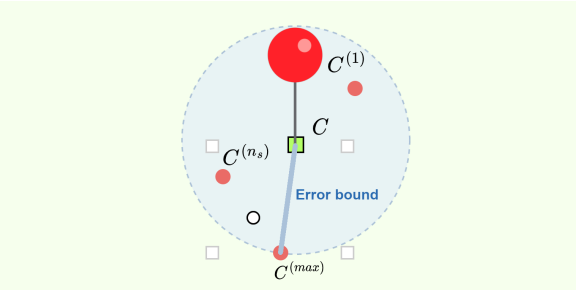
Accuracy of a single position estimate for kNN-based fingerprinting indoor positioning applying error propagation theory

Antoni Pérez-Navarro, Raúl Montoliu, Emilio Sansano-Sansano, Marina Martínez-Garcia, Ruben Femenia and Joaquín Torres-Sospedra

***Abstract*— Indoor Positioning Systems usually consider the average positioning error over a set of evaluation samples, or a quartile of that value, as the global error. However, they do not provide a metric for the uncertainty for each individual position estimation. In this paper, we apply the error propagation theory to the kNN algorithm in Wi-Fi fingerprint-based indoor positioning. Our proposed method does not only retrieve the position estimate but also describes how the uncertainties of the RSSI measurements propagate through the calculations. We have validated our proposed method with two open-access datasets.**

***Index Terms*— Indoor positioning, Fingerprinting methods, Error propagation.**



I. Introduction

LOCATION-based systems (LBS) offer endless possibilities including passenger guidance at airports, the orientation of visually impaired people and, even, the support of event management. While LBS are widely powered by Global Navigation Satellite Systems (GNSSs) outdoors, there is no standard for positioning indoors and in GNSS-denied spaces.

Manuscript received March 1, 2023; revised June 1, 2023; accepted September 1, 2023. Date of publication September 1, 2023; date of current version September 1, 2023. This work was supported in part by projects Grant PID2021-1226420B-C44 funded AEI/10.13039/501100011033/FEDER,UE, Grant PID2021-1226420B-C42 funded AEI/10.13039/501100011033/FEDER,UE, Grant TED2021-199866B–I00 funded AEI/10.13039/501100011033/ EuropeanUnion-NextGenerationEU/PRTR , UJI-A2022-12, CYTED Network "GeoLibero" and H2020-MSCA-IF 101023072. The associate editor coordinating the review of this article and approving it for publication was Prof. Name Surname (Corresponding authors: Antoni Pérez-Navarro, Raúl Montoliu and Joaquín Torres-Sospedra)

Antoni Pérez-Navarro is with the Faculty of Computer Sciences, Multimedia and Telecommunication, Universitat Oberta de Catalunya, Barcelona, Spain (e-mail: aperezn@uoc.edu).

Raúl Montoliu and Ruben Femenia are with the Institute of New Imaging Technologies, Universitat Jaume I, Castellón de la Plana 12003, Spain (e-mail: montoliu@uji.es).

Emilio Sansano-Sansano is with the Institute of New Imaging Technologies and the Department of Industrial Engineering Systems and Design, Universitat Jaume I, Castelló de la Plana 12003, Spain (e-mail: esansano@uji.es).

Marina Martínez-García is with the Mathematics Department, Universitat Jaume I, Castellón de la Plana 12003, Spain (e-mail: martigar@uji.es).

Joaquín Torres-Sospedra is with the Centro ALGORITMI, Universidade do Minho, 4800-058 Guimarães, Portugal (e-mail: info@jtorr.es).

Digital Object Identifier 10.1109/JSEN.2023.XXXXXXXX

GNSSs are able to provide navigation, positioning and timing services. In addition, these systems also offer what is known as Dynamic Accuracy Estimation (DAE), i.e. the system provides the accuracy of each provided position. This dynamic accuracy is usually represented as a circle around the estimated position, which corresponds to what is called the *error bound*. This error bound is usually the 68.26^{th} percentile confidence level, meaning that there is approximately a 68.26% chance that the true location of the device is within the DAE, i.e. inside the circle limits.

The error bound is important because it is a metric that represents how reliable is the provided position. The smaller the displayed circle, the more reliable the estimated position is. This is not only important in navigation applications, but also in advanced positioning systems based on sensor fusion where an unreliable position estimation may degrade the final output. However, what is already implemented in GNSS is lacking in indoor positioning systems, especially in those based on signals of opportunity like Wi-Fi fingerprinting.

Generally, the accuracy in indoor positioning systems is given as the average error over a set of test samples where their real positions are known. This accuracy is generally used to generate the same error bound for a group of estimates or, even, for all of them. This means that the error is calculated using the ground truth location of the position estimate and it is not updated anymore, i.e., the displayed circle will be the same size for all the position estimates. Thus, this approach is not considering the signal fluctuations over time or the particularities of the environment and indoor positioning system in a particular sub-region.

Although there have been several attempts to obtain different error bounds in different points ([1]–[3]), as far as we know, there is no mechanism to obtain *a priori* the error bound of a new position estimate. Knowing the error bound of a single position estimate is important in order to know the reliability of the position we are getting, i.e. what is the potential error for that estimated position. The three-sigma rule of thumb [4] could also be applied to provide an uncertainty of each provided position estimate. However, there are three key differences that prevent us from using this rule as the main uncertainty source. First, it requires calculating the standard deviation of the positioning error over a set of position estimates. This can only be calculated a posteriori, once we have all the positioning estimations on the test set, or over an independent validation set. Second, the same uncertainty, the three-sigma of the positioning errors, is assumed for any position estimate. In contrast to the conventional three-sigma rule of thumb, the uncertainty metric we propose is based on the mean and standard deviation of the measurements themselves (inputs), not on the positioning errors (outputs) and, in addition, each operational fingerprint will have attached its own uncertainty. i.e., the uncertainty will be somehow correlated with the variability of the RSS values in a single point. Third, it would give a higher error than the one calculated with the error propagation theory (EPT), and the zone could be too large to be useful.

An important factor that can contribute to errors in Wi-Fi fingerprinting is the fluctuation of the received signal strength (RSS) [5]. As a result of these fluctuations, there can be uncertainty in the experimentally measured variables, which then propagate through the calculations used to determine the final position. This ultimately leads to inaccuracies in the final result. The error propagation theory (EPT) [6] takes into account how these uncertainties propagate through the calculated magnitudes.

This study employs EPT to account for the influence of experimental measurement uncertainties on position calculations in Wi-Fi fingerprinting. This enables us to determine an error bound for each position estimate. Specifically, we concentrate on applying EPT to compute the error propagation in Wi-Fi fingerprinting using the kNN method as a position estimator.

Using EPT has two main advantages. On the one hand, the error is calculated for every empirical measure, therefore, we are able to give a specific error bound for every delivered position estimate. On the other hand, the formulae used in our study are obtained through analytical means, enabling us to identify the measures that significantly contribute to errors and are crucial for improving location accuracy. This mechanism allows knowing the accuracy of a single point.

The main contributions of this work are summarised as follows:

- We propose an indoor positioning system that not only provides the position estimates but also their associated area of uncertainty;
- We perform an empirical assessment of the proposed method using two real datasets.

II. Related Work

The calculation of error bound is developed in several ways in GNSS among which we can find dual-frequency GNSS equipment and augmentations systems like WAAS or EGNOS.

In the case of indoor systems, it is not possible to use the same algorithms to calculate the error bound. Therefore, we need to look for different mechanisms. To this end, it is important to understand how the process works in our particular use case, which is based on Wi-Fi fingerprinting using the kNN algorithm to estimate the positions.

There have been several works that deal with the error in indoor positioning. The ISO 18305 Standard [7], [8] provides several performance metrics that bring general accuracy of the positioning system, including mean positioning error, variance/covariance of the error, root mean square error and the 95$^{\text{th}}$ percentile of the error. Those metrics are computed over a series of independent evaluation points, bringing global accuracy. The standard also depicts how to provide accuracy for a given location, but it requires the receiver to be stationary to collect several samples and then apply one of the above-mentioned performance metrics. A popular alternative is the one used in the Indoor Positioning and Indoor Navigation (IPIN) competition ([9]–[11]). In that method, the accuracy corresponds to the third quartile (75$^{\text{th}}$ percentile) of an error metric that combines the horizontal positioning error and a penalty for wrong floor detection and wrong building detection. Recent works [12]–[16] do not rely on just one metric to report the results, providing different metrics (mean, percentile values, RMSE, among others). The main metrics in those works are backed many times with the CDF (Cumulative Distribution Function) plot of the individual errors. This multiple-metric evaluation is performed not only to report the positioning error but also to compare the proposed method to baselines or state-of-the-art methods. These kinds of measures of the error give a general value of the accuracy of the system but do not allow to know the accuracy of a single point. Also, the CDF plot does not relate the individual errors to locations. While the former metrics are calculated *a posteriori*, the error bound that we proposed is calculated on-line for each position estimate. Anyway, location-based metrics are relevant only for pure RF-based technologies, as positioning systems involving inertial measurements may end up in different metrics depending on the performed trajectory.

Marcus et al. [1] proposed a system to evaluate the accuracy of a fingerprinting system, SMARTPOS, based on dynamically setting the number of nearest neighbours in a kNN algorithm. The authors showed that errors follow a Gaussian distribution, which drove to four proposals of the error estimation: in the first one, for k greater than 2, they take the average geographic distance between the nearest neighbour to the 2$^{\text{nd}}$ until the k-th nearest fingerprint as the error. In the second proposal they take as the error the maximum geographic distance among any selected neighbour ($i > 1$) to the nearest neighbour. The third option is to estimate the distance between any two fingerprints in the sequence of selected nearest neighbours to obtain the error. The last option does not only uses the positions of the k nearest neighbours but also their corresponding weight (based

on distance in the RSSI space) and the final position estimate. The authors used the measures in the signal space to discover the fingerprints, but they also used their counterparts in the geographical space to determine the position.

Zou et al. [2] used as the error the maximum distance between reference points chosen in the kNN algorithm. This is similar to the second proposal used by Marcus et al. [1]. Marcus et al. and Zou proposals for the measurement of the accuracy are based on the structure of the fingerprints and, therefore, do not take into account differences among different regions in the target zone.

Lemic et al. [3] took into account the heterogeneity of the environment to calculate the error. They used a regression model that is trained with the information of the measured error obtained. i.e., they compare the RSS value at the real point with the RSS value at the estimated point. For positioning, they used several algorithms, among which kNN is found to be the best-performing one. In 2020, they improved the method by including Artificial Neural Networks [17]. On the one hand, Lemic et al. proposals were tested in an outdoor scenario but we consider that the mechanism used can be relevant also to indoor environments. However, a large amount of data needs to be collected for using the proposed algorithm, which may be considered an important drawback. On the other hand, Lemic et al. proposals, although looking for the anisotropy of the environment, did not take into account the anisotropy of RSS measures in time, i.e. the location error is defined and trained by data built during the offline phase. Nevertheless, in Lemic's proposals, the estimated position is included in the process, although it is only to improve the error location.

Recently, Anagnastopoulos et al. [18] reviewed several mechanisms to analyze what they call Dynamic Accuracy Estimation (DAE), which, according to them "expresses the estimated potential error of the provided location estimate". According to that work, DAE is identified by some authors as "accuracy estimation", "error estimation" or "confidence". As has been said in the introduction, it is usually depicted as a confidence circle centered in the estimated location on the map. The radius of the circle is the estimated error, and we call it error estimation. In that paper, the mechanisms to evaluate the error are divided into *rule-based methods* and *data-driven methods*. The first group analyzes the quality of the locations obtained, while the second group uses machine learning to analyze the quality of the predicted locations. In that work, a benchmarking framework is included to evaluate and compare several methods of DAE. It is, in fact, the first work that establishes the state of the art for DAE.

There have been several proposals to measure DAE. Moghtadaiee et al. [19] show that not always closer points in the geometric space are closer points in the signal space. Thus, they propose a calculation of the accuracy estimation based on the slope of the relation among the distance between points in the real space and in the signal space. Nikitin et al. [20] use the Cramer-Rao Lower Bound(CRLB) theory to estimate the best possible achievable accuracy in a single location in a magnetic field fingerprint map. Lemic et al.

From the analysed literature, it can be seen that the most commonly used mechanisms to give the error bound of a system either correspond to a common aggregated metric (equally applied to all position estimates) or need a training process in order to take into account location features, as the error can be different in different regions of the environment and in most of the cases error is calculated a posteriori. None of the proposed mechanisms takes into account the origin of the lack of accuracy in a single point, nor takes into account that the error bound is not only location but also time-dependent. i.e., in the same location, the error bound of a positioning system fluctuates depending on the dynamics of the environment.

On the other hand, the very first step of all the previously given methods is to measure a magnitude, like a distance, the intensity of a WiFi signal or a position. The obtained values have an uncertainty that is the common uncertainty of any measure. When these values are introduced in a formula, that uncertainty propagates through the calculations [21]. If the distribution of the measured values is Gaussian, we can apply error propagation theory (EPT) to take into account the propagation of the uncertainty. Nevertheless, this theory has been scarcely applied in indoor positioning systems. One example of application is in the calibration of an optical indoor positioning system by Rodriguez-Navarro et al. [22], or to analyze how it can be applied in WiFi fingerprinting with kNN [23].

Therefore, it is important to propose a metric that can give the expected error bound individually for any position estimate. In this work, we propose a mechanism to provide the error bound in a position estimate provided by kNN-based Wi-Fi fingerprinting.

III. Materials and Methods

In this section, we introduce the several elements involved in the proposed mechanism to calculate a position estimate and its associated error bound. First of all, we will review the fingerprinting-based method for indoor positioning and the kNN algorithm. Then we review the error propagation theory and explain the method proposed to estimate the error bound.

A. Fingerprinting-based indoor positioning

Fingerprinting-based indoor positioning has two phases: *offline* and *online*.

1) Offline (or training) phase: Let's suppose there are n_{ap} different access points (APs) in the scenario where the indoor positioning system is going to be deployed. The set of all APs is $\Omega = \{\omega_1, \ldots, \omega_{n_{ap}}\}$. The training database (or *radio map*) $\mathcal{R}^{tr}$ is made up of a set of *fingerprint* and their positions:

$$\mathcal{R}^{tr} = \{\mathcal{F}^{tr}, \mathcal{L}^{tr}\} \tag{1}$$

where tr indicates training, $\mathcal{F}^{tr}$ is the set of *fingerprints* obtained and $\mathcal{L}^{tr}$ their positions. The set $\mathcal{F}^{tr}$ is made up of n_{tr} *fingerprints*, saved as n_{tr} vectors of RSSI measures:

$$\mathcal{F}^{tr} = \{\lambda_1^{tr}, ..., \lambda_{n_{tr}}^{tr}\} \tag{2}$$

and each *fingerprint* λ_i^{tr} has n_{ap} RSSI values:

$$\lambda_i^{tr} = \left\{\rho_{i,1}^{tr}, ..., \rho_{i,n_{ap}}^{tr}\right\}, \; i \in [1, ..., n_{tr}] \tag{3}$$

The set $\mathcal{F}^{tr}$ can be viewed, from the point of view of machine learning algorithms, as a training database where the rows are the samples and the columns are the features.

On the other hand, $\mathcal{L}^{tr}$ is made up of n_{tr} positions, stored as vectors, representing the position associated with each sample:

$$\mathcal{L}^{tr} = \left\{\tau_1^{tr}, ..., \tau_{n_{tr}}^{tr}\right\} \quad (4)$$

where , τ_i^{tr} is the position of the $i-th$ training reference point. Each position is often provided with just a pair of coordinates (often longitude and latitude). In complex large environments, the position may also include the building, floor, or room labels. Each τ_i^{tr} can be expressed as follows:

$$\tau_i^{tr} = \left\{\nu_{i,1}^{tr}, ..., \nu_{i,n_l}^{tr}\right\}, \ i \in [1, ..., n_{tr}] \quad (5)$$

where $\nu_{i,p}^{tr}$ ($p \in [1, \ldots, n_l]$) is each one of the elements to determinate the position of the $i-th$ training reference point.

Note that, $\rho_{i,r}^{tr}$ ($r \in [1, \ldots, n_{ap}]$) is the RSSI value obtained for AP ω_r at position τ_i^{tr}, i.e. $\rho_{i,r}^{tr} = RSSI(\omega_r, \tau_i^{tr})$.

Since the RSSI values received from an AP can vary greatly even when the device used to capture the data remains stationary in the same position during the capture process, multiple measurements can be obtained at each position τ_i^{tr} to allow better capture of signal behaviour at this position. In these cases, the mean $\mu_{i,r}^{tr}$ and standard deviation $\sigma_{i,r}^{tr}$ of all measurements obtained at τ_i^{tr} can be calculated as follows:

$$\mu_{i,r}^{tr} = \frac{\sum_{m=1}^{n_m} \rho_{i,r}^{tr}(m)}{n_m} \quad (6)$$

$$\sigma_{i,r}^{tr} = \sqrt{\frac{\sum_{m=1}^{n_m} (\rho_{i,r}^{tr}(m) - \mu_{i,r}^{tr})^2}{n_m - 1}} \quad (7)$$

where n_m is the number of measurements obtained at τ_i^{tr} and $\rho_{i,r}^{tr}(m)$ is each one of these measurements.

Therefore, fingerprints λ_i^{tr} can be expressed as follows:

$$\lambda_i^{tr} = \left\{(\mu_{i,1}^{tr}, \sigma_{i,1}^{tr}), ..., (\mu_{i,n_{ap}}^{tr}, \sigma_{i,n_{ap}}^{tr})\right\}, \ i \in [1, ..., n_{tr}] \quad (8)$$

2) Online (operational/test) phase: In this phase, the device captures a *fingerprint* λ^{ts} (in this case ts indicates test) that will be compared with the *radio map* $\mathcal{R}^{tr}$ to generate an estimate of its unknown position τ^{ts}. In order to be able to compare the test sample λ^{ts} with the training ones, the order of the APs in the training and testing fingerprints have to match. Therefore, λ^{ts} must be composed of a *fingerprint* with n_{ap} RSSI values:

$$\lambda^{ts} = \left\{\rho_1^{ts}, ..., \rho_{n_{ap}}^{ts}\right\}, \quad (9)$$

Similarly, the position vector must have the same dimensions as its training counterparts:

$$\tau^{ts} = \left\{\nu_1^{ts}, ..., \nu_{n_l}^{ts}\right\} \quad (10)$$

In this phase, as in the training phase, it is also common to capture more than one measurement at the position to be estimated (i.e. at τ^{ts}). Then, the mean μ_r^{ts} and the standard deviation σ_r^{ts} of all measurements obtained at τ^{ts} can be calculated as follows:

$$\mu_r^{ts} = \frac{\sum_{m=1}^{n_m} \rho_r^{tr}(m)}{n_m} \quad (11)$$

$$\sigma_r^{ts} = \sqrt{\frac{\sum_{m=1}^{n_m} (\rho_r^{tr}(m) - \mu_r^{tr})^2}{n_m - 1}} \quad (12)$$

where n_m is the number of measurements obtained at τ^{ts} and $\rho_r^{ts}(m)$ is each one of these measurements.

Therefore, the fingerprints λ^{ts} can be expressed as follows:

$$\lambda^{ts} = \left\{(\mu_1^{ts}, \sigma_1^{ts}), ..., (\mu_{n_{ap}}^{ts}, \sigma_{n_{ap}}^{ts})\right\} \quad (13)$$

B. The kNN algorithm for indoor positioning

The kNN algorithm is a well-known classifier based on the concept of distance that compares the current sample with all the samples in the training set. In the simplest case ($k = 1$), the label of the current test sample will be the label of the closest training sample according to a distance function.

In the case of indoor positioning, the samples are the *fingerprints* and the labels are the positions. In the simplest case ($k = 1$), the position of the current test *fingerprint* will be the position of the closest training *fingerprint* in the feature space (i.e. in the fingerprint space). When $k > 1$, the final position corresponds to the centroid of the positions associated with the closest k *fingerprints* in what we call feature space, i.e. the vector space corresponding to RSSI values. In this case, the centroid position of the k nearest neighbours is calculated as follows:

$$\tau^{ts} = \frac{\sum_{h=1}^{k} \tau_{nn(h)}^{tr}}{k} \quad (14)$$

where τ^{ts} is the localisation of the test sample to be estimated and $nn(h)$ is a function that, given the $h-th$ neighbor, returns the index of the training sample that corresponds to the h-th nearest neighbor. Therefore, $\tau_{nn(h)}^{tr}$ is the position of the h-th nearest neighbor.

For estimating the distance between the test fingerprint λ^{ts} and a training one λ_i^{tr} in the feature space, we can use several distance functions, among which the most popular is the Euclidean distance, that can be expressed as follows:

$$d_i = d(\lambda^{ts}, \lambda_i^{tr}) = \sqrt{\sum_{r=1}^{n_{ap}} (\mu_r^{ts} - \mu_{i,r}^{tr})^2} \quad (15)$$

Note that we are assuming that several measurements have been captured at each point (training and test) and, therefore, we are using the fingerprints expressed as shown in Equations 8 and 13. Therefore, the mean values μ_r^{ts} and $\mu_{i,r}^{tr}$ are used to calculate the distances. The standard deviation will be taken into account in the error calculation.

C. Error propagation

To take into account how uncertainty propagates through calculations, we use EPT. In this frame, we consider a magnitude z that is given by a function f that depends on n_e independent variables:

$$z = f(x_1, \ldots, x_e, \ldots, x_{n_e}) \quad (16)$$

where f is supposed to have a Gaussian distribution.

The error of the z function will be given by:

$$(\Delta z)^2 = \sum_{e=1}^{n_e} \left(\frac{\partial f}{\partial x_e} \Delta x_e \right)^2 \quad (17)$$

where Δx_e is the error obtained when measuring the variable x_e. To estimate this error, it is common to take several measurements and calculate the standard deviation.

D. Estimating the error in the fingerprint distance

Uncertainty in indoor positioning comes from the fluctuations of measured data. Thus, when taking RSSI data from APs we take the mean value as representative of the value of every AP at a single point, but we also keep the standard deviation (see Equations 8 and 13). The standard deviation gives information about the stability and fluctuations that the signal of every single AP has at that single point.

The role played by the function f (see Equation 16) is played by the distance metric that we use in the kNN algorithm. In the current case, we use Euclidean distance given by equation 15. The variables x_e in Equation 16 are in our case the RSSI obtained at each location expressed by their mean values μ_r^{ts} and $\mu_{i,r}^{tr}$. Therefore, in our particular case $n_e = 2{*}n_{ap}$. Since we are assuming that several measurements have been captured for each location, the standard deviation of these measurements σ_r^{ts} and $\sigma_{i,r}^{tr}$ are used as the error when measuring these variables.

Applying equation 17 to the Euclidean distance, we obtain:

$$\begin{aligned}(\Delta d_i)^2 &= (\Delta d(\lambda^{ts}, \lambda_i^{tr}))^2 = \\ &= \frac{\sum_{r=1}^{n_{ap}} (\mu_r^{ts} - \mu_{i,r}^{tr})^2 \left[(\sigma_r^{ts})^2 + (\sigma_{i,r}^{ts})^2 \right]}{\sum_{r=1}^{n_{ap}} (\mu_r^{ts} - \mu_{i,r}^{tr})^2}\end{aligned} \quad (18)$$

Note that the estimation of $\Delta d(\lambda^{ts}, \lambda_i^{tr})$ is possible since several measurements have been obtained for each point (training and test), i.e. when using the definition of fingerprint as it is shown in Equations 8 and 13.

Our approach in calculating the error when estimating the distance between two fingerprints (Equation 18) not only uses the mean but also incorporates the standard deviation ($\sigma_r^{ts}, \forall r \in [1, \ldots, n_r]$). This method is crucial as it allows us to estimate an error bound during online testing when the user seeks to be located. Therefore, the error bound varies based on the area since RSSI variability can differ across locations. Low RSSI fluctuations in an area will result in a low error bound estimate, while high variability will lead to a larger error bound. Incorporating the standard deviation in our methodology provides a more comprehensive and accurate estimation of the error bound, leading to greater reliability and effectiveness of indoor localization systems.

E. Proposed method to estimate the error bound

Figure 1 shows a scheme of the proposed method to estimate the error bound for a single test point. Algorithm 1 shows the same process in algorithmic language.

Figure 1 is divided into three regions: the first region, green, on the left, corresponds to Euclidean space. There are marked the training points (squares) and the user/test point (a circle). Here, each point is characterised by its Cartesian coordinates. Therefore, in our particular case, $n_l = 2$.

The second region, light orange, in the centre, corresponds to the feature space. In this region, only RSSI data from every fingerprint is taken into account. This region is divided between the upper part and the lower part.

The upper part of this section illustrates the calculation of the distance metric d_i in the feature space between the test sample λ^{ts} and every training one λ_i^{tr} (Line 2 at Algorithm 1). Δd_i is the error measuring the distance d_i associated and it is determined using equation 18 (Line 3).

To predict the location of the test sample, we choose the k nearest neighbours, i.e., the k samples of the training dataset with the lower distance d_i to the test sample. In this way, we obtain the k elements of the training set that we will use to predict the position (Lines 5 and 6).

To estimate the error bound, as illustrated in the lower part of this section, we extract n_s distance samples among the candidates inside the interval $[d_i - \Delta d_i, d_i + \Delta d_i]$. They are called $\delta_i^{(s)}$ and can be calculated as follows (Lines 8 to 11):

$$\delta_i^{(s)} = d_i + \Delta d_i \cdot \psi \quad (19)$$

where $s \in [1, \ldots, n_s]$ and $\psi = \mathcal{U}(-1, 1)$, i.e. ψ gives a number between -1 and 1 using a uniform distribution. Then, we choose the k nearest neighbours for each sample (Lines 12 and 13). In doing so, we obtain n_s sets of k elements of the training set that we will use to establish the area of uncertainty around the estimated position.

The reason for having different candidates for an estimated position is that we take into account the uncertainty of the distance calculation. Hence, we hypothesise that the region in which the candidates lie corresponds to the representation of such uncertainty in Euclidean space.

Finally, the right-hand side of Figure 1 illustrates the transition from feature space back to a Cartesian coordinate system in Euclidean space. We estimate the predicted position $\tau^{ts} = C$, depicted by a green square, as the centroid of the k nearest neighbours obtained in the previous step using the set of distances D (Line 6). Next, we obtain $C^{(1)}, \ldots, C^{(n_s)}$ position predictions (red circles) from the n_s samples obtained from the set of distances $D^{(1)}, \ldots, D^{(n_s)}$ (Line 13). The next step is to estimate $C^{(max)}$ as the maximum distance between the estimated position C and the positions $C^{(s)}$ (Line 15).

When we use the farthest candidate from our estimate as the error bound (i.e. when $\Delta\tau^{ts} = d(C, C^{(max)})$), we assume that the real position of the user has a high probability of being inside this circle with centre C and radius the distance between C and $C^{(max)}$. To have more control over the error bound size, we can pick a candidate that is closer to our estimate and within a certain percentage (68.26% or 95.44%, for example). This gives us a smaller circle with the probability of finding the real position inside it. This is controlled by the hyperparameter q.

Therefore, the error boundary of the prediction is determined as follows (Line 16):

$$\Delta\tau^{ts} = Q(C, C^{(1)}, \ldots, C^{(n_s)}, q) \quad (20)$$

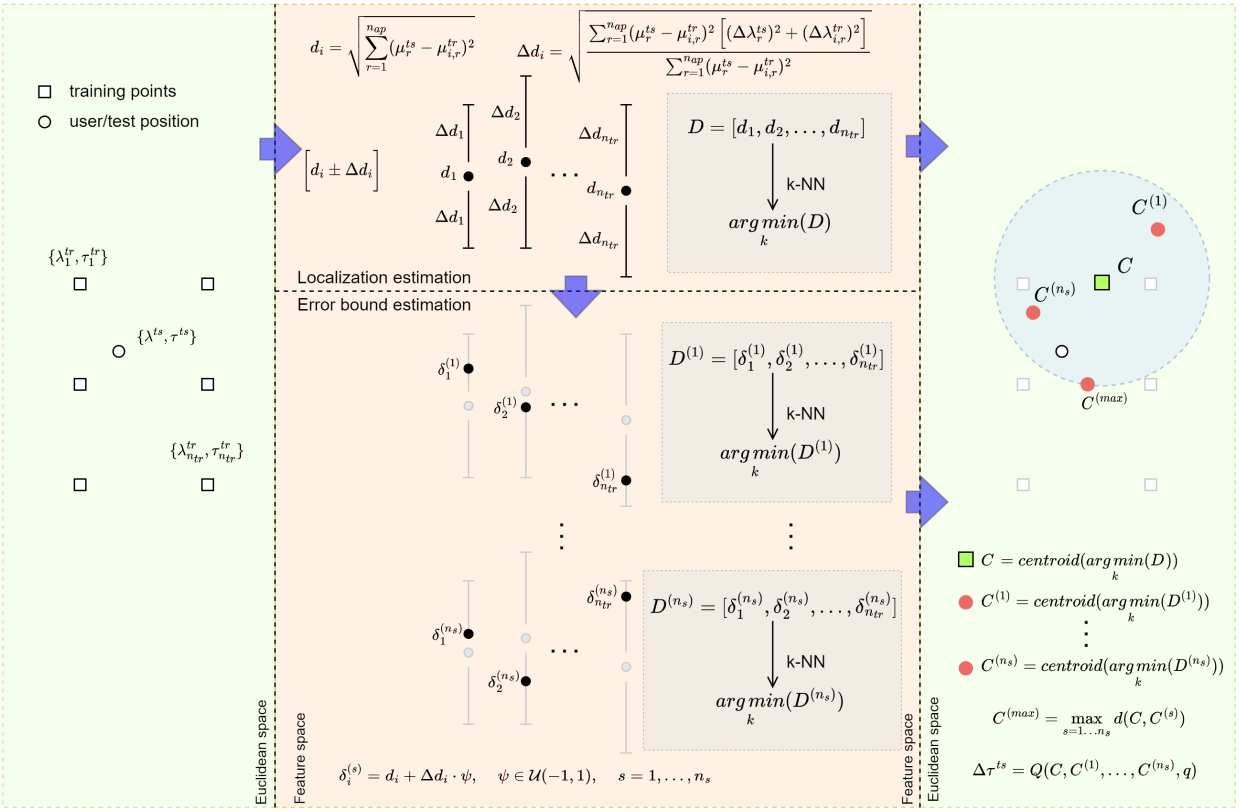


Fig. 1. A scheme of the proposed method to estimate the error bound.

where Q is a function that estimates the distances between C and each $C^{(s)}$, and after sorting the resulting distance vector, gets the value at the quantile expressed by hyperparameter q. Note that $Q(C, C^{(1)}, \ldots, C^{(n_s)}, q = 100)$ gets the distance between C and $C^{(max)}$.

Using the distance between C and $C^{(max)}$ maximizes the probability of the actual position of the user being inside the circle. As said in Section I, DAE usually provides a 68^{th} percentile confidence level. The hyperparameter q has been introduced to better simulate what DAE does.

IV. Experiments and Results

In this section we present the experiments and results that were conducted to evaluate the proposed algorithm. We start by describing the data we used for the experiments.

A. Data

The data used in this work consist of a series of databases that reflect the signal strength of the several Wi-Fi signals detected in a set of points located in indoor spaces. In particular, we used *MAN1* [24], [25], and UJILIB [26], [27].

MAN1 Wi-Fi dataset has 14300 training samples obtained at $n_{tr} = 130$ different reference points (i.e. at each reference point $n_m = 110$ measurements have been obtained), 460 test samples obtained at 46 different points (i.e. at each test point there are $n_m = 10$ measurements), and $n_r = 28$ APs.

The *UJILIB* is composed of a series of small datasets that were obtained by measuring the RSSI data at different moments in time. The RSSI data were collected at the same locations for both training and testing purposes. The capturing process took 25 months. In the first month, fifteen small datasets were captured. Then, a new small dataset was captured each month. In this paper, the first fifteen small datasets, covering the first month, have been used. We have merged all the samples into a unique dataset. Since data were captured on two different floors, we have divided the resulting dataset into two different ones, each one for each floor. They will be called *UJILIBM01F3* and *UJILIBM01F5* in the rest of the article.

The datasets *MAN1*, *UJILIBM01F3* and *UJILIBM01F5* include fingerprints (RSSI data) for training and testing purposes which are labelled with their current locations. Locations are represented as 2D positions (x,y) in a local coordinate system expressed in meters that depends on the dataset, i.e. the origin of coordinates is different for *MAN1* and *UJILIB* as datasets were collected in Germany and Spain respectively.

Figure 2 shows the localisation of the training (squares) and test points (circles) for both datasets. MAN1 dataset covers a wider space ($2100\,\mathrm{m}^2$ aprox.) than UJILIBM01F* ones ($208\,\mathrm{m}^2$ aprox.).

B. Experimental set-up

For each dataset and for each test point, $\Delta\tau^{ts}$ has been estimated using different q values. In particular we have used $q \in [68.26\%, 95.44\%, 99.73\%, 100\%]$. We have selected these values since, in EPT, given a magnitude z, it is expected that the 68.26% of the real values lies into the error bound determined by a circle with Δ_z as radius, the 95.44% for

Algorithm 1 Proposed algorithm to estimate the error bound for a single test point.

Require: λ^{ts}: Fingerprint of the test point.
Require: $\mathcal{R}^{tr}$: Radio map composed of n_{tr} training points with fingerprints λ_i^{tr} and positions τ_i^{tr}, ($i \in [1, \ldots, n_{tr}]$).
Require: n_s: Number of samples to be performed.
Require: k: Number of neighbours in kNN algorithm.
Require: q: Quantile to be used for estimating the error bound.
Ensure: : τ^{ts}: Estimated position of the test point.
Ensure: : $\Delta\tau^{ts}$: Error bound estimation.

for $i \leftarrow 1$ to n_{tr} **do**
  $d_i \leftarrow d(\lambda^{ts}, \lambda_i^{tr})$ (Eq. 15)
  $\Delta d_i \leftarrow \sqrt{\Delta d(\lambda^{ts}, \lambda_i^{tr})}$ (Eq. 18)
end for
$D \leftarrow [d_1, \ldots, d_{n_{tr}}]$
$C \leftarrow centroid(\underset{k}{arg\,min}(D))$
for $s \leftarrow 1$ to n_s **do**
  for $i \leftarrow 1$ to n_{tr} **do**
    $\psi \leftarrow \mathcal{U}(-1, 1)$
    $\delta_i^{(s)} \leftarrow d_i + \Delta d_i \cdot \psi$
  end for
  $D^{(s)} \leftarrow [\delta_1^{(s)}, \ldots, \delta_{n_{tr}}^{(s)}]$
  $C^{(s)} \leftarrow centroid(\underset{k}{arg\,min}(D^{(s)}))$
end for
$C^{(max)} \leftarrow \max_{s=1,\ldots,n_s} d(C, C^{(s)})$
$\Delta\tau^{ts} \leftarrow Q(C, C^{(1)}, \ldots, C^{(n_s)}, q)$

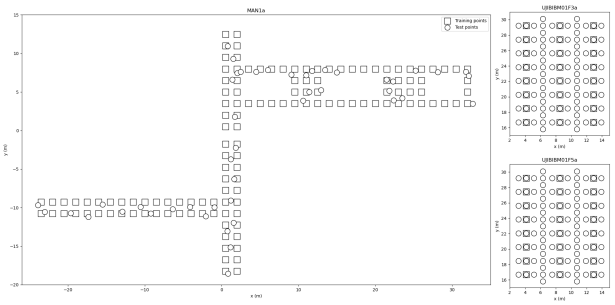


Fig. 2. Training (squares) and testing (circles) reference points used in MAN1 dataset (left), UJILIBM01F3 (top, right) and UJIBINM01F5 (bottom, right).

$2 * \Delta_z$, and 99.73% for $3 * \Delta_z$. In the case of using $q = 100$, it is expected all the real values lie inside the defined error bound.

The number of sampled distances has been empirically set to $n_s = 1000$. This number must be a compromise between obtaining always the same $C^{(max)}$ value (it needs a big enough number of iterations due to the use of random sampling) and the processing time.

In the kNN algorithm, the number of neighbours has been empirically set to $k = 3$. Experiments, not presented in this paper, using other k values have been performed concluding that the k value does not change significantly the results obtained.

In the datasets that we used for our experiments, there are some cases where no data has been received from a specific AP. In these cases, the dataset uses a 100 value to indicate the missing data. We decided to replace these values with the minimum value in each dataset minus one.

C. Results

Figure 3 shows some examples of the error bounds obtained for several test points of both datasets. The empty circles correspond to the true locations of the test points and the green squares are the localisation estimated for each case (i.e. τ^{ts}). The dotted green and orange circles are the error bound estimated when using $q = 68.26\%$ and $q = 95.44\%$ quantiles, respectively. Figure 4 shows that different points have different error bounds. The size of the error bound is related to the uncertainty of the RSSI signal both from the training data and from the test data point, as discussed in Section III. In the cases when the uncertainty of the RSSI signal (in training and/or in the test data) is significant, for instance in estimated location at approximate coordinates $[-20, -10]$ in MAN1 dataset, our proposed method obtains a bigger error bound (i.e a bigger circle) indicating that the uncertainly about the position estimated is greater. In the opposite case, for instance, in the estimated location at approximate coordinates$[-12, 5]$ the uncertainty of the RSSI signal is lower. Then, our proposed method is able to obtain a lower error-bound estimate. This means that the proposed method is able to consider both the environment when we make the radio map and how the environment changes affect accuracy later on.

In Figure 4, we compare the estimated error bounds for the three datasets and four quantiles. The test points inside the error bound estimated (blue points) are the ones whose real positions fall inside the error bound considering a given quantile q. The red ones represent the opposite case, i.e. the scenario in which the real positions fall outside the bound error calculated considering q. Figure 4 shows how the proposed method has a performance as expected in a DAE system. On the one hand, when increasing q the probability of the real position being inside the circle is very high, but the prize to be paid is having a very high circle. This can not be appropriate for some real applications. On the other hand, by reducing q, a smaller circle can be obtained but the probability of the real position being inside the circle is also reduced. The correct value of the q parameter depends on the particular application where the indoor localisation system will be deployed.

Table I shows the number and percentage of test points where the real position is inside (or outside) the error bound estimated for the experiment shown in Figure 4. Note that the percentages shown in the table are approximately in accordance with the chosen quantile, but there are some discrepancies. This may be due to the not exactly Gaussian behaviour of the RSSI signal. The signal strength can vary significantly over time and space due to environmental factors such as changes in temperature, humidity, and interference from other wireless devices. These non-stationary effects can cause the RSSI distribution to deviate from a Gaussian distribution. Moreover, in some cases, the signal strength can become saturated due to a very strong signal or overloading

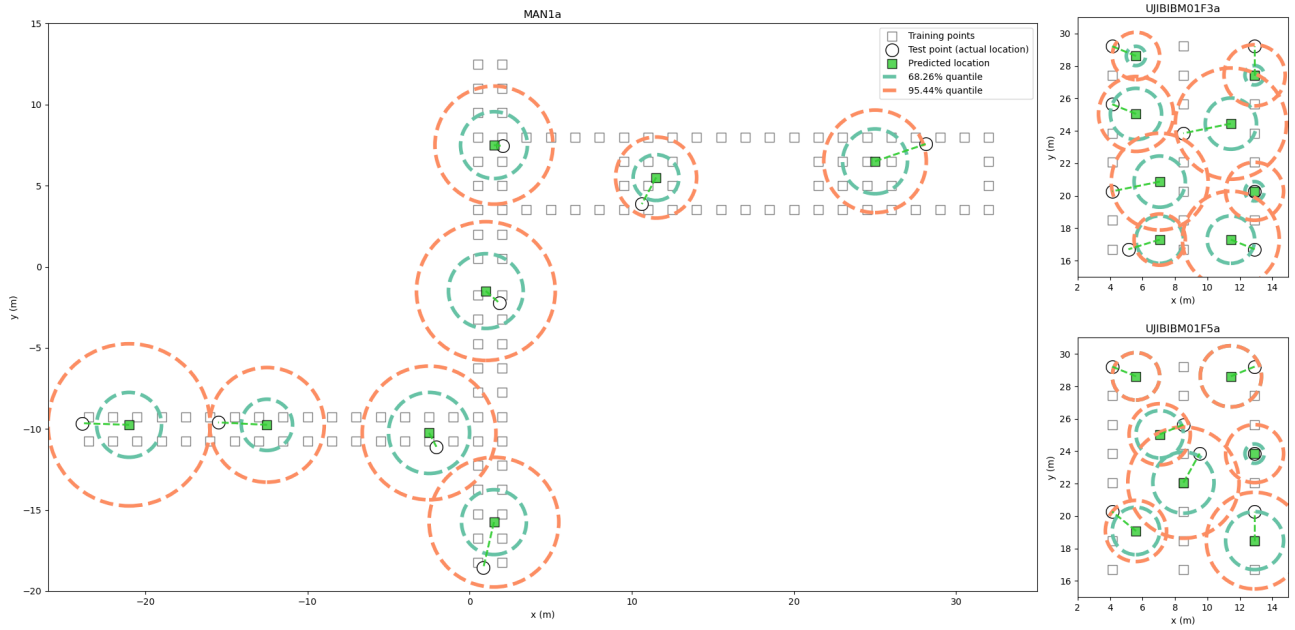


Fig. 3. Some examples of the error bound obtained for several testing points in the three datasets.

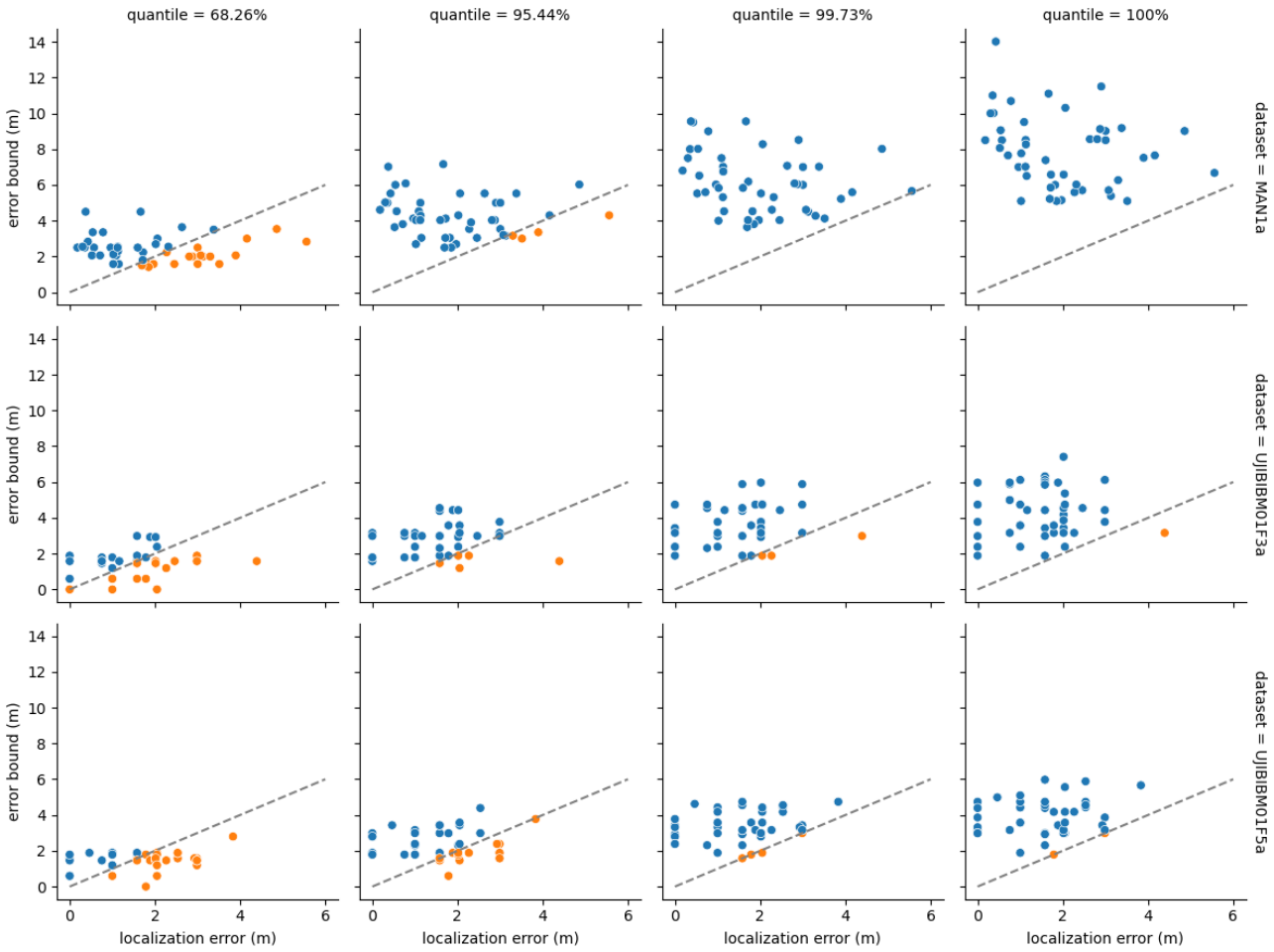


Fig. 4. Blue points are test points where the real localisation is inside the error bound given different values of the percentile. The orange ones are the ones outside. The top row is for MAN1 dataset, the middle one is for UJILIBM01F3 and the bottom one is for UJILIBM01F5 dataset.

TABLE I
NUMBER AND PERCENTAGE OF TEST POINTS WHERE THE REAL POSITION IS INSIDE (OR OUTSIDE) THE ERROR BOUND ESTIMATED.

Dataset	q	In	Out	Percentage
MAN1a	68.26%	27	19	58.70%
MAN1a	95.44%	42	4	91.30%
MAN1a	99.73%	46	0	100.00%
MAN1a	100%	46	0	100.00%
UJIBIBM01F3a	68.26%	20	26	43.48%
UJIBIBM01F3a	95.44%	39	7	84.78%
UJIBIBM01F3a	99.73%	43	3	93.48%
UJIBIBM01F3a	100%	45	1	97.83%
UJIBIBM01F5a	68.26%	19	27	41.30%
UJIBIBM01F5a	95.44%	30	16	65.22%
UJIBIBM01F5a	99.73%	40	6	86.96%
UJIBIBM01F5a	100%	43	3	93.48%

of the receiver. This can cause the RSSI distribution to become skewed or truncated.

Overall, the proposed method performs as expected by accounting for the variability of the signal and its effects on the final prediction using the EPT.

V. CONCLUSIONS

In this paper, we propose an indoor positioning system that gives an estimated position with its corresponding error bound. The error bound is calculated for every single estimated position. Thus, it takes into account variability in space but also changes in time, since every new estimation of a position calculates a new error bound.

We take into account changes in environment, time and space, because the error bound is calculated every time in the online phase with the measured data at that point. Therefore, the error bound will change if an AP has changed or we use a different smartphone or in any other case. These variations of the environment can modify the distances of the kNN algorithm or the standard deviations of the new measures of the signal, and those changes can give a different result.

The positioning method proposed is based on Wi-Fi fingerprinting with the kNN algorithm. The error bound is obtained by applying error propagation theory in the feature space when the Euclidean distance among signals is calculated. That gives an interval of distances among every two fingerprints. Thus, the nearest neighbours can be different when different points of the interval are considered. Therefore, we consider several possible nearest neighbours, and every one of these possibilities gives a candidate to be the estimated position. The radius of the error bound is given by the quantile of candidates considered, ordered by their distance to the predicted position.

Future work should focus on studying the impact of input data distribution on obtained results. The EPT assumes that the source data follows a Gaussian distribution, which may not be applicable to RSSI signals in general. Therefore, it may be worthwhile to explore techniques to transform the RSSI input into a Gaussian distribution prior to calculating the error bound and to evaluate the impact of such techniques on the results. In addition, it can also be of interest to study how to adapt the shape of the bound error to the environment, exploring non-symmetric ways of characterizing the error bound, such as ellipses, that reflect both the uncertainties and the spatial distribution of the training set.

APPENDIX I
NOMENCLATURE

Symbols meaning *Number of*:

- n_{ap}: Number of access points in the scenario. It is also the dimension of the fingerprint vectors.
- n_{tr}: Number of training reference points included in the Radio map.
- n_l: Number of dimensions of each position vector.
- n_s: Number of samples performed.
- n_e: Number of variables of the f function.
- n_m: Number of measurements obtained at each position.
- k: Number of neighbors in the kNN algorithm.

Symbols for sets or vectors:

- Ω: The set for including all access points.
- $\mathcal{R}^{tr}$: Radio map containing all training reference points.
- $\mathcal{F}^{tr}$: Set with the fingerprints of the training reference points.
- $\mathcal{L}^{tr}$: Set with the position vectors of the training reference points.
- λ_i^{tr}: Fingerprint of the $i-th$ training reference point.
- λ^{ts}: User (or test) fingerprint.
- τ_i^{tr}: Position of the $i-th$ training reference point.
- τ^{ts}: User (or test) position to be estimated.
- D: Set of all distances between the user fingerprint and each one into the Radio map.
- $D^{(s)}$: $s-th$ set of all sampled distances between the user fingerprint and each one into the Radio map.

Symbols for iterators:

- i: iterator across training samples, $i \in [1, \ldots, n_{tr}]$.
- r: iterator across access points, $r \in [1, \ldots, n_{ap}]$.
- p: iterator across the position vector, $p \in [1, \ldots, n_l]$.
- $^{(s)}$: iterator across distances sampled, $s \in [1, \ldots, n_s]$.
- h: iterator across neighbors, $h \in [1, \ldots, k]$.
- e: iterator across variables in f function.
- m: iterator across measurements, $m \in [1, \ldots, n_m]$.

Symbols related to distances:

- d_i: distance between the user fingerprint and the $i-th$ training one.
- Δd_i: Error obtained when calculating d_i.
- $\delta_i^{(s)}$: $s-th$ sampled distance between the user fingerprint and the $i-th$ training one.

Symbols related to centroids:

- C: Centroid calculated using the k nearest distances between the user fingerprint and the training ones.
- $C^{(s)}$: Centroid calculated using the k nearest $s-th$ sampled distances between the user fingerprint and the training ones.

Other symbols:

- $\rho_{i,r}^{tr}$: RSSI value of the $i-th$ training reference point related to $r-th$ access point.
- ρ_r^{ts}: RSSI value of the user (or test) point related to $r-th$ access point

- $\mu_{i,r}^{tr}$: Mean value of all RSSI values obtained at the $i-th$ training reference point related to $r-th$ access point.
- $\sigma_{i,r}^{tr}$: Standard deviation of all RSSI values obtained at the $i-th$ training reference point related to $r-th$ access point.
- μ_{r}^{ts}: Mean value of all RSSI values obtained at the user point related to $r-th$ access point.
- σ_{r}^{ts}: Standard deviation of all RSSI values obtained at the user point related to $r-th$ access point.
- $\nu_{i,p}^{tr}$: $p-th$ element of the position vector of the $i-th$ training reference point.
- ν_{p}^{ts}: $p-th$ element of the position vector of the user point.
- ω_r: $r-th$ access point, $\omega_r \in \Omega$.
- $C^{(max)}$: Maximum distance between the estimated centroid C and the ones $C^{(s)}$ obtained using the sampled distances.
- q: Quantile value for obtaining the final error bound.
- ψ: Random uniform variable between $[-1, 1]$
- $\Delta\tau^{ts}$: Desired error bound of the user position.
- f: A generic function.
- z: A generic magnitude.
- Q: A function for ordering a set of distances and getting the one located at a particular quantile.

REFERENCES

[1] P. Marcus, M. Kessel, and M. Werner, "Dynamic Nearest Neighbors and Online Error Estimation for SMARTPOS," *International Journal on Advances in Internet Technology*, vol. 6, pp. 1–11, 1 2013.

[2] D. Zou, W. Meng, and S. Han, "An Accuracy Estimation Algorithm for Fingerprint Positioning System," in *2014 Fourth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC'14)*, 2014, pp. 573–577.

[3] F. Lemic, V. Handziski, M. Aernouts, *et al.*, "Regression-Based Estimation of Individual Errors in Fingerprinting Localization," *IEEE Access*, vol. 7, pp. 33 652–33 664, 2019.

[4] F. Pukelsheim, "The three sigma rule," *The American Statistician*, vol. 48, no. 2, pp. 88–91, 1994.

[5] J. Conesa, A. Pérez-Navarro, J. Torres-Sospedra, *et al.*, *Geographical and fingerprinting data for positioning and navigation systems: challenges, experiences and technology roadmap*. Elsevier, 2018.

[6] Carl von Ossietzky, "Error theory and regression analysis," pp. 34–53, 2008.

[7] F. Potortì, A. Crivello, P. Barsocchi, *et al.*, "Evaluation of Indoor Localisation Systems: Comments on the ISO/IEC 18305 Standard," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–7.

[8] N. Moayeri, "Response to "evaluation of indoor localisation systems: Comments on the ISO/IEC 18305 standard"," in *WiP Proceedings of the Eleventh International Conference on Indoor Positioning and Indoor Navigation*, A. Pérez-Navarro, R. Montoliu, and J. Torres-Sospedra, Eds., ser. CEUR Workshop Proceedings, vol. 3097, CEUR-WS.org, 2021.

[9] V. Renaudin, M. Ortiz, J. Perul, *et al.*, "Evaluating Indoor Positioning Systems in a Shopping Mall: The Lessons Learned from the IPIN 2018 Competition," *IEEE Access*, vol. 7, pp. 148 594–148 628, 2019.

[10] F. Potorti, S. Park, A. Crivello, *et al.*, "The IPIN 2019 Indoor Localisation Competition—Description and Results," *IEEE Access*, vol. 8, pp. 206 674–206 718, 2020.

[11] F. Potorti, J. Torres-Sospedra, D. Quezada-Gaibor, *et al.*, "Off-line Evaluation of Indoor Positioning Systems in Different Scenarios: The Experiences from IPIN 2020 Competition," *IEEE Sensors Journal*, pp. 5011–5054, 2021.

[12] H. Zhang, Z. Wang, W. Xia, *et al.*, "Weighted adaptive knn algorithm with historical information fusion for fingerprint positioning," *IEEE Wireless Communications Letters*, vol. 11, no. 5, pp. 1002–1006, 2022.

[13] J. Torres-Sospedra, D. P. Quezada Gaibor, J. Nurmi, *et al.*, "Scalable and efficient clustering for fingerprint-based positioning," *IEEE Internet of Things Journal*, vol. 10, no. 4, pp. 3484–3499, 2023.

[14] X. Yang, Y. Zhuang, F. Gu, *et al.*, "DeepWiPos: A Deep Learning-Based Wireless Positioning Framework to Address Fingerprint Instability," *IEEE Transactions on Vehicular Technology*, pp. 1–16, 2023.

[15] A. Alitaleshi, H. Jazayeriy, and J. Kazemitabar, "EA-CNN: A smart indoor 3D positioning scheme based on Wi-Fi fingerprinting and deep learning," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105 509, 2023.

[16] J. Bi, M. Zhao, G. Yao, *et al.*, "PSOSVRPos: WiFi indoor positioning using SVR optimized by PSO," *Expert Systems with Applications*, vol. 222, p. 119 778, 2023.

[17] F. Lemic and J. Famaey, "Artificial Neural Network-based Estimation of Individual Localization Errors in Fingerprinting," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC'20)*, IEEE, 2020, pp. 1–6.

[18] G. G. Anagnostopoulos and A. Kalousis, "Can I Trust This Location Estimate? Reproducibly Benchmarking the Methods of Dynamic Accuracy Estimation of Localization," *Sensors*, vol. 22, p. 1088, 3 2022.

[19] V. Moghtadaiee, A. G. Dempster, and B. Li, "Accuracy indicator for fingerprinting localization systems," IEEE, Apr. 2012, pp. 1204–1208.

[20] A. Nikitin, C. Laoudias, G. Chatzimilioudis, *et al.*, "Indoor localization accuracy estimation from fingerprint data," 2017.

[21] J. R. Taylor, *Introduction to error analysis. The study of uncertainties in physical measurements*, 2nd. Edition. University Science Books, 1997, pp. 1–343.

[22] D. Rodríguez-Navarro, J. Lázaro-Galilea, I. Bravo-Muñoz, *et al.*, "Mathematical model and calibration procedure of a psd sensor used in local positioning systems," *Sensors*, vol. 16, p. 1484, 9 Sep. 2016.

[23] A. Perez-Navarro, "Accuracy of a single point in knn applying error propagation theory," IEEE, Nov. 2021, pp. 1–7.

[24] T. King, T. Butter, M. Brantner, *et al.*, "Distribution of fingerprints for 802.11-based positioning systems," in *International Conference on Mobile Data Management (MDM'07)*, 2007.

[25] T. King, S. Kopf, T. Haenselmann, *et al.*, *CRAWDAD dataset mannheim/compass (v. 2008-04-11)*, [available] https://crawdad.org/mannheim/compass/20080411, 2008.

[26] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, *et al.*, "Long-term wifi fingerprinting dataset for research on robust indoor positioning," *Data*, vol. 3, no. 1, 2018.

[27] G. M. Mendoza-Silva, P. Richter, J. Torres-Sospedra, *et al.*, *Long-Term Wi-Fi fingerprinting dataset and supporting material*, version 2.2, Zenodo, Apr. 2020.

Antoni Perez-Navarro Dr. Antoni Perez-Navarro has a PhD in Physics for the Universitat Autònoma de Barcelona (UAB) since 2000. He got his bachelor in physics, also from UAB, in 1995. He is lecturer in the Computer Science, Multimedia and Telecommunication department (EIMT department) at Universitat Oberta de Catalunya (UOC), since 2005. His main research interests are indoor positioning, prevention of diseases via technology, and e-Learning of sciences. He is member of the Steering Committee of IPIN.

Raúl Montoliu Dr. Raúl Montoliu received the B.S degree in computer sciences in 1988 from the Universitat Jaume I of Castellon (Spain). In 2008, he received the Ph. D. degree in advanced computer systems from the same university. He is currently an Associate Professor at the Department of Computer Science and Engineering and Senior Researcher at the Institute of New Imaging Technologies (INIT), both from Jaume I University. His current research interests include indoor positioning and indoor navigation, Human activity monitoring, and Game AI.

Emilio Sansano-Sansano Emilio Sansano-Sansano received the B.S. degree in industrial engineering from the Polytechnic University of Valencia (UPV), Spain, in 2002, the B.S. degree in computer science from the National Distance Education University (UNED), Spain, in 2017, and the Ph.D. degree in computer science from Jaume I University (UJI) in 2020. He is an Assistant Professor with the Department of Industrial Systems Engineering and Design (DESID). He is also a member of the Institute of New Imaging Technologies (INIT). His current research interests include machine learning and deep learning with applications to smart environments and human activity recognition.

Marina Martinez-Garcia Marina Martinez-Garcia received the B.S. degree in mathematics from the Polytechnic University of Catalonia (UPC), Spain, in 2009, and the Ph.D. degree in computer science from Pompeu Fabra University (UPF) in 2014. She is a Professor in the mathematics Department of Universitat Jaume I, Castelló Spain. He is also a member of the Institut de Matemàtiques de Castello (IMAC). Her current research interests include machine learning and deep learning with applications in neuroscience.

Ruben Femenia Rubén Femenía is a student in Computer Science at the Universitat Jaume I of Castellon (Spain) with a collaboration grant. His current research interests include artificial intelligence, indoor positioning, and indoor navigation.

Joaquín Torres-Sospedra received his PhD from Universitat Jaume I in 2011. He is Senior Researcher at the University of Minho (Guimarães, Portugal), where he works on Indoor Positioning (Wi-Fi, BLE, VLC) and Machine Learning for Industrial applications. He authored 160+ articles in journals and conferences and supervised 19 Master and 5 PhD Students. Currently, he supervises 4 PhD students. He is the chair of the IPIN International Standards Committee and IPIN Smartphone Competition.