

A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength

Khuong An Nguyen

To cite this article: Khuong An Nguyen (2017) A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength, Journal of Information and Telecommunication, 1:1, 41-65, DOI: [10.1080/24751839.2017.1295659](https://doi.org/10.1080/24751839.2017.1295659)

To link to this article: <https://doi.org/10.1080/24751839.2017.1295659>



© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group on behalf of Ton Duc Thang University.



Published online: 03 Mar 2017.



Submit your article to this journal [↗](#)



Article views: 2353



View related articles [↗](#)



View Crossmark data [↗](#)

A performance guaranteed indoor positioning system using conformal prediction and the WiFi signal strength

Khuong An Nguyen 

Department of Computer Science, Royal Holloway, University of London, London, UK

ABSTRACT

Indoor navigation provides the positioning service to the indoor users, where the GPS coverage is not available. The challenges for most signal-based indoor positioning systems are the unpredictable signal propagation caused by the complex building interiors, and the dynamic of the environment caused by the peoples' movements. However, most existing systems made no assumption about the quality of their predictions, which is crucial in such noisy indoor environment. To address this challenge, this article proposes a confidence measure to reflect the uncertainty of the positioning prediction. More importantly, the users may control the size of the prediction set by setting the confidence level tailoring to their personal requirement. The proposed approach in this article has been validated in three real office buildings with challenging indoor environments, which indicated that it performed up to 20% more accurate than traditional Naïve Bayes and Weighted K-nearest neighbours (W-KNN) algorithms.

ARTICLE HISTORY

Received 15 August 2016
Accepted 31 January 2017

KEYWORDS

Indoor positioning;
conformal prediction; WiFi
fingerprinting

1. Introduction

Global Navigation Satellite Systems (GNSS) such as GPS have been successfully deployed in the past two decades, and are indispensable for outdoor navigation. However, people spend most of their times indoors, where limited or no GNSS service is available at all, because the satellites' signals are too weak to penetrate the building. More importantly, GPS cannot provide the indoor users with the positioning accuracy they need for room-level tracking.

Many systems have attempted to tackle this challenging problem in recent years. Overall, based on how the systems interact with the indoor environment, they can be divided into two broad categories, which are infrastructure-based systems and infrastructure-free systems. With the former, the system relies on a piece of hardware that needs to be installed into the building. These hardware are often specifically designed for indoor positioning, and are costly to install and to maintain. In contrast, the latter are self-contained and require no additional changes to the indoor environment. These systems piggyback on top of the structures that already exist in the building (e.g. the WiFi network) to provide the positioning service. Overall, the infrastructure-based systems

CONTACT Khuong An Nguyen  khuong.nguyen.2007@live.rhul.ac.uk

© 2017 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group on behalf of Ton Duc Thang University. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

often provide excellent positioning accuracy, at the expense of additional hardware. Whereas, the advantage of the infrastructure-free approach is the ease of deployment, at the expense of a lower positioning accuracy.

This article opted for the infrastructure-free option for maximum ubiquity. Amongst those is WiFi fingerprinting, which is considered one of the most effective WiFi-based indoor tracking methods to date, and will serve as the foundation for the system in this article. Fingerprinting uses a training database to estimate the user position, which will be discussed in detail in this article. However, most existing fingerprint-based systems made no assumption about the quality of their predictions. To address this challenge, this article proposes a confidence measure to reflect the uncertainty of the positioning prediction. The impact of such confidence measure will be evaluated in real-world data. Overall, the contributions of the article are:

- A confidence level is introduced to represent the uncertainty of the positioning prediction.
- It provides the users with a flexible tool to control the size of the prediction set, tailoring to their own application.
- Two WiFi fingerprinting datasets collected by the author in real office environments are introduced for further researches.

2. Location fingerprinting

Compared to the outdoor space, the indoor environment is more challenging for most wireless signal-based technologies to work reliably. While the standard satellite signals such as GPS struggle to penetrate the building structure, other indoor wireless technologies such as WiFi or Bluetooth could not rely on their standard properties such as the time-of-flight, the angle-of-arrival or the received signal strength (RSS) measure to calculate the distance between two positions, because of the complex layout of the building. As the wireless signals travel in the air, they reflect from the metal objects, diffract around sharp corners, scatter off the walls, floors and ceilings, which result in multiple copies of the original signal travelling in different directions. When two in-phase waves of the signal meet, constructive interference forms a new stronger wave of signal. In contrast, two out-phase waves will cancel each other out, resulting in a weaker version. The receiving signal at the end user is a combination of these distorted products. This phenomenon is known as the signal multi-path problem (Sen, Lee, Kim, & Congdon, 2013). Furthermore, the building is often crowded with many users who move around to create a harsh and dynamic environment.

Fingerprinting, however, uses this challenging environment as its core function. The indoor positions are manually calibrated to capture the full dynamic of the signal characteristic at each position. Thus, the more diverse and tricky the signal propagation is, the more unique the 'location fingerprint' is. Those training fingerprints will be matched against the user's real-time fingerprint to estimate his position. More details about the fingerprinting processes will be discussed in the next section.

Fingerprinting was originally proposed with the WiFi RSS recorded by a laptop from the nearby WiFi access points (APs) (Bahl & Padmanabhan, 2000; Wang, Zhou, Yang, & Mo, 2015; Weber, Birkel, Collmann, & Engelbrecht, 2010; Youssef & Agrawala, 2005). Since

then, other wireless signals have been tested with fingerprinting (e.g. Bluetooth, FM, Cellular) (Chen et al., 2013; Huang & Chan, 2011; Ibrahim & Youssef, 2013; Subhan, Hasbullah, Rozyyev, & Bakhsh, 2011). This article decides to use the WiFi RSS and a smart phone as the main components to perform fingerprinting for two reasons. Firstly, the WiFi AP is becoming a norm, and it is common to have many WiFi APs indoors. Furthermore, they are starting to move beyond the buildings to provide a seamless transitional coverage from the indoor space to the outdoor space. Secondly, most people carry a smart phone with them wherever they are. These devices have the computing power and the storage of a mini computer, as well as including a WiFi receiver.

3. The two phases of fingerprinting

This section discusses the inner processes of fingerprinting and the difficulties for each step (see Figure 1).

3.1. The off-line phase of fingerprinting

This phase is also known as the training or planning phase. The purpose of this phase is to generate a training database (i.e. the fingerprinting database) to reflect how the signal propagates inside the building. For WiFi fingerprinting, it is normally done by an expert holding a WiFi-enabled device (e.g. smart phone, laptop) and walking around the building to record the WiFi RSS at different training positions. Some key issues for the experts to take into considerations are:

- *How granular the tracking space is?* The higher the granularity is, the more training positions the expert needs to cover. The tracking zone is normally divided into a metre-by-metre grid for the ease of planning.
- *How often the measures are taken at each position?* The WiFi signal is noisy, thus, it is recommended to capture the full histogram distribution of the WiFi RSS by measuring them repeatedly at each training position.
- *How to label the signal data?* Most fingerprint-based systems used their own co-ordinate metric (e.g. the Earth's latitude and longitude) to label the training WiFi RSS. Others used a more human-readable presentation (e.g. room number).

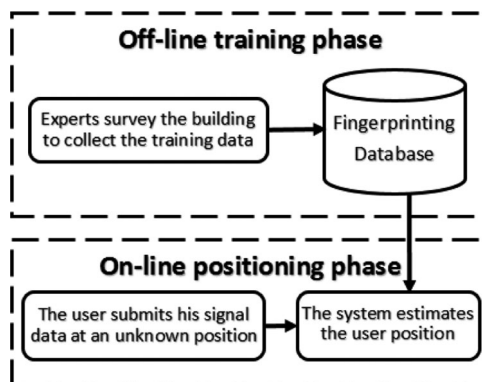


Figure 1. The two phases of fingerprinting.

The first two plans directly decide the type of the fingerprinting algorithm to be used in the positioning phase. In most cases, they also affect the performance accuracy of the system. For instance, it may not be reasonable to expect a low-grained fingerprinting dataset with too few signal measures to perform well with any algorithm.

The challenges at this phase are the sheer amount of the building space to be meticulously surveyed, the time it takes to perform such process, and the lack of physical reference for the training positions. Some works have attempted to alleviate the first two challenges with a robot (Haverinen & Kemppainen, 2009; Jung, Lee, Kim, Park, & Myung, 2013; Nguyen, 2011). However, the robot does have its own problem of knowing where it exactly is in the building. Others relied on crowd-sourcing to train the system automatically (Brabham, 2008; Chaudhry, 2013). However, the lack of ground-truth references is a major challenge to label the crowd-sourced data. Other researchers set up landmarks in the building, where the users can contribute manually (Ledlie et al., 2012; Lee & Han, 2012). Others used an independent tracking system (e.g. Active Bat) to provide the location references for fingerprinting (Want, Hopper, Falcao, & Gibbons, 1992). Despite the above challenges, this training phase normally needs to be performed once at the beginning for every building.

3.2. The on-line phase of fingerprinting

This phase is known as the positioning phase or the estimation phase. It will be performed whenever the user wishes to discover his whereabouts. For WiFi fingerprinting, the user needs to carry a WiFi-enabled device (e.g. a smart phone) to measure the WiFi signal at his current unknown position. Given this real-time WiFi reading, the system looks up the training database generated in the previous phase to estimate the user position based on the surveyed co-ordinate labels.

The challenge at this phase is what type of algorithm to estimate the user position. Clearly, since there is no guarantee that the training database is well-generated in the last phase, it is beneficial to introduce a confidence level to represent the uncertainty of the positioning prediction.

3.3. Modelling the WiFi fingerprinting problem

Without loss of generality, the location fingerprinting problem is formally modelled as follows. The database indexes M training examples $T_i = (\overrightarrow{RSS}_i, \overrightarrow{L}_i)$ ($1 \leq i \leq M$). Each example T_i represents a training position, where \overrightarrow{RSS}_i is the WiFi RSS vector observed at that position, and $\overrightarrow{L}_i = (d_x^i, d_y^i)$ is the two-dimensional Cartesian label of the position. The WiFi RSS vector contains all the individual RSS from N nearby APs, $\overrightarrow{RSS}_i = (AP_1^i, \dots, AP_N^i)$, where AP_j^i is the RSS received from the AP j th ($1 \leq j \leq N$). It is possible that there are duplicated \overrightarrow{L}_i in the training set, because the WiFi RSS are captured several times at the same position. Table 1 illustrates such training database. There may be other features attached to each training example such as the user orientation, the time of measure, the calibration device.

The task is, given a WiFi RSS vector $\overrightarrow{RSS}_u = (AP_1^u, \dots, AP_N^u)$ at an unknown position u lying somewhere in the tracking zone, the system estimates the Cartesian label $\overrightarrow{L}_u = (d_x^u, d_y^u)$ for this position.

Table 1. An example of the fingerprinting database.

Position label	WiFi RSS measurement
...	...
(51, 136)	(-57, -41, -62, -59, -86, -91)
(51, 136)	(-57, -42, -60, -58, -87, -93)
(51, 146)	(-59, -36, -65, -58, -82, -95)
(51, 156)	(-56, -38, -69, -58, -84, -93)
(51, 156)	(-54, -35, -69, -57, -84, -95)
(51, 166)	(-63, -32, -73, -57, -87, -99)
(51, 176)	(-67, -33, -68, -60, -91, -99)
...	...

4. A comparative review of fingerprinting’s performance

Fingerprinting is a decade mature technology. How is it keeping up with other infrastructure-based systems, and especially with the latest trends such as inertial-based tracking in the same infrastructure-free category? This section assesses the performance accuracy of WiFi fingerprinting in a Microsoft competition where all contestants were ranked under the same test domain. The underlying algorithms of fingerprinting are then analysed individually to understand which options are suitable to perform fingerprinting.

4.1. Performance review of fingerprinting at the Microsoft IPSN competition

Since 2014, Microsoft have been organizing a yearly indoor positioning competition, where the competitors from the academia, the industry and start-ups come together to evaluate their latest technologies in a realistic, unfamiliar environment. For Microsoft IPSN 2014, the 2500 ft² evaluation area includes two rooms and the hallway surrounding them.¹ It is interesting to see how well fingerprint-based systems performed in this same test environment with other systems. There were two pure WiFi fingerprint-based only systems in this competition (MapUme and Nanyang). Inertial-based tracking dominated the selection of the remaining contestants in the infrastructure-free category. There was no WiFi fingerprint based only systems enrolled in the following two subsequent years.

Figure 2 compares the performance accuracy of the systems enrolled in this competition, where WiFi fingerprinting ranked 2nd and 7th with the positioning accuracy of 1.6 and 2.22 m, respectively, out of 22 contestants including some best papers at the international conferences. In particular, the same fingerprinting systems came first and third amongst all nine infrastructure-free systems. The big surprise was that some hybrid fingerprinting and inertial tracking systems performed less accurate than these two pure fingerprint-based systems. Perhaps the sensor noises from the mobile phone degrade the positioning accuracy of those hybrid systems. Overall, this is a highly encouraging result for fingerprint-based research.

It is worth noting that many of the systems participating in this competition came from the industry and did not reveal much of their underlying algorithms. The next section assesses the impact of different machine learning algorithms for fingerprinting.

4.2. Performance review of the machine learning approaches to WiFi fingerprinting

This section assesses the most popular machine learning algorithms for fingerprinting in the literature. In particular, they are Weighted K-nearest neighbours (W-KNN), Naïve Bayes,

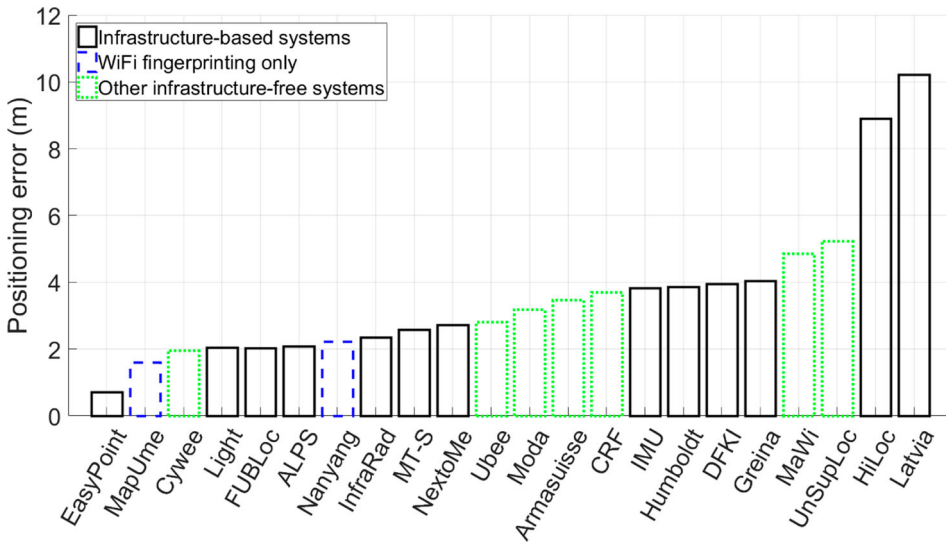


Figure 2. Performance accuracy of fingerprinting at Microsoft IPSN 2014 competition.

neural network and histogram. The histogram method simply compares the WiFi RSS distribution of both the test position and the training position. The neural network approach uses a mathematical model to recognize the patterns in the database. Once the neural network has been trained on the examples of the fingerprinting database, it is able to predict by detecting similar patterns for the new samples. The experiments were conducted and reported under the same test domain by three independent review papers (Dawes & Chin, 2011; Honkavirta, Perälä, Ali-Löytty, and Piché, 2009; Lin & Lin, 2005). To emphasize on the performance of the algorithms, all reviewed systems are WiFi RSS based only. No other technique apart from fingerprinting was employed in these systems.

The first review was conducted in a corridor of 24.6 by 17.6 m, with at least 5 nearby WiFi APs (Lin & Lin, 2005). A total of 84 training positions were recorded, with 100 readings of the WiFi RSS per position. The training dataset's granularity was 1 m. Figure 3(a) demonstrated that W-KNN had the most accurate performance at 3.1 m, 95% probability. However, it was suggested that the higher the number of training examples per location is, the more performance gain the Naïve Bayes approach may benefit from (see Figure 3(b)).

The second review was conducted on a floor of 2160 m², which is five times larger than the first review's (Dawes & Chin, 2011). However, the training points were much sparser with a granularity of 6.2 m covering 56 positions. This sparse training set was compensated by a denser histogram of 224 WiFi RSS per position, covering all 4 orientations (N/W/S/E). This review compared 17 variations of KNN and Naïve Bayes. Giving such strong RSS coverage for each training position, Naïve Bayes was expected to triumph. However, W-KNN slightly edged out again at 2 m positioning error on average, compared to that of 2.3 m for the Naïve Bayes approach (having applied the filter mode) (see Table 2). This review noticed that the Naïve Bayes approach achieved competitive performance with fewer APs than W-KNN.

The third review was conducted over 3 floors in a university building with a total of 177 training positions, although it was unclear what the granularity of these training points was (Honkavirta et al., 2009). The WiFi RSS was recorded repeatedly for 60 s at each

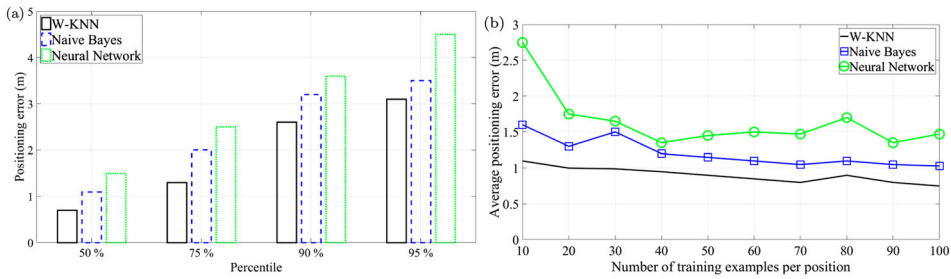


Figure 3. Performance accuracy of W-KNN, Naïve Bayes and neural network, reported in first review (Lin & Lin, 2005). (a) Performance accuracy of the three algorithms and (b) impact on the number of training examples.

position, which is expected to collect about half the amount of the WiFi RSS as in the second review. This review compared the performance accuracy of W-KNN, Naïve Bayes and the histogram method. The histogram approach compares the WiFi RSS distribution of both the test position and the training position, using Kullback–Leibler, Lissack–Fu (Bishop, 2006; Dawes & Chin, 2011). This method may not be applicable in the real-world, where it is difficult to obtain more than one reading per location for the moving users in real-time. In this review, Naïve Bayes came up on top at 12.3 m positioning error, 95% probability, while W-KNN was just slightly behind at 13.7 m, 95% probability (Table 3).

In summary, all three review papers suggested that with only the WiFi RSS as the measurement metric, many complex algorithms may not perform as well as simpler ones. Despite its simplicity, W-KNN excelled in most fingerprinting reviews. It is worth noting that the MapUme system that came second over 22 contestants in the Microsoft IPSN 2014 competition also employed W-KNN as the main underlying algorithm. However, the Naïve Bayes approach improves its accuracy as the number of training examples per location is high, which is an indication that beside the WiFi RSS, additional information will be needed to enhance the performance of fingerprinting further.

5. A confidence machine approach to fingerprinting

The concept of confidence machine is that a prediction made by any learning algorithm should be governed by a confidence parameter measuring the belief of the algorithm on this prediction. The confidence learning algorithm that is used in this article is called

Table 2. Performance accuracy of nearest neighbours and Naïve Bayes, report in the second review (Dawes & Chin, 2011).

Algorithms	Prediction error (m)
Weighted K-nearest neighbours, with best orientation	2.2
Naïve Bayes with filtering	2.3
Average K-nearest neighbours	2.7
Weighted K-nearest neighbours, closest to all four orientations	3.2
Naïve Bayes	3.3
1-Nearest neighbour	5.3

Table 3. Performance accuracy of nearest neighbours and Naïve Bayes, report in the third review (Honkavirta et al., 2009).

Algorithms	Prediction error (m)	Median (m)	Max (m)	95% (m)
Naïve Bayes	5.4	4.1	98.6	12.3
Weighted K-nearest neighbours	5.6	4.4	119.6	13.7
Histogram	7.0	5.0	106.0	19.4

conformal prediction (CP), which produces a set of predictions, given a new sample, a training database and a confidence level (Shafer & Vovk, 2008; Vovk, Gammerman, & Shafer, 2005). For instance, given a 95% confidence level, a training database with three training examples $\{t_1, t_2, t_3\}$ and a new sample s , the set of predictions that CP produces is $\{t_1, t_2\}$, which can be interpreted as ‘I am 95% confident that “s” belongs to this prediction set. However, there is 5% chance that I may be wrong’. It has been mathematically proven that in the on-line setting, where many predictions are performed one after another with the new samples added into the training data after each prediction, CP is correct in the sense of maintaining such error rate. Even with off-line learning using the same training database to make predictions, the confidence level offered by CP can adjust the size of the prediction set. For instance, using the same above example, given a 100% confidence level and a new sample s , CP produces a prediction set $\{t_1, t_2, t_3\}$, which is actually the whole training data. This is interpreted as ‘I am 100% confident that “s” must belong to this prediction set, and there is no chance that I can be wrong’. This prediction set is not too useful, however, although it is statistically correct. As the confidence level decreases, CP automatically reduces the size of the prediction sets accordingly. Ideally, it is preferred to have a high confidence level with a small prediction set, which may be achieved with different nonconformity measures. More details of how this process works will be discussed in the next section. Overall, the use of confidence machine offers the following benefits.

- Each prediction has an associated confidence level to express how likely the prediction is correct.
- The predictions produced by CP are statistically correct under the on-line setting.
- The confidence level can be adjusted to produce a bigger or a smaller prediction set.

CP has been successfully tested in some real-world applications such as cancer diagnosis, image analysis and network traffic prediction (Bellotti, Luo, Gammerman, Van Delft, & Saha, 2005; Dashevskiy & Luo, 2009; Lambrou, Papadopoulos, & Gammerman, 2009; Lambrou et al., 2010). This is the first time that CP is used for the indoor positioning research. Empirical studies in this article will demonstrate that the proposed algorithms with CP perform up to 20% more accurate than other systems without confidence machine.

Without loss of generality, CP fingerprinting is formally modelled as follows. Given the training database $T = (T_1, \dots, T_M)$ where each training example $T_i = (\overrightarrow{RSS}_i, \overrightarrow{L}_i)$ maps the WiFi RSS vector \overrightarrow{RSS}_i to its Cartesian co-ordinate $\overrightarrow{L}_i = (d_x^i, d_y^i)$ ($1 \leq i \leq M$), a new sample $T_{M+1} = (\overrightarrow{RSS}_{M+1}, \overrightarrow{L}_{M+1})$ at an unknown location \overrightarrow{L}_{M+1} , with known $\overrightarrow{RSS}_{M+1}$, and a confidence level $(1 - \xi)$, with ξ is the significance level, CP will find a set of training examples for the new sample T_{M+1} in the following three steps.

Step 1: Firstly, the new sample T_{M+1} is added into the training database, with the label L_{M+1} being assumed as one of the training labels \vec{L}_i ($1 \leq i \leq M$). Then, given a nonconformity function $A(T, T_i)$ (which will be discussed soon, for now such function is assumed to exist), the nonconformity measure $\alpha_i = A(\{T_1, \dots, T_{i-1}, T_{i+1}, \dots, T_{M+1}\}, T_i)$, ($1 \leq i \leq M + 1$) is calculated for every training example. This α score demonstrates the difference between the example T_i and all other training examples including the new sample. Intuitively, the algorithm wants to observe how well the new sample with the assumed label fits into the whole training database. However, for this reason, since this nonconformity function may be designed in any way the readers want, the score α_{M+1} of the new sample, by itself does not tell how similar A finds T_{M+1} to be. For that purpose, α_{M+1} needs to be compared to other α_i .

Step 2: Thus, the second step uses the α_i ($1 \leq i \leq M$) of all training examples to calculate a p -value which represents the adaptability of the assumed label L for the new sample, as follows (Shafer & Vovk, 2008; Vovk et al., 2005).

$$p(L) = \frac{\#\{j = 1, \dots, M + 1 : \alpha_j(L) \geq \alpha_{M+1}(L)\}}{M + 1}. \tag{1}$$

This p -value lies within $1/(M + 1)$ and 1 to indicate the fraction of the training examples that are similar to the new sample. The higher the p -value is, the better it indicates that the assumed Cartesian label L helps the new sample T_{M+1} fit into the training data. Otherwise, the lower the p -value is, which means α_{M+1} is much bigger than the majority of other α_i , the stronger the indication is that the assumed label L makes this new sample an outlier. This process is repeated for the remaining training labels to calculate a p -value for each label.

Step 3: In the final step, the algorithm outputs the predictions based on the user's requirement. If the user wants a single prediction, the label with the biggest p -value is chosen as the predicted label. If the user prefers a prediction set, the algorithm will ask for a confidence level $(1 - \xi)$ from 0% to 100%, where ξ is called the significance level. Then, any training label with a p -value greater than ξ will be included in the prediction set Γ^ξ .

$$\Gamma^\xi(T_1, \dots, T_{M+1}) = \{L \mid p(L) > \xi\}. \tag{2}$$

The last objective of this section is to define the nonconformity function $A(T, T_i)$ to calculate the difference between a training example T_i and all other training examples, as mentioned above. It is worth reminding that CP produces valid predictions with any nonconformity function, although the prediction's efficiency may vary. The guidelines in Shafer and Vovk (2008) and Vovk et al. (2005) recommend that the nonconformity function should involve both the label and the object. For CP, the function should compute the difference between the nearest training example with the same label and the nearest training example with a different label (Shafer & Vovk, 2008; Vovk et al., 2005). However, with fingerprinting, the nearest training example may not necessarily be the optimal one due to the noisiness of the WiFi RSS and the signal multi-path problem. Therefore, this article employs W-KNN to consider a set of K training examples, where different values of K will be evaluated later on. In principle, the idea is to group these K training examples into one weighted averaged position, and follows the same guidelines as described above. The details of such nonconformity function will be explained below.

The concept of W-KNN has been described earlier in Section 2.5.1, and will not repeat here.

The first step in calculating the nonconformity function $A(T, T_i)$ for the training example T_i is to find K other training examples $\{T_1, \dots, T_K\}$ which have the smallest Euclidean distance in terms of the WiFi RSS to T_i . These examples must also have a different Cartesian label from T_i . In the second step, these K training examples are combined into one weighted average position $\vec{e} = (d_x^e, d_y^e)$ as follows, ϵ is a small constant to avoid division by zero.

$$\vec{e} = \frac{\sum_{j=1}^K \frac{1}{\text{Dist}(\vec{\text{RSS}}_j, \vec{\text{RSS}}_i) + \epsilon} \vec{L}_j}{\sum_{j=1}^K \frac{1}{\text{Dist}(\vec{\text{RSS}}_j, \vec{\text{RSS}}_i) + \epsilon}}. \quad (3)$$

The nonconformity measure α_i is the difference between the Cartesian label \vec{L}_i of T_i and the above weighted position \vec{e} .

$$\alpha_i = \|\vec{L}_i - \vec{e}\| = \sqrt{(d_x^i - d_x^e)^2 + (d_y^i - d_y^e)^2}. \quad (4)$$

The implementation of CP is summarized in Algorithm 1. The implementations in R and Java are also available to download.²

6. The fingerprinting test beds

This article uses three test beds to validate the algorithms. The first one (Royal Holloway) was manually collected by the authors in a standard office's environment using a smart phone. The second one (Cambridge) was collected automatically by a robot designed by the authors in a fairly ideal environment, supported by an independent tracking system for ground-truth. The last one (UJIIndoorLoc) is a public dataset which covers a huge indoor area spanning across three buildings in a very challenging environment. All three test beds are available for further research.³ The UJIIndoorLoc dataset has recently been used for the EvAAL 2015 fingerprinting competition which provides a relative benchmark for the results in this article. Each test bed has a large training database which includes both the signal measure (i.e. the WiFi RSS) and its label (i.e. the positioning co-ordinate); and a smaller test database which was collected randomly and separately to provide the test samples that may not be covered in the training set. Their detailed information are described below.

6.1. Royal Holloway test bed

This test bed was collected on the ground floor of the McCrea building at Royal Holloway, University of London. The floor plan of 45.4 by 32.6 m composes of 3 corridors and 27 offices. Most of the tracking space was in the corridors (see Figure 4). There were 9 WiFi APs directly inside the building to provide strong RSS (see Figure 5). There were also many other weaker APs from nearby buildings which brought the total number of WiFi APs in the training set to 131. Any training position can observe at least 28 APs.

Data: Training database $B = \{T_1, \dots, T_M\}$, significance level ξ , new sample T_{M+1} with known RSS

Result: Prediction region R

```

/* Calculate the Euclidean distance of two RSS vectors */
Function Dist( $P, Q$ )
    distance = 0;
    /* N is the total number of APs */
    for  $i = 1 \rightarrow N$  do
        |  $distance = distance + (AP_i^P - AP_i^Q)^2$ ;
    end
    return  $\sqrt{distance}$ ;

/* Calculate the weighted average position */
/* Kset contains K nearest neighbours to z */
Function Weighted_location ( $Kset, z$ )
    for  $i = 1 \rightarrow K$  do
        |  $weight1 = 1/(Dist(Kset.RSS_i, z.RSS) + \epsilon) * Kset.L_i$ ;
        |  $weight2 = 1/(Dist(Kset.RSS_i, z.RSS) + \epsilon)$ ;
    end
    return  $weight1/weight2$ ;

Function Conform( $B, z$ )
    /* Find K nearest neighbours to z, with different labels */
     $Kset = \{\}$ ;
    for  $i = 1 \rightarrow M$  do
        | if ( $T_i.L \neq z.L$ ) and  $min(Dist(T_i.RSS, z.RSS))$  then
        | |  $Kset = Kset + \{T_i\}$ ;
        | end
    end
     $e = Weighted_location(Kset, z)$ ;
    return  $\sqrt{(z.L - e)^2}$ ;

/* Y is the set of training positions */
for  $y \in Y$  do
     $T_{M+1} = (RSS_{M+1}, y)$ ;
     $B = B + \{T_{M+1}\}$ ;
    for  $T_j$  in  $B$  do
        |  $\alpha_j = Conform(B, T_j)$ 
    end
     $p(y) = \frac{\#\{j=1, \dots, M+1: \alpha_j \geq \alpha_{M+1}\}}{M+1}$ 
end
Prediction set  $R = \{y : p(y) > \xi\}$ ;
return  $R$ ;
    
```

Algorithm 1. Fingerprinting CP with W-KNN.

This dataset was generated in a standard manner, taking into considerations of the lessons learnt to produce a good training set. For instance, the granularity was 1 m to cover most positions in the building. Each training position recorded 200 readings to capture the full WiFi RSS variation, and also for the probabilistic methods to work well. The orientation of each training example also covered the four main cardinal directions (N/W/S/E). The collection time was short to avoid the temporal environmental changes. The mobile device used to collect the fingerprints was the Nexus 5. An app was developed to support the training process (see Figure 6). Out of the three test beds, this one has the highest number of measure per training location.



Figure 4. The Royal Holloway test environment. (a) South corridor, (b) southeast corridor and (c) east corridor.

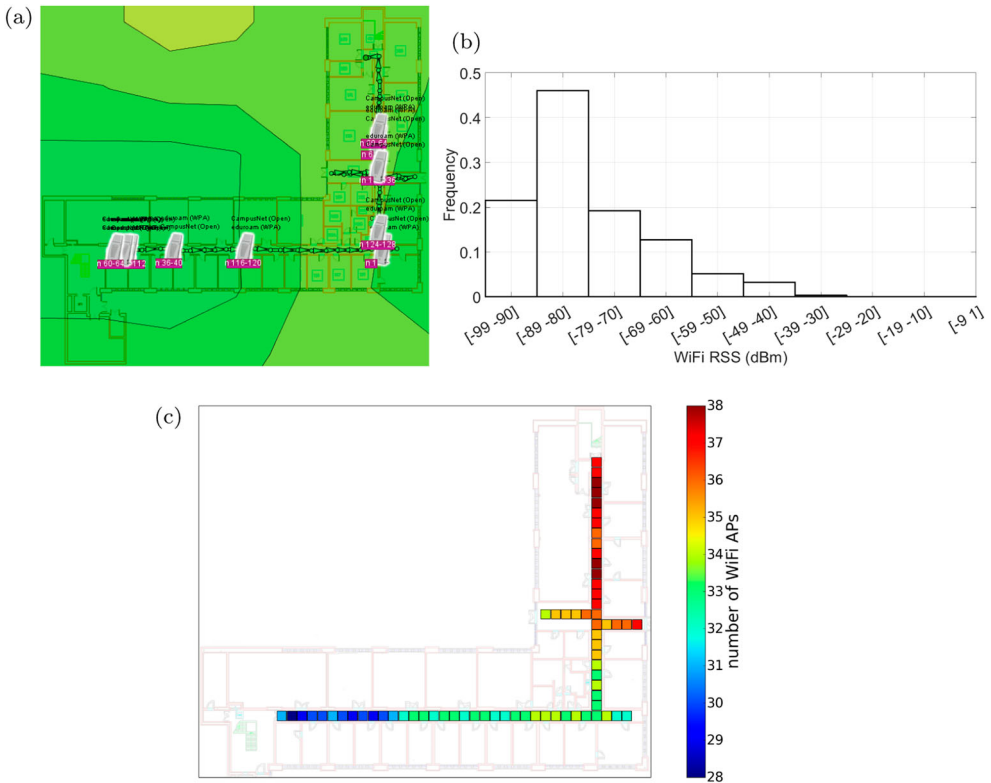


Figure 5. The coverage of the WiFi APs of the Royal Holloway test bed. (a) The rough position of the nine WiFi APs inside the building, generated by Ekahau, (b) the WiFi RSS distribution of the Royal Holloway test bed and (c) the number of WiFi APs covered by the training positions.

6.2. Cambridge test bed

This test bed was collected on the second floor in the North corridor of the Computer Lab at the University of Cambridge. The tracking space contains a long corridor of 45 by 1.7 m,

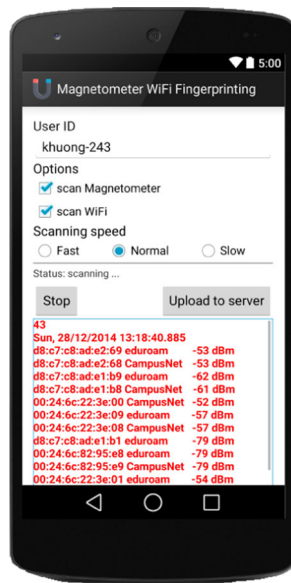


Figure 6. The Android app used to collect the fingerprints for the Royal Holloway test bed.

and a single room of about 29.3 m² (6.1 by 4.8 m) (see [Figure 7](#)). There were 5 WiFi APs inside this area to provide strong RSS, in which 4 APs were positioned at the two ends of the corridor, and 1 AP was in the middle of the corridor. There were also weaker APs from the surrounding areas. In total, there are 450 training positions along the corridor and 1500 training positions in one single room.

A robot designed by the author was used to collect the training fingerprints (see [Figure 8](#)). It carries a netbook (Sony P-115) on its back, and an Active Bat tag attached to its head. The netbook has a Java program which is responsible to co-ordinate the robot movements, and to record the WiFi RSS and the Cartesian label provided by the Active Bat system. More details about this robot can be found in [Nguyen \(2011\)](#).

The highlights of this test bed are the high precision of the reference labels provided by the Active Bat system (up to 3 cm error, 95% probability), and the fine-grained 10 cm resolution of the training space (30 cm resolution for the corridor) that was made possible only with a robot. The environment of this test bed was fairly ideal, with a wide and long corridor, and an empty room with no furniture. The training set was compiled over the weekend with no people around.

6.3. UJIIndoorLoc test bed

This dataset covers 3 buildings of 110,000 m² of the Universitat Jaume I in Spain, with a total of 13 floors (see [Figures 9 and 10](#)). Out of the three test beds used in this article, this is the only one covering multiple buildings and floors. It was used in the EvAAL fingerprinting competition in 2015.⁴

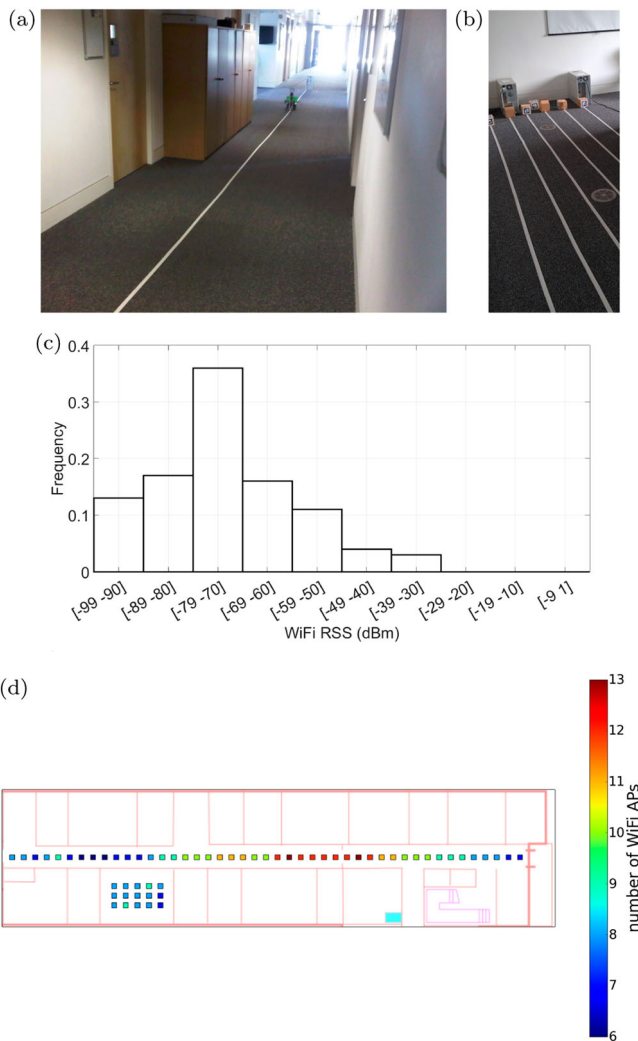


Figure 7. The Cambridge test bed. (a) The $45\text{ m} \times 1.7\text{ m}$ corridor, (b) the $6.1\text{ m} \times 4.8\text{ m}$ room, (c) the WiFi RSS distribution of the Cambridge test bed and (d) the number of WiFi APs covered by some training positions, at 1 m resolution.

The challenge of this dataset is not just the sheer amount of tracking space, but also the nature in which the dataset was generated. Firstly, there were 25 different mobile devices involved in the process. Secondly, 18 contributors used those devices to record the WiFi RSS without any pre-arrangement. Thirdly, the user taps on the touch screen to label the recorded WiFi RSS. Then, a central server converts this rough estimation into a numeric latitude and longitude label. With so many different contributors, their expectations may be different, therefore, the labels they provide may not be uniform. Lastly, the training time was long (20 days) and the two test sets were collected 3 and 18 months afterwards, which no longer reflect the original state of the indoor infrastructure. These conditions make this test bed highly challenging for any algorithm to estimate the user position.

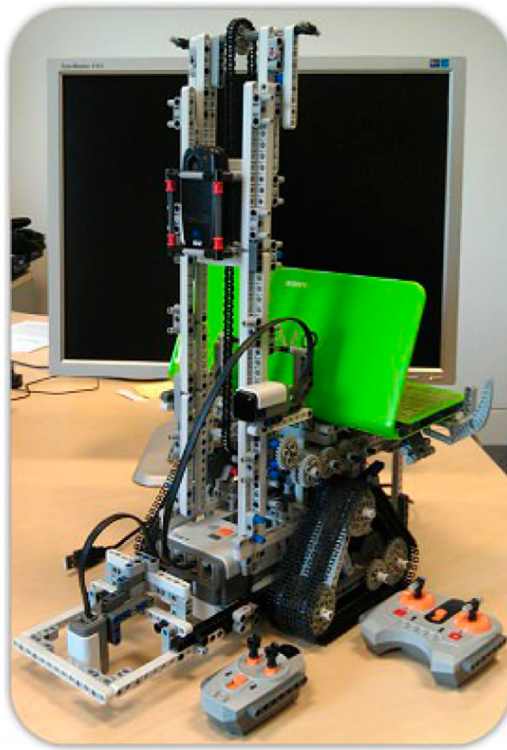


Figure 8. The LEGO robot used to collect the WiFi fingerprints for the Cambridge test bed.

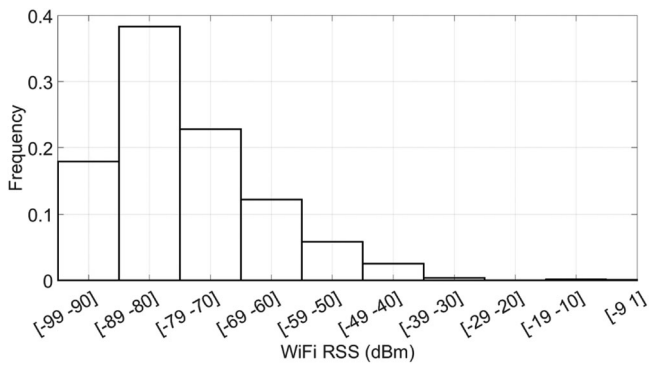


Figure 9. The WiFi RSS distribution of the UJIIndoorLoc test bed.

6.4. Summary of the three test beds

In summary, the three fingerprinting datasets used in this article were selected to represent three different indoor environments, which include a standard one (Royal Holloway), an ideal one (Cambridge) and a challenging one (UJIIndoorLoc). The Royal Holloway dataset has the highest number of measure per location, while the Cambridge

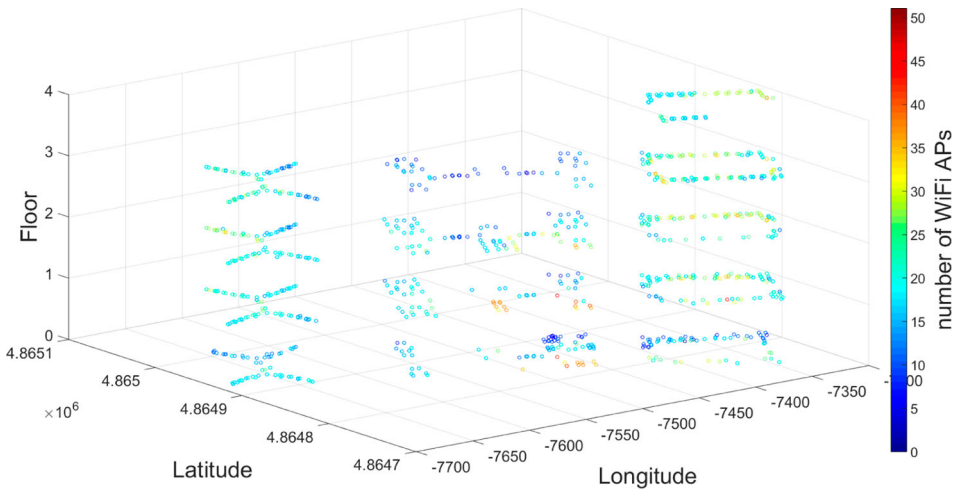


Figure 10. The coverage of the WiFi APs of the UJIIndoorLoc test bed.

Table 4. Summary of the three fingerprinting test beds used in this article.

	Royal Holloway	Cambridge	UJIIndoorLoc
Abbreviation	RH	Cam	UJI
Training examples	13,600	78,000	19,937
Training area	1480 m ²	540 m ²	110,000 m ²
Surveyed space	3 corridors	1 corridor, 1 room	3 buildings
Training time	1 day	2 days	20 days
Granularity	1 m	10 cm for room 30 cm for corridor	Up to 2 m
Test samples	150	100	1,111
Last test sample	Same day	1 day later	3 months later
Measuring time	Working hours	Weekend	Working hours
Measures per location	200	40	Up to 30
Distinct positions	68	1950	933
Orientations per location	4	4	Unknown
Label type	Cartesian (metre scale)	Cartesian (metre scale)	Longitude and latitude
Label generator	Manually by the surveyor	Automatically by Active Bat system	Manually by the surveyor
Total building(s)	1	1	3
Total floor(s)	1	1	13
WiFi APs	131	43	529
Device(s) used	1 phone	1 netbook	25 phones
Surveyor(s)	1 person	1 robot	18 people

one has the highest training resolution. Out of the three test beds, the UJIIndoorLoc dataset covers the widest indoor space. The majority of the WiFi RSS in all three datasets were around $[-80 -70]$ (dBm). Table 4 summarizes the characteristics of the three test beds.

7. Evaluation of performance

This section compares the performance accuracy of the proposed algorithm to other popular machine learning algorithms for fingerprinting in the literature such as Naïve Bayes and W-KNN. The advantage of using the confidence measure will be evaluated to

assess whether it produces valid predictions, and if there is any improvement on the positioning result.

7.1. The validity of CP evaluation

The so-called error rate of CP is the percentage indicating how often the algorithm does not produce a prediction set that contains the true location. The algorithm is considered valid under a confidence level $(1 - \xi)$, if the error rate does not exceed ξ . The formulae to compute this error rate for a given confidence level is defined below, where N is the total number of tests, L_i is the Cartesian label and R_i is the prediction set or region of the test sample i th $(1 \leq i \leq N)$.

$$\text{Error_rate} = \frac{\#|i = 1, \dots, N : L_i \notin R_i|}{N} \tag{5}$$

To study the validity of the predictions, the first experiment performs 10-fold cross-validation on each training database of the three test beds. This is the special case of leave-one-out validation, where each training example is left out and is used as the test sample. However, due to the high number of training examples in all three datasets, it is preferred to use 10-fold cross-validation. The training examples are randomly divided into 10 roughly equal portions, for which 9 of them will be used as the training set and the single remaining portion will be used for testing. In particular, each fold has about 1360 examples for the RH test bed, 7800 examples for the Cam test bed and 1990 examples for the UJI test bed. This process is repeated nine times for the other portions, so that every training example has a chance to be considered as test sample. The error across all 10 trials will be averaged to obtain the error rate.

Figure 11 demonstrates that CP is valid on all three datasets. The error rates were around the specified confidence level in all cross-validations (subject to statistical fluctuation). When the significance level is zero, which is equivalent to 100% confidence level, there is no error since the predictions are the whole training set. With a significance level of 1, which is equivalent to 0% confidence level, the error rate is 100% since the prediction set is empty. The test did not go below 60% confidence level, because many test samples started to return empty prediction sets at this level. This result also indicates that this is the threshold for the fingerprinting datasets used in this article.

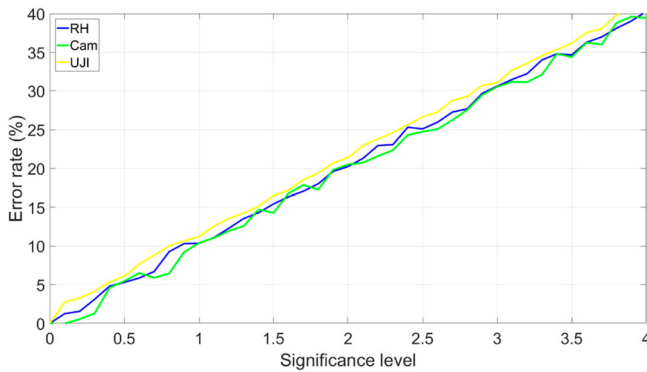


Figure 11. The validity of CP under three test beds.

This experiment has demonstrated the validity of the predictions provided by CP, under different significance (or confidence) levels. However, it is more interesting to the fingerprinting researchers that the prediction set (or the prediction region) is small. Thus, the next experiment evaluates the narrowness of the prediction sets and regions, with respect to the positioning accuracy and the confidence level. The same training sets from the above 10-fold cross-validation were used. For each confidence level, the positioning prediction accuracy for all test samples were calculated and then averaged.

7.2. The narrowness of CP evaluation

With this approach, there are two parameters to control, which are the K neighbours and the confidence level. The value of K decides how many training examples to be considered to calculate an average position. A big K includes the examples that are too far away in terms of the Euclidean distance. A small K (such as 1-nearest neighbour) may not include the correct prediction, because the training database is noisy. The value of K for optimal positioning prediction was empirically shown to be around 60–80 for the RH test bed, 30–40 for the Cam test bed, and 10–20 for the UJI test bed (see Figure 12). Depending on the chosen confidence level, the value of K slightly varies. For the ease of comparison, a fixed K of 70, 35 and 15 were selected for the RH, Cam and UJI test bed, respectively.

Using the optimal values of K found above, the next experiment evaluates the performance of CP under different confidence levels to understand the size of the prediction set and the performance accuracy. Since CP produces a set of predictions, there are two methods to obtain a single prediction to compare with the true position. The first method is averaging the whole prediction set to a single position. The second method

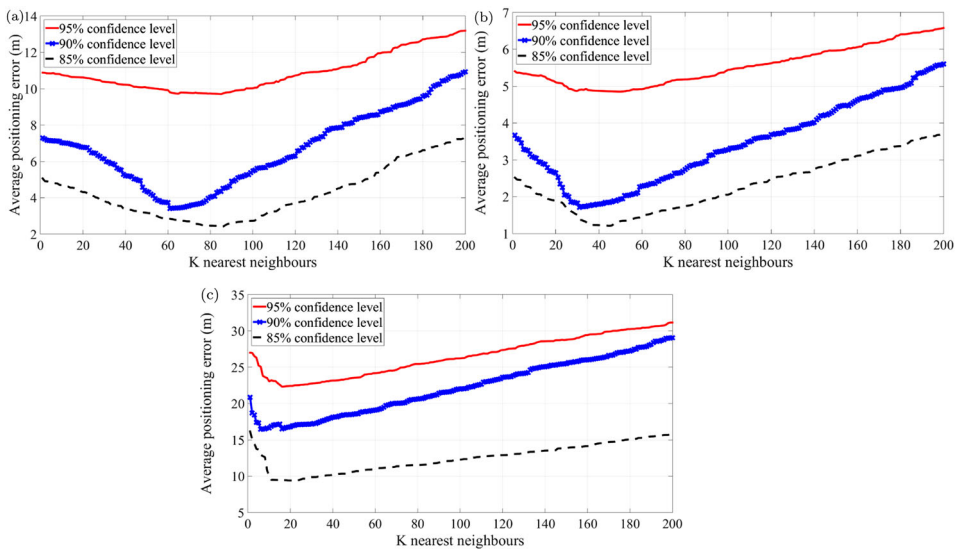


Figure 12. Performance accuracy of CP under different K nearest neighbours. (a) RH test bed, (b) Cam test bed and (c) UJI test bed.

is simply using the training position with the largest p -value. Their performances will be evaluated below.

Tables 5–7 demonstrate that the size of the prediction set strictly decreases as the confidence level decreases for all three test beds, as expected. At 100% confidence, the full training set of all nine folds was returned. At 60% confidence levels for all three test beds, several test samples started to return empty prediction sets, which indicated that this seems to be the threshold. Overall, the recommended confidence levels for CP are 75% for the RH test bed, 80% for the Cam test bed and 65% for the UJI test bed, which achieve 2.42, 0.7 and 9.3 m error, respectively, by averaging the prediction set.

When the training position with the largest p -value was used as the predicted position, the positioning error was smaller than that when the whole prediction set was used, for the confidence levels within 85% and 100%. However, as the confidence level decreases from 85% downwards, the positioning error got larger with the largest p -value prediction, for all three test beds (see Figure 13). This is because when the confidence level is high, the

Table 5. Performance of CP for the RH test bed.

Confidence level (%)	Prediction error (m)	Predicted positions	Error rate (%)
100	18.2	62	0
95	6.7	46	4.8
90	3.8	31	10.3
85	2.44	19	11.2
80	2.5	14	19.6
75	2.42	9	25.3
70	2.58	9	29.7
65	2.51	8	34.8
60	2.53	4	39.6

Table 6. Performance of CP for the Cam test bed.

Confidence level (%)	Prediction error (m)	Predicted positions	Error rate (%)
100	7.6	1755	0
95	4.9	1312	4.6
90	1.8	1106	9.2
85	1.2	879	14.7
80	0.7	513	19.8
75	0.75	307	24.3
70	0.82	156	29.5
65	0.88	79	34.8
60	0.83	38	39.6

Table 7. Performance of CP for the UJI test bed.

Confidence level (%)	Prediction error (m)	Predicted positions	Error rate (%)
100	37.4	837	0
95	22.3	592	5.6
90	16.5	522	10.8
85	9.8	469	15.8
80	10.2	317	21.2
75	10.1	143	26.7
70	9.7	82	31.3
65	9.3	29	36.4
60	10.6	16	41.5

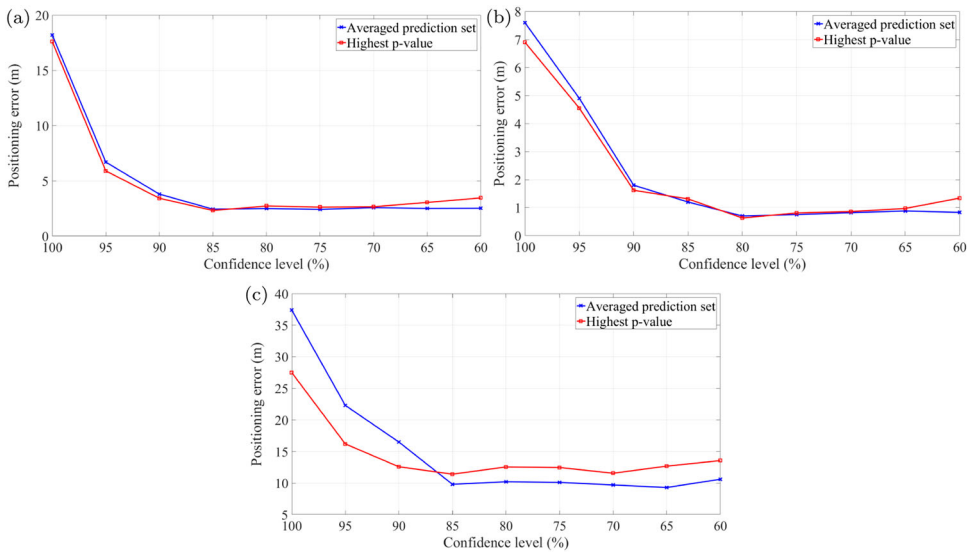


Figure 13. Comparison of the performance accuracy of the single prediction with CP, under the averaged prediction set and largest p -value prediction. (a) RH test bed, (b) Cam test bed and (c) UJI test bed.

prediction set contains many predictions including those that are not close to the true position. Hence, the overall positioning error when using the whole training set was higher. However, the prediction with the largest p -value is not always the correct prediction, due to the noisiness of the training database. Thus, its positioning error slightly suffers when the size of the prediction set decreases.

7.3. Overall prediction accuracy evaluation

To understand how the proposed CP algorithms perform in the real world, they will be verified with the test sets which have been independently collected and may contain samples that were not seen in the training sets. This experiment uses the most optimal parameters found from the cross-validation experiments in the earlier sections. In doing so, it compares the performance of confidence machine against other well-known traditional approaches in fingerprinting including Naïve Bayes and W-KNN. The test set of the UJI test bed was the public version collected three months after the training set. There is another competition version that was generated 18 months later, which will be evaluated in the next section.

Figure 14 uses the cumulative distribution function to demonstrate the superiority of the proposed confidence learning algorithms, which outperform the traditional fingerprinting algorithms by 15–20% on average in all three independent test sets. In particular, compared to W-KNN, CP reduced the positioning error from 4.2 m error to under 3.6 m error, 95% probability, for the RH test set. For the Cam test bed, the positioning error was reduced from 3.2 to 2.2 m, 95% probability. For the UJI test bed, the improvement was more significant, from 17 to under 7 m, 95% probability. W-KNN performed better than Naïve Bayes in all three test sets, which suffered heavily with so few measures per location in the UJI training set.

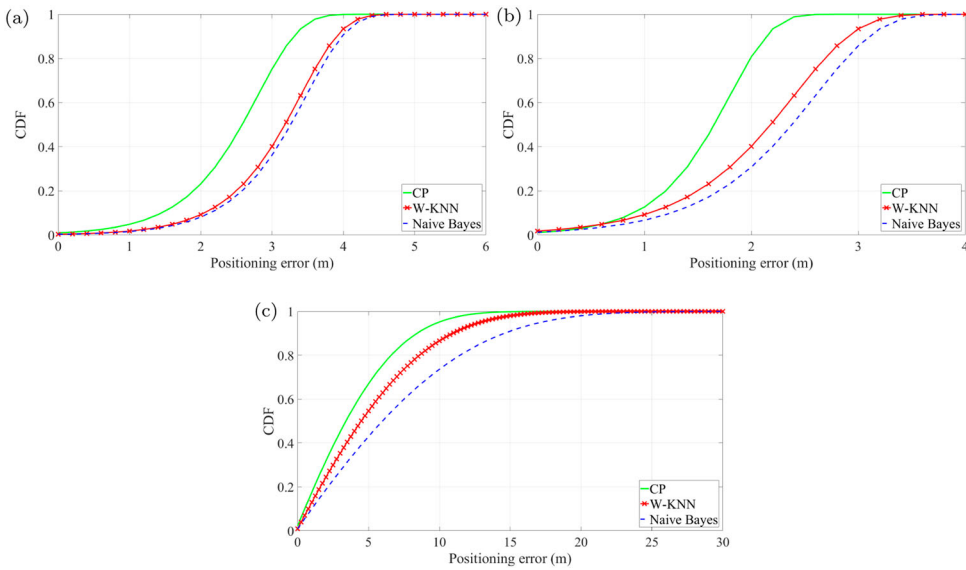


Figure 14. Performance accuracy of confidence machine. (a) RH test bed, (b) Cam test bed and (c) UJI test bed.

7.4. UJI's floors and buildings hit rate evaluation

Out of the three test beds, the UJI test bed has multiple floors and buildings. Although the proposed algorithms in this article were designed with two-dimensional space in mind, it would be interesting to assess how well they identify the user position in three-dimensional. Since CP returns a set of predictions containing the precise training positions, a simple majority will decide which floor and building for the whole prediction set.

Out of 1111 test samples, a very high 99.4% successful hit rate for the building prediction and 78.2% hit rate for floor prediction were achieved. Out of 21.8% test samples that were estimated on the wrong floors, most of them were only 1 floor below or above the correct one (see Figure 15).

7.5. EvAAL competition test set evaluation

With the UJI test bed, the training set is un-changed. However, there are two versions of the test set. The first one is publicly available to anyone to self-evaluate their systems, which has been used in the previous sections. The second one is un-labelled so that it can be used for the EvAAL 2015 competition. This test set may be obtained by contacting the test bed's author directly. Its performance is presented here.

In the EvAAL competition, each entrant was allowed five submissions, and the most accurate one was chosen as the final result. The parameters for CP were $K = \{10, 15\}$ which are around the optimal parameters suggested by the earlier experiments. It is worth noting that although the proposed algorithms were not officially entered in the competition, the submissions were evaluated using the same criteria as with other competitors. In particular, the positioning error was calculated as follows, where R is the true positioning label, and E is the estimated positioning label. For every wrongly

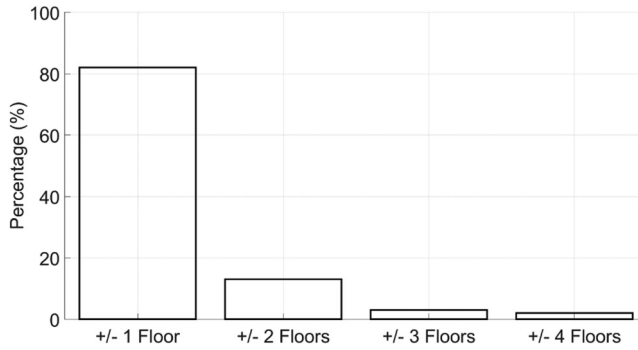


Figure 15. Performance of CP with the UJI multi-floor test set.

predicted building, a 50 m penalty is added into the result, and for every wrongly predicted floor, a 4 m penalty is added.

$$\text{Error}(R, E) = \text{Euclidean_Distance}(R, E) + (50 * \text{buildingfail}) + (4 * \text{floorfail}). \quad (6)$$

The final results put CP as second with 9.1 m error, 75% probability, and CP as third with 10.4 m error, 75% probability, out of the five competitors (see Figure 16). The detailed performance of CP under the competition test set is also available.⁵

The information of other competitors can be found on the competition website.⁶ It is worth noting that some of them came from the industry and did not reveal much of their underlying algorithms. The RTLS system that came first in this competition also used W-KNN. However, it applies several filtering methods to the training database, and then divides it into smaller training sets. In other words, it was designed to tackle this dataset specifically. The proposed CP algorithm in this article was tested with the full complete training database. In addition, the proposed confidence learning approach in this article offers more information (i.e. the confidence level) which was not utilized at all since the competitors were ranked solely on the positioning error. More importantly, the proposed algorithms can produce a prediction set or a prediction region, which was forced to reduce into a single prediction for evaluation.

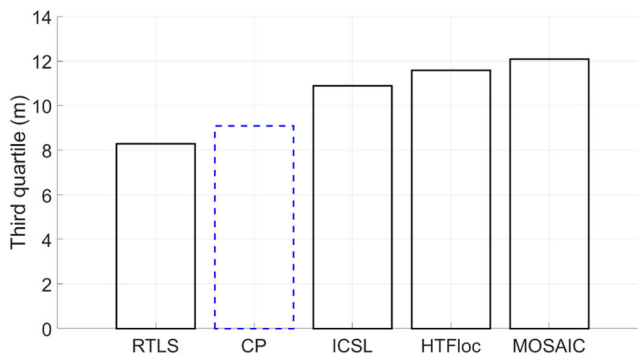


Figure 16. Performance of CP with the EvAAL competition test set.

8. Conclusion and further work

This article has proposed a novel confidence learning approach to estimate the user position indoors with the WiFi signal strength. It introduces a confidence measure, which is not only useful in reflecting the uncertainty of the positioning predictions, but is also capable of adjusting the size of the prediction set accordingly. The positioning accuracy has been empirically shown to be around 2.4 m, 75% probability under a normal test bed, around 70 cm, 75% probability under an ideal test bed, and around 8.8 m, 75% probability under a challenging test bed. These results outperformed non-confidence machine learning algorithms tested on the same test beds by up to 20% more accurately. The prediction sets produced by the proposed algorithms have also been demonstrated to be valid. The proposed confidence learning algorithms were compared to other machine learning algorithms in the EvAAL 2015 competition under the same testing criteria. The result ranked the proposed algorithm second out of the five competitors.

Although the approaches presented in this article do not require the site map of the building at all, such map can be combined to provide extra ground-truth information to remove the violated predictions, where the user walks through the walls. This has been a popular idea with robot-based Simultaneous Localization And Mapping (Durrant-Whyte & Bailey, 2006; Faragher & Harle, 2013; Howard, 2006; Wolf & Sukhatme, 2005). The building map is also often available for most buildings.

Furthermore, a highly adaptable system should be able to improve itself over time. With reinforcement learning, the system changes its model based on the rewards received from time to time to maximize its performance (Schölkopf & Smola, 2002; Yang, 2010). The challenge to apply reinforcement learning for fingerprinting is that there is often no such rewards telling the positioning system of how well it performs at any stage. A possible means to introduce such rewards is by asking the users directly. At the end of the journey when the user has reached his destination, he is asked to rank the system's performance accuracy. Based on this feedback, the system's action is to prioritize the training examples that lead to a good feedback, and avoids those that result in bad feedbacks.

Notes

1. <http://research.microsoft.com/en-us/events/ipsn2014indoorlocalizationcompetition>
2. <http://www.cs.rhul.ac.uk/~wruf265/CP/>
3. <http://www.cs.rhul.ac.uk/~wruf265/datasets/>
4. <http://evaal.aaloa.org/current-competition/track-3>
5. <http://www.cs.rhul.ac.uk/~wruf265/CP/>
6. <http://evaal.aaloa.org/current-competition/track-3>

Disclosure statement

No potential conflict of interest was reported by the author.

Funding

This research is partly funded by the Computer Science Department of Royal Holloway, University of London.

Notes on contributor

Khuong An Nguyen holds a Ph.D. in Computer Science at Royal Holloway, University of London, UK. Before that, he completed an M.Phil. in Advanced Computer Science at the University of Cambridge, where he implemented a Bluetooth-based tracking system in the Computer Lab. His research interests include indoor positioning and machine learning.

ORCID

Khuong An Nguyen  <http://orcid.org/0000-0001-6198-9295>

References

- Bahl, P., & Padmanabhan, V. N. (2000). *Radar: An in-building rf-based user location and tracking system*. Proceedings of the IEEE INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies, Tel Aviv, Israel (Vol. 2, pp. 775–784).
- Bellotti, T., Luo, Z., Gammernan, A., Van Delft, F. W., & Saha, V. (2005). Qualified predictions for microarray and proteomics pattern diagnostics with confidence machines. *International Journal of Neural Systems*, 15(4), 247–258.
- Bishop, C. M. (2006). *Pattern recognition and machine learning*. New York: Springer.
- Brabham, D. C. (2008). Crowdsourcing as a model for problem solving an introduction and cases. *Convergence: The International Journal of Research into New Media Technologies*, 14(1), 75–90.
- Chaudhry, S. (2013). *Indoor location estimation using an nfc-based crowdsourcing approach for bootstrapping* (Master's Thesis). University of Oulu, Department of Computer Science and Engineering.
- Chen, L., Pei, L., Kuusniemi, H., Chen, Y., Kröger, T., & Chen, R. (2013). Bayesian fusion for indoor positioning using bluetooth fingerprints. *Wireless Personal Communications*, 70(4), 1735–1745.
- Dashevskiy, M., & Luo, Z. (2009). Reliable probabilistic classification of internet traffic. *International Journal of Information Acquisition*, 6(2), 133–146.
- Dawes, B., & Chin, K.-W. (2011). A comparison of deterministic and probabilistic methods for indoor localization. *Journal of Systems and Software*, 84(3), 442–451.
- Durrant-Whyte, H., & Bailey, T. (2006). Simultaneous localization and mapping: Part i. *IEEE Robotics & Automation Magazine*, 13(2), 99–110.
- Faragher, R., & Harle, R. (2013). *Smartslam – An efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing*. IEEE ION GNSS, Nashville, Tennessee (Vol. 13, pp. 1–14).
- Haverinen, J., & Kemppainen, A. (2009). Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10), 1028–1035.
- Honkavirta, V., Perälä, T., Ali-Löytty, S., & Piché, R. (2009). *A comparative survey of wlan location fingerprinting methods*. 6th workshop on positioning, navigation and communication, 2009 (WPNC 2009), Hannover, Germany (pp. 243–251).
- Howard, A. (2006). Multi-robot simultaneous localization and mapping using particle filters. *The International Journal of Robotics Research*, 25(12), 1243–1256.
- Huang, C.-N., & Chan, C.-T. (2011). Zigbee-based indoor location system by *k*-nearest neighbor algorithm with weighted rssi. *Procedia Computer Science*, 5, 58–65.
- Ibrahim, M., & Youssef, M. (2013). *Enabling wide deployment of gsm localization over heterogeneous phones*. 2013 IEEE international conference on communications (ICC), Budapest, Hungary (pp. 6396–6400).
- Jung, J., Lee, S., Kim, H., Park, B., & Myung, H. (2013). *Mobile robot relocation using ambient magnetic fields and radio sources*. 2013 13th international conference on control, automation and systems (ICCAS), Gwangju, Korea (pp. 1766–1768).
- Lambrou, A., Papadopoulos, H., & Gammernan, A. (2009). *Evolutionary conformal prediction for breast cancer diagnosis*. 2009 9th international conference on information technology and applications in biomedicine, Larnaca, Cyprus (pp. 1–4).

- Lambrou, A., Papadopoulos, H., Kyriacou, E., Pattichis, C. S., Pattichis, M. S., Gammerman, A., & Nicolaides, A. (2010). *Assessment of stroke risk based on morphological ultrasound image analysis with conformal prediction*. IFIP international conference on artificial intelligence applications and innovations, Larnaca, Cyprus (pp. 146–153).
- Ledlie, J., Park, J.-G., Curtis, D., Cavalcante, A., Camara, L., Costa, A., & Vieira, R. (2012). Molé: a scalable, user-generated WiFi positioning engine. *Journal of Location Based Services*, 6(2), 55–80.
- Lee, M., & Han, D. (2012). *Qrloc: User-involved calibration using quick response codes for wi-fi based indoor localization*. 2012 7th international conference on computing and convergence technology (IC CCT), Seoul, Korea (pp. 1460–1465).
- Lin, T.-N., & Lin, P.-C. (2005). *Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks*. 2005 international conference on wireless networks, communications and mobile computing, Maui, USA (Vol. 2, pp. 1569–1574).
- Nguyen, K. A. (2011). *Robot-based evaluation of bluetooth fingerprinting* (Unpublished master's thesis). Computer Lab, University of Cambridge.
- Schölkopf, B., & Smola, A. J. (2002). *Learning with kernels: Support vector machines, regularization, optimization and beyond*. Cambridge, MA: MIT Press.
- Sen, S., Lee, J., Kim, K.-H., & Congdon, P. (2013). *Avoiding multipath to revive in building wifi localization*. Proceeding of the 11th annual international conference on mobile systems, applications, and services, Taipei, Taiwan (pp. 249–262).
- Shafer, G., & Vovk, V. (2008). A tutorial on conformal prediction. *The Journal of Machine Learning Research*, 9, 371–421.
- Subhan, F., Hasbullah, H., Rozyyev, A., & Bakhsh, S. T. (2011). *Indoor positioning in bluetooth networks using fingerprinting and lateration approach*. 2011 international conference on information science and applications (ICISA), Jeju Island, Korea (pp. 1–9).
- Vovk, V., Gammerman, A., & Shafer, G. (2005). *Algorithmic learning in a random world*. New York: Springer.
- Wang, B., Zhou, S., Yang, L. T., & Mo, Y. (2015). Indoor positioning via subarea fingerprinting and surface fitting with received signal strength. *Pervasive and Mobile Computing*, 23, 43–58.
- Want, R., Hopper, A., Falcao, V., & Gibbons, J. (1992). The active badge location system. *ACM Transactions on Information Systems*, 10(1), 91–102.
- Weber, M., Birkel, U., Collmann, R., & Engelbrecht, J. (2010). *Comparison of various methods for indoor RF fingerprinting using leaky feeder cable*. 2010 7th workshop on positioning navigation and communication (WPNC), Dresden, Germany (pp. 291–298).
- Wolf, D. F., & Sukhatme, G. S. (2005). Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, 19(1), 53–65.
- Yang, Z. R. (2010). *Machine learning approaches to bioinformatics (Vol. 4)*. Singapore: World Scientific.
- Youssef, M., & Agrawala, A. (2005). *The horus wlan location determination system*. Proceedings of the 3rd international conference on mobile systems, applications, and services, Seattle, USA (pp. 205–218).