments.

Fourth, the work can be combined with smartphone-based human activity recognition works (Nandy, Saha, & Chowdhury, 2020) to build an expert system for indoor behavior tracking. For instance, people perform certain activities at a certain part of a building, i.e., climbing stairs, watching TV, etc. When a person falls or any mishap happens, it is important to locate that person first in the indoor space. An expert system for indoor group detection is another interesting dimension where our proposed method can be utilized through combining the location trails of multiple users. Multiple users may carry smartphones of different configurations, so, the proposed method would be beneficial to determine their location trails within an indoor environment.

CRediT authorship contribution statement

Priya Roy: Conceptualization, Methodology, Investigation, Software, Data curation, Writing - original draft. **Chandreyee Chowdhury:** Conceptualization, Methodology, Supervision, Writing - review & editing. **Mausam Kundu:** Visualization, Investigation. **Dip Ghosh:** Validation, Writing - review & editing. **Sanghamitra Bandyopadhyay:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research work is supported by the State Government Fellowship scheme of the School of Mobile Computing and Communication, Jadavpur University funded by Government of West Bengal, India.

References

Bahl, P., & Padmanabhan, V. N. (2000). RADAR: an in-building RF-based user location and tracking system. In *Proceedings 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). https://doi.org/10.1109/INFCOM.2000.832252.

Belmonte-Fernández, Ó., Montoliu, R., Torres-Sospedra, J., Sansano-Sansano, E., & Chia-Aguilar, D. (2018). A Radiosity-based method to avoid calibration for indoor positioning systems. *Expert Systems with Applications, 105*, 89–101. https://doi.org/10.1016/j.eswa.2018.03.054.

Bergeron, F., Bouchard, K., Gaboury, S., & Giroux, S. (2018). Tracking objects within a smart home. *Expert Systems with Applications, 113*, 428–442. https://doi.org/10.1016/j.eswa.2018.07.009.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*(2), 123–140. https://doi.org/10.1007/BF00058655.

Chen, Z., Zou, H., Jiang, H., Zhu, Q., Soh, Y. C., & Xie, L. (2015). Fusion of WiFi, smartphone sensors and landmarks using the Kalman filter for indoor localization. *Sensors, 15*(1), 715–732. https://doi.org/10.3390/s150100715.

Ghosh, D., Roy, P., Chowdhury, C., & Bandyopadhyay, S. (2016). An ensemble of condition based classifiers for indoor localization. In *2016 IEEE international conference on advanced networks and telecommunications systems (ANTS)* (pp. 1–6). https://doi.org/10.1109/ANTS.2016.7947872.

Kang, W., & Han, Y. (2015). SmartPDR: smartphone-based pedestrian dead reckoning for indoor localization. *IEEE Sensors Journal, 15*(5), 2906–2916. https://doi.org/10.1109/JSEN.2014.2382568.

Kim, Y., Shin, H., & Cha, H. (2012). Smartphone-based Wi-Fi pedestrian-tracking system tolerating the RSS variance problem. In *IEEE international conference on pervasive computing and communications (PerCom), 2012* (pp. 11–19). https://doi.org/10.1109/PerCom.2012.6199844.

Kuncheva, L. I. (2014). *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

Li, N., & Das, S. K. (2013). A trust-based framework for data forwarding in opportunistic networks. *Ad Hoc Networks, 11*(4), 1497–1509. https://doi.org/10.1016/j.adhoc.2011.01.018.

Liu, K., Zhang, H., Ng, J. K. Y., Xia, Y., Feng, L., Lee, V. C. S., & Son, S. H. (2018). Toward Low-Overhead Fingerprint-Based Indoor Localization via Transfer Learning: Design, Implementation, and Evaluation. *IEEE Transactions on Industrial Informatics, 14*(3), 898–908. https://doi.org/10.1109/TII.2017.2750240.

Mascharka, D. & Manley, E. (2015). Machine learning for indoor localization using mobile phone-based sensors, arXiv preprint arXiv:1505.06125. DOI: 10.1109/CCNC.2016.7444919..

Menéndez, P., Campomanes, C., Trawiński, K., & Alonso, J. M. (2011). Topology-based indoor localization by means of WiFi fingerprinting with a computational intelligent classifier. In *2011 11th international conference on intelligent systems design and applications (ISDA)* (pp. 1020–1025). https://doi.org/10.1109/ISDA.2011.6121792.

Nandy, A., Saha, J. & Chowdhury, C. (2020). Novel features for intensive human activity recognition based on wearable and smartphone sensors. *Microsystem Technologies*. DOI: 10.1007/s00542-019-04738-z..

Oussar, Y., Ahriz, I., Denby, B., & Dreyfus, G. (2011). Indoor localization based on cellular telephony RSSI fingerprints containing very large numbers of carriers. *EURASIP Journal on Wireless Communications and Networking, 1*, 81. https://doi.org/10.1186/1687-1499-2011-81.

Panov, P., & Džeroski, S. (2007). Combining bagging and random subspaces to create better ensembles. In *International symposium on intelligent data analysis* (pp. 118–129). https://doi.org/10.1007/978-3-540-74825-0_11.

Rai, A., Chintalapudi, K. K., Padmanabhan, V. N., & Sen, R. (2012). Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking* (pp. 293–304). https://doi.org/10.1145/2348543.2348580.

Rossi, M., Seiter, J., Amft, O., Buchmeier, S., & Tröster, G. (2013). RoomSense: an indoor positioning system for smartphones using active sound probing. In *Proceedings of the 4th Augmented Human International Conference* (pp. 89–95). https://doi.org/10.1145/2459236.2459252.

Roy, P., Chowdhury, C., Ghosh, D. & Bandyopadhyay, S. (2019). JUIndoorLoc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity. *Wireless Personal Communications*, 1–24. doi: 10.1007/s11277-019-06188-2..

Sánchez-Rodríguez, D., Hernández-Morera, P., Quinteiro, J. M., & Alonso-González, I. (2015). A low complexity system based on multiple weighted decision trees for indoor localization. *Sensors, 15*(6), 14809–14829. https://doi.org/10.3390/s150614809.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning, 5*(2), 197–227. https://doi.org/10.1007/BF00116037.

Shafer, G. (1976). *A mathematical theory of evidence* (Vol. 42). Princeton University Press..

Shih, C. Y., Chen, L. H., Chen, G. H., Wu, E. H. K., & Jin, M. H. (2012). Intelligent radio map management for future WLAN indoor location fingerprinting. In *2012 IEEE wireless communications and networking conference (WCNC)* (pp. 2769–2773). https://doi.org/10.1109/WCNC.2012.6214271.

Singh, V., Aggarwal, G., & Ujwal, B. (2018). Ensemble based real-time indoor localization using stray WiFi signal. In *2018 IEEE international conference on consumer electronics (ICCE)* (pp. 1–5). https://doi.org/10.1109/ICCE.2018.8326317.

Sun, Z., Chen, Y., Qi, J. & Liu, J. (2008). Adaptive localization through transfer learning in indoor Wi-Fi environment. In *Seventh international conference on machine learning and applications, 2008. ICMLA'08* (pp. 331–336). DOI: 10.1109/ICMLA.2008.53..

Taniuchi, D., & Maekawa, T. (2014). Robust Wi-Fi based indoor positioning with ensemble learning. In *2014 IEEE 10th international conference on wireless and mobile computing, networking and communications (WIMOB)* (pp. 592–597). https://doi.org/10.1109/WiMOB.2014.6962230.

Torres-Sospedra, J., Montoliu, R., Martínez-Usó, A., Avariento, J. P., Arnau, T. J., Benedito-Bordonau, M., & Huerta, J. (2014). UJIIndoorLoc: a new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems. In *2014 International conference on indoor positioning and indoor navigation (IPIN)* (pp. 261–270). https://doi.org/10.1109/IPIN.2014.7275492.

Trawiński, K., Alonso, J. M., & Hernández, N. (2013). A multiclassifier approach for topology-based WiFi indoor localization. *Soft Computing, 17*(10), 1817–1831. https://doi.org/10.1007/s00500-013-1019-5.

Wang, X., Gao, L., & Mao, S. (2016). CSI phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet of Things Journal, 3*(6), 1113–1123. https://doi.org/10.1109/JIOT.2016.2558659.

Wang, H., Sen, S., Elgohary, A., Farid, M., Youssef, M., & Choudhury, R. R. (2012). No need to war-drive: unsupervised indoor localization. In *Proceedings of the 10th international conference on mobile systems, applications, and services* (pp. 197–210). https://doi.org/10.1145/2307636.2307655.

Webb, G. I. (1999). Decision tree grafting from the all-tests-but-one partition. *IJCAI, 2*, 702–707.

Yang, Z., Wu, C. & Liu, Y. (2012). Locating in fingerprint space: wireless indoor localization with little human intervention. In *Proceedings of the 18th annual international conference on mobile computing and networking* (pp. 269–280). DOI: 10.1145/2348543.2348578..

Yim, J. (2008). Introducing a decision tree-based indoor positioning technique. *Expert Systems with Applications, 34*(2), 1296–1302. https://doi.org/10.1016/j.eswa.2006.12.028.

Youssef, M., & Agrawala, A. (2005). The horus WLAN location determination system. In *Proceedings of the 3rd international conference on mobile systems, applications, and services* (pp. 205–218). https://doi.org/10.1145/1067170.1067193.

YuFeng, JiangMinghua, LiangJing, QinXiao, HuMing, PengTao & HuXinrong. (2014). An improved indoor localization of WiFi based on support vector machines, *International Journal of Future Generation Communication and Networking 7*(5) (2014) 191–206..

Zhou, M., Tang, Y., Tian, Z., Xie, L., & Nie, W. (2017). Robust neighborhood graphing for semi-supervised indoor localization with light-loaded location fingerprinting. *IEEE Internet of Things Journal, 5*(5), 3378–3387. https://doi.org/10.1109/JIOT.2017.2775199.