A Comprehensive and Reproducible Comparison of Clustering and Optimization Rules in Wi-Fi Fingerprinting

Joaquín Torres-Sospedra, Philipp Richter, Adriano Moreira, Germán M. Mendoza-Silva, Elena Simona Lohan, Sergio Trilles, Miguel Matey-Sanz and Joaquín Huerta

Abstract—Wi-Fi fingerprinting is a well-known technique used for indoor positioning. It relies on a pattern recognition method that compares the captured operational fingerprint with a set of previously collected reference samples (radio map) using a similarity function. The matching algorithms suffer from a scalability problem in large deployments with a huge density of fingerprints, where the number of reference samples in the radio map is prohibitively large. This paper presents a comprehensive comparative study of existing methods to reduce the complexity and size of the radio map used at the operational stage. Our empirical results show that most of the methods reduce the computational burden at the expense of a degraded accuracy. Among the studied methods, only k-means, affinity propagation, and the rules based on the strongest access point properly balance the positioning accuracy and computational time. In addition to the comparative results, this paper also introduces a new evaluation framework with multiple datasets, aiming at getting more general results and contributing to a better reproducibility of new proposed solutions in the future.

Index Terms—Indoor positioning, Wi-Fi fingerprinting, clustering, computational costs, time complexity, benchmarking, reproducibility

1 Introduction

Location information bridges the gap between the physical and the digital worlds, creating new opportunities and challenges. As smart devices and pervasive mobile connectivity are increasingly penetrating our daily lives, more and more applications and location-based service (LBS) are built on the location awareness.

Outdoors, Global Navigation Satellite System (GNSS) technologies successfully provide position estimates, even in low satellite coverage situations, as long as they are combined with other well-known technologies such as inertial sensors, cellular networks, or IEEE 802.11 Wireless LAN (Wi-Fi) [1, 2, 3]. However, people spend about 80 % of their time indoors [4, 5], so indoor positioning and tracking is of high relevance for LBS. The difficulty of achieving a model that fits every indoor environment and which can deal with particularities such as signal multipath and heterogeneity of devices and building structures, makes the indoor location estimation a challenge [6]. Despite that, Wi-Fi fingerprinting (FP) is among the preferred indoor positioning technologies.

- *J. Torres-Sospedra, is with UBIK Geospatial Solutions, Spain. E-mail: torres@ubikgs.com*
- *P. Richter is with u-blox, Finland. E-mail: philipp.richter@u-blox.com*
- *A. Moreira is with the Algoritmi Research Centre, Universidade do Minho, Portugal. E-mail: adriano.moreira@algoritmi.uminho.pt*
- *J. Torres-Sospedra, G. M. Mendoza-Silva, S. Trilles, M. Matey-Sanz and J. Huerta are with the Institute of New Imaging Technologies, Universitat Jaume I, Spain. E-mail: {jtorres,gmendoza,strilles,mmatey,huerta}@uji.es*
- *E. S. Lohan is with the Dept. of Electronics and Communications Engineering, Tampere University, Finland. E-mail: elena-simona.lohan@tuni.fi*

Manuscript received xx December 2019; revised xx April 2020; accepted xx xxxx 2020. Date of publication xx xxxx 2020; date of current version xx xxxx 2020. (Corresponding author: Joaquín Torres-Sospedra.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109-TMC.2020.XXXXXXX

Wi-Fi fingerprinting exploits the detected Received Signal Strength (RSS) from Wi-Fi Access Points (APs) – the Wi-Fi fingerprint – to predict user's arbitrary location. In contrast to other approaches using Wi-Fi as the main positioning technology (e.g. proximity or ranging), Wi-Fi fingerprinting does not require information about the position of the APs. Wi-Fi fingerprinting relies on a set of fingerprints taken at well-known positions for the position estimation; i.e., Wi-Fi FP requires a reference dataset (or *radio map*) to operate. Different well-known methods tackle this problem, including the Nearest Neighbour (NN) algorithm k-NN [7], Gaussian kernels [8], Bayesian models [9], Neural Networks [10] and, even, Deep Learning [11, 12].

Among the techniques mentioned above, the ones based on advanced Machine Learning (ML) are the most accurate ones [13]. The high accuracy comes at the expense of high complexity; which is often prohibitive for smartphone implementation [14]. The other methods balance better positioning accuracy and computational complexity. Especially k-NN stands out, because it is simple but able to achieve very good positioning performance. For these reasons, k-NN has been widely adopted [7, 15, 16] –even on international competitions [17, 18, 19]– and we adopt it as base algorithm in this study. Nevertheless, the computational complexity can be an issue and requires further attention; especially because of the inherent trade-off between positioning accuracy and computational complexity.

Most FP methods (except the advanced ML methods) share the drawback that in the on-line phase the computational costs increase the with number of fingerprints [20]. This operation is the computationally most demanding operation [21], because for each operational fingerprint the distance to all reference fingerprints are calculated. How-

ever, the computational load in the on-line phase can be reduced if the distance calculation is restricted to a subset of reference fingerprints relevant to the operational fingerprint.

The approaches to alleviate the computational burden in the operational phase can be divided in two categories, namely *clustering* and *optimization rules*. Some of them focus on creating groups (clusters) of similar fingerprints off-line, and then apply a two-level search in the on-line (operational) phase, for instance, using k-Means clustering to split the radio map and create the cluster centroids [22]. For each operational fingerprint, the distances to the centroids are first computed and, then, the distances to all the reference fingerprints in the nearest cluster are calculated to estimate the final position (see Figure 1a). Other works correspond to optimization rules (heuristics) based on signal propagation characteristics, where the distance calculation is restricted to the reference fingerprints that are relevant (see Figure 1b). For instance, keeping the reference fingerprints whose strongest AP matches the operational one [23].

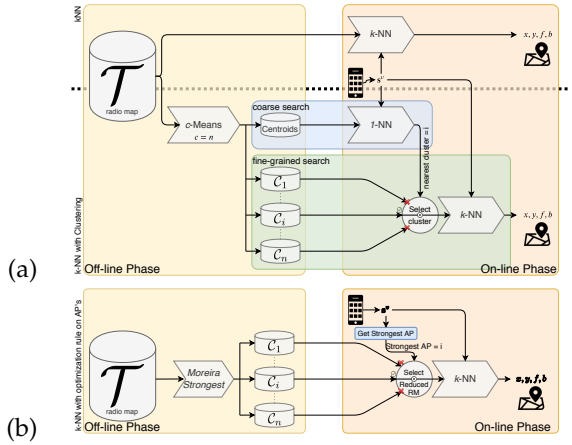

Fig. 1. Workflow of Wi-Fi FP. (a) based on k-NN with and without clustering (e.g., k-Means Clustering [22]). (b) based on k-NN with an optimization rule (e.g. Moreira Strongest [23]).

Given the large amount of different approaches, either clustering methods [8, 22, 24, 25] or optimization rules [23, 26, 27], it would be interesting to know which of the proposed methods achieves the best trade-off between positioning accuracy and computational complexity. Most of the studies focus on the positioning accuracy only. Moreover, these studies are not comparable, as they differ in their experimental setups, the area of evaluation, the localization devices, the strategy to collect data, or they are tailored to specific environments [28]. A few studies address the computational effort [8, 27]. Similar issues prevail here, the methodology for evaluating the computational complexity and the used metrics are so different that these methods are not comparable either. Having a comprehensive evaluation framework will enable the research community to get general results and allow for direct results comparisons.

This paper aims to fill this gap and introduces a comparison, through experimental evaluation, of existing clustering and optimization rules for Wi-Fi fingerprinting based on k-NN . Moreover, it also provides the tools to reproduce and extend this work, which might be useful for the research community when evaluating a new method. The major contributions of this paper are:

- Identification of the best strategy (clustering or optimization rule) depending on the scenario features.
- Identification of the best general methods to reduce computational costs in fingerprinting.
- Independent evaluation of surveyed methods in terms of positioning error *and* execution time.
- Evaluation procedure to normalize results and extract general conclusions from different perspectives.
- Supplementary materials for research reproducibility and to allow this work extension.

2 Background and Related Work

In fingerprinting (FP) positioning systems, the position estimate is frequently computed from a fingerprint representing the RSS from wireless signals, such as Wi-Fi or Bluetooth Low Energy (BLE), at unknown locations and a reference set with previously collected fingerprints. FP methods typically have two well differentiated phases: the *off-line phase* (also known as training or learning phase) and the *on-line phase* (also known as operational or test phase).

During the off-line phase, some known locations are selected for system calibration. In each of those locations, also known as Reference Points (RPs), multiple fingerprints are usually collected to capture the inherent temporal diversity of radio signals due to reflections, refraction, diffraction, scattering and interference. However, there is no standard procedure to carry out the off-line phase which depends on the deployment, the developer's strategies and other characteristics of the environment. The data-collection strategy includes the spatial distribution of RPs, the number of repeated measurements (fingerprints) per RP, the height for the sensing device [26], the user orientations [7], the devices and users [29, 30], and also the collection times [29, 31]. A common notion, though, is that the more dense the dataset, the lower the positioning error [32, 33]. However, this is not always the case, as shown by comparing the density values (Table 2) to the position errors (Table 4). For instance, TUT 6 provides the lowest error with low density values, and UJI 1 provides high error with higher density values.

During the on-line phase, one fingerprint collected at an unknown location is further processed using the selected positioning algorithm to compute a position estimate. A FP algorithm only relies on the radio map to estimate the position. The core idea behind FP is that a pair of similar fingerprints are physically close to each other and, therefore, similar samples in the radio map can be used to compute the location of the unknown fingerprint. Selecting the best model at this stage is a crucial step for FP.

Wi-Fi FP was introduced by Bahl *et al.* [7] in 2000. RADAR was presented as a system for locating and tracking users inside a building using only the RSS traces from a Wi-Fi network; and it had the above-mentioned two-phase procedure. In the off-line phase, multiple fingerprints were collected at each RP and user's orientation. In the on-line phase, the position was estimated using *multiple Nearest Neighbour* (k-NN). Twenty years later, Wi-Fi FP and methods based on k-NN are still very popular [19, 34, 35, 36].

There is an intrinsic trade-off when generating the radio map, because the accuracy of the FP-based positioning system increases typically with the density of the radio map,

i.e. the number of fingerprints per area, and the number of RSS values per fingerprint [37, 38]. However, generating the radio map is a demanding task. Some works have applied crowdsourcing [39, 40, 41, 42, 43], interpolation [37, 44, 45], signal-propagation models [34, 46] or Simultaneous Localization and Mapping (SLAM) [47, 48] to reduce data granularity and manual site surveying. However, the methods that augment the radio map to reduce the positioning error (e.g. applying Universal Krigging to generate a denser radio map as in [37]) also increase the computational costs in the on-line phase as they depend on the radio map size [49].

Although reducing computational costs is not identified as the main objective in previous works, computing the distances to all the reference fingerprints for every location request might be too inefficient [50]. A literature review on FP in Scopus and Web of Science raised many attempts to improve the accuracy of FP methods and reduce their computational load, being the relevant reproducible works implemented in this paper. Some of them apply optimization rules to reduce the radio map in the operational stage, while others apply clustering to group similar fingerprints.

The Horus indoor positioning system [51] addressed this problem in 2003, where a multi-level clustering process was proposed to estimate the position by means of a probabilistic approach. Only the detected APs in the on-line phase were used for computation and the search is restricted to the RPs covered by the strongest AP – the AP with the largest detected RSS value in the operational fingerprint.

Kushki *et al.* [8] proposed a kernel-based FP system where spatial filtering was done in the on-line phase. The idea behind the filtering is that the Wi-Fi coverage is similar in adjacent locations. First, a coarse estimation is done to get the RPs which have similar coverage (as the number of common APs) as the on-line fingerprint. Then, the reduced radio map only contains the fingerprints of these RPs.

Gallagher *et al.* [50] investigated a simple approach for radio-map reduction before computing the fingerprint distances [52, 53]. It keeps only the RPs which contained the strongest AP of the operational fingerprint. Gallagher et al. proposed some variants where the number of matching APs from the operational fingerprint was higher –without clearly specifying which APs were added– and by filtering by similar RSS values (± 15 or ± 20 dB).

Shin *et al.* [22] applied k-Means clustering (renamed in this paper as c-Means to avoid confusion k-NN) to extract and organise spaces from the radio map. However, c-Means has also been used for coordinate-based clustering [54], floor-wise fingerprint clustering [55] and, even, to cluster the positions of the nearest neighbors obtained with k-NN [56].

Marques *et al.* [26] filtered the radio map based on the notion that fingerprints are dominated by just one or two APs. For an on-line fingerprint, the reduced radio map only contains the reference samples whose strongest AP matches the strongest AP of the operational fingerprint, or the two strongest operational APs, if their RSSs values are close.

Chen *et al.* [24] and Caso *et al.* [57] decomposed the radio map in multiple clusters using the *Affinity Propagation* algorithm and applied the traditional two step, coarse and fine, location strategy. The novelty in [24] was to select the samples from multiple clusters to avoid the edge problem, whereas the novelty in [57] was the adoption of different metrics in different steps of the estimation procedure.

Razavi *et al.* [55] proposed a floor estimation method based on c-Means clustering. Their method clusters the data of each floor separately, thus relies on a preliminary search of the database's fingerprints according to their floor label. Subsequently, the mean of the clusters of each floor is computed. The reduction of data and communication overhead is achieved because only the cluster heads of each floor are used to estimate the floor.

Yu *et al.* [27] proposed to filter the radio map on the fly during the on-line stage. First, the non-detected APs in the on-line fingerprint are removed from the radio map. Then, the fingerprints without any detected RSS values are also removed. Finally, the filtered radio map, which contained the reference fingerprints in the same region where the unknown one was collected, is used for the localization.

Moreira *et al.* [23] proposed some rules to create subsets of the radio map. To estimate the building, the APs from the on-line fingeprint were sorted from the strongest to the weakest. The reference samples whose strongest AP matched the first AP of the sorted list were selected for the reduced radio map. If the reduced radio map was empty, they moved to the next AP in the list. This was repeated until reaching a valid radio map. For the floor estimation, they restrict the radio map to those reference fingerprints where the strongest AP corresponds to the first, second or third strongest AP of the on-line fingerprint.

Chen *et al.* [25] reduce the radio map by selecting the fingerprint with the strongest RSS for an AP as a cluster center, thus having as many clusters as APs, in an scenario with eight APs. In the operational stage, they used weighted k-NN upon the most similar cluster selection. No guides were given for contexts with larger amounts of APs, let alone for cases where there are more APs than samples.

Liu *et al.* [58] explored location-based clustering, where clusters are determined using the minimal radius circles that enclose all RPs. The circles are then used to cluster the RP by their distance to the circles' centers. In the operational stage, they used k-NN or a variant of weighted k-NN upon the most similar cluster selection.

The difficulty in comparing the performance of the methods introduced above is that they have not been evaluated using a common comprehensive evaluation setup. Therefore, from the existing literature, it is impossible to compare their relative merit no matter the considered perspective.

3 Material and Methods

Table 1 introduces our notations used further to compare and analyse different radio-map FP methods.

3.1 Fingerprinting with reduced radio maps

The approaches to reduce the computational costs can be roughly divided into clustering and optimization rules, where the later commonly refers to some kind of thresholding to decide if a fingerprint deemed relevant and therefore is considered or not. Algorithm 1 shows the processing stages of FP with k-NN over a reduced radio map.

The clustering and optimization rules trade-off on-line computation for mostly off-line computation, but as well

TABLE 1
Symbols and notation used in the paper.

$\mathcal{T}$	radio map, set of fingerprints and associated reference positions
$\hat{\mathcal{T}}$	reduced radio map after clustering or filtering, $\hat{\mathcal{T}} \subseteq \mathcal{T}$
$\mathcal{V}$	Evaluation dataset, set of labelled fingerprints for testing
$\mathcal{P}$	set of reference positions/labels, $\mathbf{p} \in \mathcal{P}$
$\lvert\cdot\rvert$	cardinality (e.g., $\lvert\mathcal{C}\rvert$ number of clusters, $\lvert\mathcal{C}_i\rvert$ number of samples in i-th cluster)
$\mathcal{C}$	set of clusters $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{\lvert C\rvert}\}$
$\mathcal{C}_i$	i-th cluster, $\mathcal{C}_i \subseteq \mathcal{T}$ or $\mathcal{C}_i \subseteq \mathcal{P}$
$\mathcal{A}$	set of access points, $\text{AP}_\gamma \in \mathcal{A}$
δ	average number of fingerprints per m^2 in a 5 m radius circle
$\mathbf{s}^t$	fingerprint of radio map or reduced radio map
$\mathbf{s}^v$	evaluation fingerprint
s	RSS value
γ	identifier associating s_γ to the AP_γ emitting the signal
f	floor
b	building
k	number of neighbours of NN
τ_{DB}	Execution time of the evaluation set (all operational samples)
$\tilde{\tau}_{\text{DB}}$	Execution time of the evaluation set, normalized to baseline
ϵ_{3D}	Positioning error of the evaluation set (all operational samples)
$\tilde{\epsilon}_{\text{3D}}$	Positioning error of the evaluation set, normalized to baseline

Algorithm 1 Pseudocode of k-NN for positioning

1: **input** $|\mathcal{T}|$, $|\mathcal{V}|$, k, distance metric, RSS representation
2: Off-line pre-processing of training datasets
3: **for** $i = 1$ **to** $|\mathcal{V}|$ **do**
4: Generate reduced radio map, $\hat{\mathcal{T}}$, using $\mathcal{T}$ and $\mathbf{s}_i^v$
5: **for** $j = 1$ **to** $|\hat{\mathcal{T}}|$ **do**
6: Compute distance between $\mathbf{s}_i^v$ and $\mathbf{s}_j^t$
7: **end for**
8: Sort distances in RSS space
9: Select the k closest candidates
10: Estimate building, floor and position
11: **end for**
12: **return** Estimated positions, floors, buildings

for some additional on-line computation. Both approaches pre-process the training datasets in the off-line phase. That is, training data (i.e., the radio map) is clustered or certain statistics of the training data – frequently thresholds for subsequent filtering – are computed (cf. line 2 in Alg. 1). During the on-line phase, the clustering methods perform a coarse search over the found clusters, to find the closest cluster (cf. line 4 in Alg. 1) and then match the fingerprints of that cluster with the operational fingerprint (cf. line 6 in Alg. 1). The optimization rules compute the respective statistic of the operational fingerprint, filter this fingerprint according to the computed statistic, intersect the filtered operational fingerprint and the filtered training dataset to obtain the reduced training data (cf. line 4 in Alg. 1) and then match the reduced dataset with the operational fingerprint (cf. line 6 in Alg. 1). The estimation of the position equals that of the baseline algorithm without radio map reduction and is common for clustering and optimization rules (cf. lines 8 to 10).

3.2 Methods implemented

For the experimental evaluation described here, some of clustering and optimization-rules introduced in Section 2 were implemented, their code is available in [69]. The ones with lack of key implementation details, i.e. not following reproducible research principles, were discarded.

3.2.1 Clustering methods

The method based on Kushki *et al.* [8] is **Kushki$x$**. In **Kushki$x$**, x refers to the threshold value to filter the RPs. We have considered threshold values in the range of $[1 \dots 15]$.

The methods based on c-Means clustering [70] are ***c*MeansTrad** and ***c*MeansAlt**. For the two methods, c refers to number of intended clusters. The particular values that correspond to $c = \sqrt{|\mathcal{T}|}$ and $c = \frac{|\mathcal{T}|}{25}$ are identified by 'rfp1' and 'rfp2', respectively. ***c*MeansTrad** corresponds to the traditional method used in several indoor positioning works [22, 55]. ***c*MeansAlt** variant uses the Manhattan distance and the centroid initialization proposed in [71].

The methods based on Affinity Propagation clustering [72] are **APCSpaTrad** and **APCSpaAlt**. **APCSpaTrad** uses the sparse implementation provided in [73]. **APCSpaTrad** computes the pairwise similarities for distances among all fingerprints. The alternative version considers the pairwise similarities as the distances among all fingerprints that have at least one AP in common, to reduce the memory and computational cost of the clustering stage.

The methods based on grid-based clustering [74] using fix-sized square cells are **Grid$x$** and **GridOverl$x$**. The grid-based clustering was suggested for RSS-based clustering by Liu *et al.* [58]. For the two methods, x refers to the size of the cell in meters. **GridOverl$x$** adds new cells that uniformly overlap each original set of four neighboring cells.

3.2.2 Optimization rules

The method inspired by Gallagher *et al.* [50] and Machaj *et al.* [75] is **Prcntil$x$**. The operational RSSs values are ranked from the strongest to the weakest. The strongest APs falling in the x percentile of that rank are used to find all the fingerprints in the radio map which contain these APs in the x percentile of the corresponding ranks.

The method proposed by Marques *et al.* [26] is **Marques10**. It uses the 1st and 2nd strongest APs of the on-line fingerprint if their difference is 10 dBm or lower.

The methods based on Yu *et al.* [27] are **FengYu**, **FengYuOpt** and **FengYuOpt$x$%**. **FengYu** follows the original method. **FengYuOpt** additionally pre-computes the reduced radio map for each AP off-line. **FengYuOpt$x$%** modifies **FengYuOpt** by considering only fingerprints from the radio map that have at least x % of the total number of APs detected in the on-line fingerprint, similar to [50]. **FengYuOpt$x$%** does not apply its modification if the reduced radio map is empty.

The methods based on Moreira *et al.* [23] are **Moreira1st**, **Moreira3st**, **MoreiraS06** and **MoreiraS12**. **Moreira1st** applies the basic filtering described for building estimation. **Moreira3st** applies the filtering described for the variant 1 of floor estimation. **MoreiraS06** and **MoreiraS12** apply the filtering described for the variant 2 of floor estimation using 6 or 12 dB as maximum RSS difference respectively.

TABLE 2
Features of the selected databases. Fingerprint density is computed as the number of fingerprints per reference point (δ_{fp}). Local density is computed as the average number of fingerprints per m^2 in a circle with a radius of 5 m from each reference and evaluation point ($\delta^{\mathcal{T}}$ and $\delta^{\mathcal{V}}$). The scenario size is represented by its dimensions or approximate area (Dimension/Area), number of floors (#f), and number of buildings (#b). The average number of APs detected in the fingerprints is shown (Valid APs). The number of devices used to collect the dataset is also shown (Dev.).

DB	$\lvert\mathcal{T}\rvert$	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{P}\rvert$	δ_{fp}	Dimension/Area	#f	#b	$\delta^{\mathcal{V}}$	$\delta^{\mathcal{T}}$	Valid APs	Dev.	Ref
DSI 1	1369	348	157	230	6	100 m×18 m	1	1	0.70 ± 0.27	0.73 ± 0.27	23.6 ± 7.9	1	[59]
DSI 2	576	348	157	230	2 to 3	100 m×18 m	1	1	0.29 ± 0.12	0.31 ± 0.11	23.6 ± 7.9	1	[59]
LIB 1	576	3120	174	48	12	15 m×10 m	2	1	2.41 ± 0.70	2.42 ± 0.59	21.0 ± 6.3	1	[31]
LIB 2	576	3120	197	48	12	15 m×10 m	2	1	2.41 ± 0.70	2.42 ± 0.59	18.8 ± 5.1	1	[31]
MAN 1	14300	460	28	130	110	50 m×36 m	1	1	20.55 ± 5.12	20.88 ± 4.48	10.5 ± 2.4	1	[60, 61]
MAN 2	1300	460	28	130	10	50 m×36 m	1	1	1.87 ± 0.47	1.90 ± 0.41	14.1 ± 2.7	1	[60, 61]
SIM	10710	1000	8	1071	10	50 m×20 m	1	1	8.86 ± 1.50	8.86 ± 1.80	8 ± 0	1	[62]
TUT 1	1476	490	309	1476	1	124 m×57 m	4	1	0.41 ± 0.10	0.33 ± 0.13	25.0 ± 7.3	1	[63]
TUT 2	584	176	354	584	1	145 m×88 m	3	1	0.12 ± 0.05	0.11 ± 0.05	21.9 ± 7.1	1	[63]
TUT 3	697	3951	992	694	1	130 m×62 m	5	1	0.16 ± 0.08	0.17 ± 0.07	49.7 ± 38.7	21	[64]
TUT 4	3951	697	992	3843	1	130 m×62 m	5	1	0.91 ± 0.48	0.96 ± 0.50	48 ± 38.4	21	[64]
TUT 5	446	982	489	446	1	85 m×145 m	3	1	0.07 ± 0.02	0.08 ± 0.03	34.8 ± 13.5	1	[65]
TUT 6	3116	7269	652	3116	1	135 m×62 m	4	1	0.63 ± 0.31	0.65 ± 0.31	34.7 ± 15.9	1	[66]
TUT 7	2787	6504	801	2787	1	88 m×137 m	3	1	0.47 ± 0.29	0.48 ± 0.30	27.1 ± 11.1	1	[66]
UJI 1	19861	1111	520	933	20	108.703 m^2 total	4 to 5	3	2.45 ± 1.62	2.46 ± 1.84	16.5 ± 6.9	25	[29]
UJI 2	20972	5179	520	1968	1 or 20	108.703 m^2 total	4 to 5	3	2.42 ± 1.59	2.63 ± 1.76	16.2 ± 4.8	30	[67, 68]

3.3 Description of data sets

The experiments carried out in this paper include 16 datasets (see Table 2) from 12 different data sources. They have been selected for the experiments because they have diverse characteristics: small, medium and large scenarios; single-floor and multiple-building scenarios; unprocessed RSS data and averaged RSS data per reference point or grid cell; single device collection and device diversity; systematic and crowdsourced data collection; and spatially-sparse but dense radio maps. Moreover, two datasets were collected in the same place following the same data collection strategy with a time interval of 10 months. The strategies used to collect the datasets are summarized in Table 3.

TABLE 3
Strategy of data collection including if it was collected in a *Systematic* way, the actor(s) who collected the data and RSS post-processing.

DB	Systematic	Actor	RSS post-process
DSI 1/2	✓	Professional	✗
LIB 1/2	✓	Professional	✗
MAN 1	✓	Professional	✗
MAN 2	✓	Professional	RP Average
SIM	✓	Path-loss	✗
TUT 1/2	✓	Professional	Cell Average (Grid size: 1 m)
TUT 3/4	✗	Volunteers	✗
TUT 5	✓	Professional	Cell Average (Grid size: 5 m)
TUT 6/7	✓	Professional	✗
UJI 1/2	✓	Prof. & Vol.	✗

Dataset DSI 1 was collected at the Department of Information Systems of the Universty of Minho (Portugal). The dataset DSI 2 is a version of DSI 1 where the repeated instances of the same fingerpring – same RSS values in the same RP – have been removed. Dataset LIB 1 was collected on two floors of Universitat Jaume I Library (Spain) in June 2016, whereas LIB 2 was collected in the same conditions in April 2017. MAN 1 is a dataset that covers the corridors of the second floor of an office building on the campus of the University of Mannheim (Germany), the evaluation set has been reduced by randomly picking 10 fingerprints per evaluation point with respect to the original dataset [60]. In the MAN 2 dataset, the fingerprints from the original dataset have been averaged in 10 blocks of 10 fingerprints for the training and evaluation sets to have one dataset with averaged RSS and fingerprints. We include an artificial dataset, SIM, based on simple the path-loss model with additive Gaussian noise (eq.1) as done in [62, 76, 77, 78].

$$s_p = s_0 - (\alpha \cdot 10 \cdot \log_{10}(\frac{d}{d_0})) + X(t) \tag{1}$$

where $s_0 = -40\,\text{dB}$, $\alpha = 2$, d refers to the distance to the AP ap, $d_0 = 1$ and $X(t)$ corresponds to the noise modelled as a Gaussian random process with null mean and $\sigma = 2\,\text{dB}$ (as it is usually between 2 and 3 [54, 76, 77, 79]).

Datasets TUT 1, TUT 3 and TUT 6 were all independently collected in a five-floor building at Tampere University (Finland), but different actors and data collection strategies were used (see Table 3). TUT 4 is identical to TUT 3, but we used the training points as the test points and vice versa. TUT 3 and TUT 4 were collected by crowdsourcing means. TUT 2, TUT 5 and TUT 7 were all independently collected in a three-floor building of Tampere University using different actors and data collection strategies (see Table 3). UJI 1 was collected on three multi-storey buildings of the School of Technology in Universitat Jaume I. UJI 2 contains all UJI 1 data as training set and a new blind test data set collected in a 12-month interval for the IPIN 2015 competition [68].

Although the selected datasets mainly come from four different research teams, the data collection strategy, data structure and the data formats are diverse, cf. [29, 60, 63]. Due to the different dataset sources and formats, we had to normalize the datasets and apply a common simple format.

The suggested common data format includes training and evaluation data in separated structures, where the inputs (RSS values) and outputs (positions) are also separated. The input values (RSS) for the training data are stored in a $\lvert\mathcal{T}\rvert \times \lvert\mathcal{A}\rvert$ matrix, where the non-detected APs are expressed with the value $+100$. The output values (position and labels) for the training data are expressed in a $\lvert\mathcal{T}\rvert \times 5$ matrix,

where each row contains: the x, y-coordinates (in meters, either in a local or global reference system), the height, the floor identifier and the building identifier. Similarly, the evaluation data include a $|\mathcal{V}| \times |\mathcal{A}|$ matrix with the inputs and a $|\mathcal{V}| \times 5$ matrix with the ground truth.

For single-floor and/or single-building datasets, the floor and building identifiers were set to 0. For the datasets where the floor height was undefined, it has been calculated as the product of the integer floor identifier and a default height value (3.7 m). The supplementary materials include a copy of the datasets or the scripts to generate them [69].

4 Theoretical Analysis and Experiments

One of the objectives of this work is to compare the computation burden associated to each one of the evaluated methods. This section approaches such comparison from a theoretical, when possible, and experimental points of view.

4.1 Time complexity of fingerprinting on-line stage

Before we present the experimental results, we provide an analysis of the asymptotic time complexities of the FP on-line stage, including the clustering and optimization rules. The on-line stage consists of three main processes: i) reduction of the radio map, ii) matching of the training and operational fingerprints and iii) the computation of the position, floor, and building (cf. Alg. 1). For the analysis, we exclude the position estimation stage from the analysis, because it is common for all methods based on k-NN.

4.1.1 Reduction of the radio map

The reduction of the training dataset differs for clustering methods and for optimization rules. Furthermore, both the clustering and optimization rules process either RPs, or RSSs or simply the APs identifiers.

Clustering-based methods obtain the reduced training dataset by finding the cluster that is closest to the operational fingerprint. To that end, the minimum distance between cluster heads and the operational fingerprint is computed. This operation has linear complexity $\mathcal{O}(c)$, assuming a simple distance measure is used and where $c = |\mathcal{C}|$ is the number of clusters.

In the optimization rules, the RSSs of the operational fingerprints are processed and intersected with the training dataset to obtain the reduced dataset. This operation is linear $\mathcal{O}(m)$ for many methods too, where m is the number of elements in the training dataset ($m = |\mathcal{T}|$) or the set of reference positions ($m = |\mathcal{P}|$). However, some optimization rules use the strongest(s) APs to filter the radio map or sort the APs to compute the quantiles on the operational fingerprint. In these cases the worst case complexity of reducing the training database is at best $\mathcal{O}(m \log(m))$, with $m = |\mathcal{A}|$, in common implementations[1]. The methods FengYu (and variants), Prcntil$k$, Marques10, Moreira1st and Moreira3st employ algorithms with that asymptotic complexity.

In worst case, the number of clusters may reach theoretically the number of training fingerprints, $c \leq |\mathcal{T}|$, or RPs, $c \leq |\mathcal{P}|$, depending on the clustered quantity. Also the the optimization rules may process all RSSs or labels, so that $m \leq |\mathcal{A}|$. Unreasonable parameter choices or degenerative distributions of RSSs or RPs may cause such situation. In practise however, these are rather unlikely situations as the number of clusters is usually much smaller than the number of reference fingerprints. For instance, the number of clusters was set to small fractions ($\rho = 0.01, 0.05, 0.1$) with respect to the reference fingerprints [55]. Similarly, the number of processed APs in optimization rules is much lower than the set of APs identified in the dataset.

Figure 2 supports the notion of a moderate average computational complexity also for the methods that require sorting to reduce the training set. It exemplifies the histogram of the number of valid APs per reference fingerprint using all datasets. The histogram shows that two thirds of all reference fingerprints considered in this work contain less than 30 valid APs. That is, the number of elements to sort is indeed much smaller than the total number of APs, and therefore, it has usually has not a large computational cost attached and is not critical. However, the number of detected APs in an operational fingerprint is variable and also depends on the dataset, the location of the operational sample, and the device used to collect the fingerprint, as reported in Table 2.

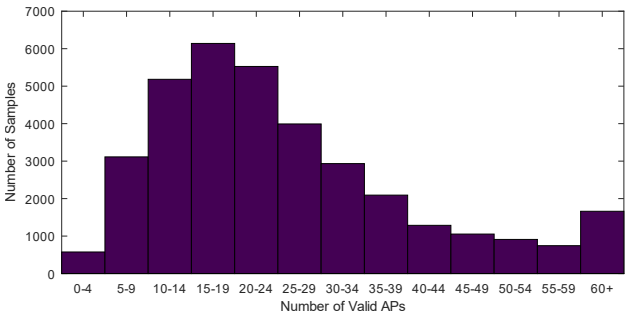

Fig. 2. Histogram of the number of valid APs per reference fingerprint considering all fingerprints from the sixteen reference datasets.

4.1.2 Matching of the training and operational fingerprints

Once the radio map is reduced, the worst-case time complexity of the fingerprint matching is determined by the choice of the distance measure. The majority of distance measures (see [80, 81]) have in fact the same worst-case time complexity, namely linear complexity $\mathcal{O}(n)$, where $n = |\hat{\mathcal{T}}|$ is the number of fingerprints of the reduced radio map.

The reduced radiomap depends on the clustering and optimization rule. Theoretically, there exist scenarios for all clustering and optimization rules in which the training dataset will not be reduced, so that the maximal value of fingerprints is $n = |\mathcal{T}|$. Not only unreasonable parameter choices may lead to that case, a degenerative distribution of fingerprints or APs may induce such a situation too. For example, if the environment is small and only one dominant AP is observed (the methods and variants proposed by Moreira *et al.* [23], Marques *et al.* [26], and Yu *et al.* [27] are vulnerable to this), if the RPs are all located in a small part of the environment and end up in a single grid, or if the number of fingerprints per RP is one for the method

1. Octave implements 'Timsort' which has a worst-case complexity of $\mathcal{O}(n \log(n))$. Octave's computation of the median exploits `nt_element` and `partial_sort` of the standard template library of C++, both also with asymptotic complexity of $\mathcal{O}(n \log(n))$.

proposed by Kushki *et al.* [8]. However, the inclusion of optimization rules and clustering methods in commercial [82] and competing systems [23] somehow indicate that they efficiently reduce the computational costs when they are properly configured for a particular area.

The computational cost for the clustering methods can be simplified with $f(\mathcal{C}) = |\mathcal{C}| + \frac{|\mathcal{T}|}{|\mathcal{C}|}, \forall |\mathcal{C}| \in \mathbb{N} : |\mathcal{C}| \leq |\mathcal{T}|$, which shows the vector comparisons required for the coarse ($|\mathcal{C}|$ vector distance calculations) and fine-grained ($\frac{|\mathcal{T}|}{|\mathcal{C}|}$ vector distance calculations) searches if the fingerprints are equally distributed in the clusters. The global minimum of the cost function is located at $|\mathcal{C}| = \sqrt{|\mathcal{T}|}$. Therefore, the best scenario is the one where the number of clusters is the square root of the number of the training samples. If the reference fingerprints are equally distributed into disjoint clusters, the coarse and fine grained searched are both $\mathcal{O}(|\mathcal{T}|^{\frac{1}{2}})$. The computational costs of both searches would be balanced and optimized, providing the lowest joint computational load since $\mathcal{O}(|\mathcal{T}|^{\frac{1}{2}}) << \mathcal{O}(|\mathcal{T}|)$. This is intuitive, because having a small number of clusters would end in large clusters, whereas a large number of clusters would end in very small clusters. The extreme cases, c equal to 1 or $|\mathcal{T}|$, would end with the fine-grained or coarse search of $\mathcal{O}(|\mathcal{T}|)$ respectively. The number of clusters can only be set in c-Means clustering, as it is automatically determined in Affinity Propagation and it corresponds to the number of reference points/cells in Kushki and Grid-based clustering. However, none of the analyzed methods can ensure that the generated clusters are balanced, so the size of the reduced radiomap depends on the operational fingerprint.

In contrast, the optimization rules apply some knowledge-based rules to decide whether a reference fingerprint is included in the reduced radio map or discarded. The rules based on the strongest(s) APs, for instance, keep those fingerprints that are near the dominant APs. Thus, the area covered in the reduced radio map not only depends on the location of the operational and reference fingerprints but also on the the APs distribution. Again, the reduced radio map size is tightly coupled to the operational fingerprint.

The worst-case complexity analysis helps to understand the trade-offs to be made, but it does not provide a realistic comparison between the considered methods. A theoretical assessment of the average computational complexity would provide a more accurate picture and possibly a guideline to alleviate the computation burden of FP methods. However, such an assessment depends on multiple dimensions, including the distributions of RSSs, RPs and/or APs of the datasets and, finally, the parameter choices for the clustering and optimization rule(s). This analysis is not feasible in this paper due to number of analyzed alternatives to reduce the radio map and the inner diversity of the considered indoor environments (i.e, the 16 datasets). Instead, for the remaining of this study, we opted for an experimental approach and we assess the average computational complexity through measured execution times, including as well the effect on the positioning accuracy. We encourage that further fingerprint models based on optimization rules or clustering include a theoretical assessment as done, for instance, in [8].

4.2 Empirical experiments

The purpose of this analysis is to explore the weaknesses and strengths of the selected methods to reduce the complexity of the radio map and identify hidden general patterns with respect to the datasets. All the experiments were executed on a cluster based on Intel Xeon E5-2670 processors, with 128 GB of RAM and GNU Octave 3.8.2. The results have been confirmed using distinct hardware and software.

4.2.1 Evaluation Framework

According to [81], the three main parameters of this model are: the data representation for the RSS values, the distance metric and the value of k. We selected two parameter configurations to test how the clustering models perform on different datasets with different parametrization of k-NN: i) the *Simple Configuration* with 1-NN, Manhattan distance (also known as City Block or L1 distance) and positive data representation; and the *Best Configuration*, which resulted from exploring multiple combinations of the main parameters. For setting the Best Configuration, we have considered three data representations, namely *positive*, *exponential* and *powed* [81], eight distance metrics, namely *Euclidean*, *Manhattan*, *Euclidean*^{2}, *Neyman*, *Sørensen*, *LGD*, *PLGD10*, *PLGD40* [80, 81, 83], and $k = \{1, 3, 5, 7, 9, 11\}$. We apply this evaluation setup on the 16 datasets introduced in Section 3.3. The results on plain k-NN are shown in Table 4 for each dataset.

Table 4 provides the absolute error ϵ_{3D} and cost τ_{DB} for all the datasets, but it also includes the normalized values ($\tilde{\epsilon}_{3D}$ and $\tilde{\tau}_{DB}$). The Simple Configuration on the plain k-NN has been used as the baseline for normalization, so that the normalized results showed on this paper are all relative to it. The last row shows the average over the 16 datasets for extracting general conclusions and further comparisons.

The table also shows three relevant outputs. First, the optimal value of k seems to depend on the local density ($\delta^{\mathcal{T}}$ in Table 2). For an operational fingerprint, the number of relevant very similar reference fingerprints highly depends on the fingerprint density of the radio map as already suggested in [84]. Therefore, the value of k should be set accordingly (e.g. $k = 1$ for datasets with low $\delta^{\mathcal{T}}$). Second, selecting the best performing configuration can have a significant impact in both the accuracy and the computational costs. The normalized positioning accuracy, $\tilde{\epsilon}_{3D}$, is halved for some datasets and the normalized computational times, $\tilde{\tau}_{DB}$, drop about 5 %, increase about 15 % and reaches 3 times the value of the normalized computational times for data sets where the *Sørensen*-, the *Euclidean*^{2}- and Probabilistic Log-Gaussian (namely *PLGD10* and *PLGD40*) [83] distances were used, respectively. That happened under different configurations of k and data representation, which seems to indicate that the impact of the distance metric is constant. i.e., the computational costs are independent to k and the data representation, as clearly shown in database MAN 1. Third, the normalized aggregated metrics provided in the last row show that, in general, selecting the Best Configuration improves the accuracy by 26 % at the expense of increasing the computational burden by 76 %. This computational increase is mainly caused by the six cases where a PLGD distance metric was selected, and it could have been avoided by selecting an alternative configuration with similar accuracy but lower costs (e.g. *Sørensen*). PLGD includes

TABLE 4
Comparison of positioning error and computation time for Simple and Best parameter configurations using plain k-NN for each dataset.

	Simple Conf.				Best Conf.						
Database	ϵ_{3D}	τ_{DB}	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}_{DB}$	data rep.	distance	k	ϵ_{3D}	τ_{DB}	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}_{DB}$
DSI 1	4.95	31.59	1	1	pow	Sørensen	11	3.79	36.32	0.77	1.15
DSI 2	4.95	13.46	1	1	pos	PLGD10	9	3.80	39.99	0.77	2.97
LIB 1	3.02	107.43	1	1	pos	Euclidean^2	11	2.48	102.33	0.82	0.95
LIB 2	4.18	108.22	1	1	pos	PLGD10	9	2.27	322.48	0.54	2.98
MAN 1	2.82	353.68	1	1	exp	Manhattan	11	2.06	353.64	0.73	1
MAN 2	2.47	32.14	1	1	exp	Neyman	11	1.86	50.12	0.75	1.56
SIM	3.24	567.16	1	1	exp	Euclidean^2	11	2.41	532.66	0.74	0.94
TUT 1	9.59	50.38	1	1	pos	PLGD40	3	4.45	152.76	0.46	3.03
TUT 2	14.37	7.34	1	1	pow	Sørensen	1	8.09	8.47	0.56	1.15
TUT 3	9.59	208.88	1	1	pos	Sørensen	3	8.55	239.06	0.89	1.14
TUT 4	6.36	218.57	1	1	pos	PLGD10	3	5.40	705.29	0.85	3.23
TUT 5	6.92	29.20	1	1	pos	PLGD40	3	5.26	91.27	0.76	3.13
TUT 6	1.94	1617.56	1	1	pos	Sørensen	1	1.91	1850.11	0.98	1.14
TUT 7	2.69	1287.97	1	1	pos	Sørensen	1	2.24	1541.50	0.83	1.2
UJI 1	10.81	1766.85	1	1	pow	Sørensen	11	6.56	2019.87	0.61	1.14
UJI 2	8.05	8686.48	1	1	exp	Neyman	11	6.09	12 410.65	0.76	1.43
average			1.0	1.0						0.74	1.76

complex penalty terms and exponential operations [83], whereas *Sørensen* just adds a dynamic normalization term to Manhattan distance [80]. Despite DSI 2 is the reduced version of DSI 1, the computational cost of DSI 2 is much higher than DSI 1 in the best configuration.

4.2.2 Dataset-wise Analysis

In order to analyse and present the results over the multitude of methods and datasets, we introduce a visualization that allows to depict the four relevant dimensions on once: the datasets, the method to reduce the computational costs, the positioning accuracy and the computation time. This visualization shows the normalized aggregated metrics for each combination of dataset and method, as colored ellipses. The color indicates $\tilde{\tau}_{DB}$ compared to the baseline, where dark green stands for 0, white for 1 and the darker the red the higher the computation time. The shape stands for the $\tilde{\epsilon}_{3D}$, a horizontal ellipse represents values closer to 0, a circle identifies an error of 1 and a vertical ellipse stands for an increased error, compared to the baseline. The methods based on grid clustering, Kushki and c-Means, depend on an additional parameter; which are set for each dataset according to the best error (BE) and best time (BT) as shown in the example Figure 3. This reduces the reported methods and enhances the clarity of the full results shown in Figure 4.

In the SIM dataset, the eight APs are detected in all the reference samples. An AP from the datasets MAN 1 and MAN 2 was detected in most of the evaluation area. A

Extract of the normalized results for TUT 4

	Best Conf.	
Clustering	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}_{DB}$
5-MeansAlt	**0.88**	1.45
10-MeansAlt	0.90	0.90
15-MeansAlt	0.91	0.62
20-MeansAlt	0.91	0.50
25-MeansAlt	0.92	0.42
rfp1-MeansAlt	0.94	0.17
rfp2-MeansAlt	0.99	**0.09**

→

BE-cMeansAlt $\tilde{\epsilon}_{3D}$=0.88 $\tilde{\tau}_{DB}$=1.45

BT-cMeansAlt $\tilde{\epsilon}_{3D}$=0.99 $\tilde{\tau}_{DB}$=0.09

Fig. 3. Example of how full results are visually represented for the TUT 4 dataset and the best models (Best Error and Best Time) for cMeansAlt.

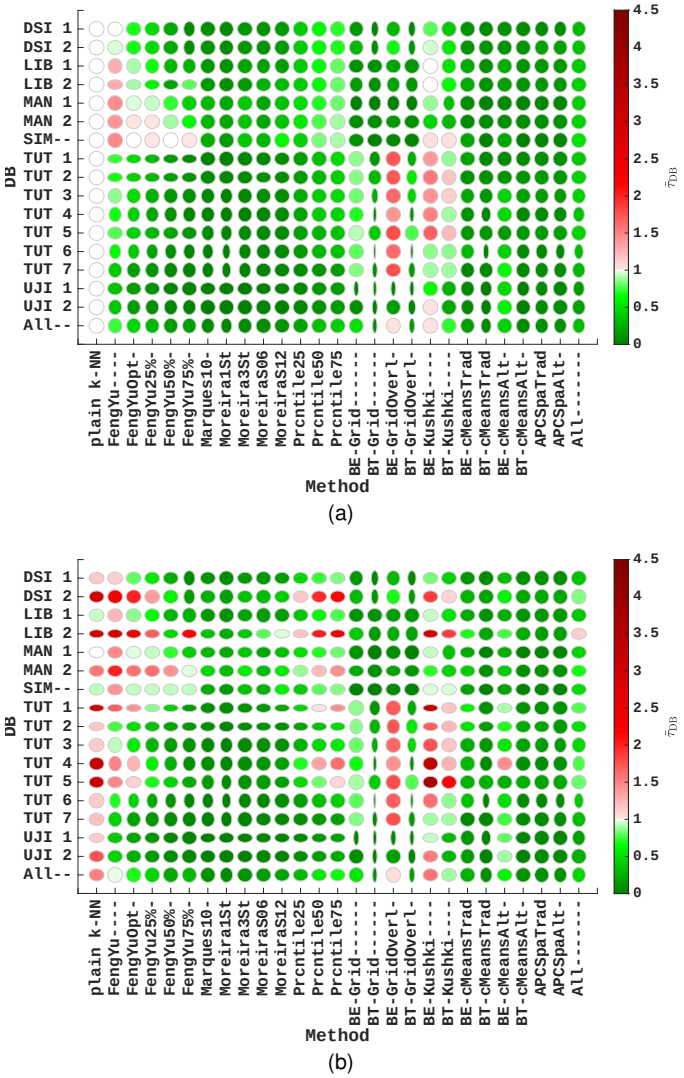

Fig. 4. Visualization of $\tilde{\epsilon}_{3D}$ and $\tilde{\tau}_{DB}$ for (a) Simple Configuration and (b) Best Configuration. The magnitude of vertical or horizontal stretching of the ellipses represents the $\tilde{\epsilon}_{3D}$ values above or under 1, respectively. The best models include the best result for each dataset.

similar behavior is found in datasets LIB 1 and LIB 2, where the evaluation environment had 8 APs which were detected in almost all the RP. Those cases include relative small areas with low attenuation where an important subset of APs are received over all the area and, therefore, the rule proposed by Yu *et al.* [27] does not reduce the computations.

For the Best Configuration, the visual results of the plain k-NN indicate that the alternative distance metrics are, in general, more time-consuming than the Manhattan distance (there are reddish ellipsoids for almost all datasets). The choice of notably expensive distance metrics also affected the computational burden for clustering methods (reddish ellipsoids on Feng Yu and Percentile rules).

BE-Kushki reports an accuracy similar to plain k-NN. However, Figure 4 shows that BE-Kushki is not suitable for TUT databases and UJI 2. They contain many reference points with just 1 fingerprint. In those cases, the number of clusters is so high that the coarse grained search has a computational load similar to the matching stage of the plain k-NN. Moreover, it is not suitable for the SIM dataset, in which all APs are detected in all fingerprints. Therefore, the coarse search never filters the radio map. However, it has the expected behavior in the other datasets, reducing the computational load while keeping the accuracy.

The rules proposed by Marques and Moreira gave good results in both dimensions for many cases. The cases include datasets LIB 2, TUT 1 and TUT 2 for the Best Conf., which reported very high computational cost in plain k-NN. For datasets TUT 6 and TUT 7, it seems that only Moreira3st, MoreiraS03 and MoreiraS06 are suitable. Their low positioning error in the baseline was really hard to improve.

Finally, the results reflects the heterogeneity of datasets. TUT 1 and TUT 2, where the samples lie in a regular 1 m grid, provided the worst accuracy for the baseline. The methods based on grid clustering report either very large error or very large computational time in most of cases. In contrast, TUT 6 provided the lowest accuracy for the baseline and further reductions on its computational cost usually result in large increases in the error. TUT 6 applied a strict systematic data collection without post-processing, in contrast to TUT 1, TUT 3 and TUT 4 that applied grid average or crowdsourcing in the same environment. Furthermore, methods based on grid-clustering never provide good positioning error for UJI 1, which has fingerprints from at least two devices at every reference point.

4.2.3 Method-wise analysis

Table 5 shows the aggregated normalized results for all the clustering methods and optimization rules described in Section 3.2. It contains the average normalized error and computational time over the sixteen datasets. The values in parenthesis show the standard deviation of the averaged values. These two metrics have been calculated as shown in the two last rows of Table 4 for the plain k-NN method. Most of the methods have more than one entry in the table since different parameters were tested, BE and BM denotes the best error and best time for each dataset considering different parameter values (see Figure 3).

As before, the baseline corresponds to the results of Simple Configuration in Table 4. Thus, the plain k-NN algorithm without any radio map reduction provides an

TABLE 5
Positioning error and computation time of all methods in two cases.

Method	Simple Conf. $\tilde{\epsilon}_{3D}$	Simple Conf. $\tilde{\tau}_{DB}$	Best Conf. $\tilde{\epsilon}_{3D}$	Best Conf. $\tilde{\tau}_{DB}$
plain k-NN	1.00 (0.00)	1.00 (0.00)	0.74 (0.14)	1.76 (0.93)
FengYu	0.95 (0.21)	0.90 (0.40)	0.86 (0.23)	1.30 (0.71)
FengYuOpt	0.95 (0.21)	0.58 (0.31)	0.86 (0.23)	0.98 (0.67)
FengYuOpt25%	0.95 (0.21)	0.44 (0.32)	0.86 (0.23)	0.70 (0.51)
FengYuOpt50%	0.95 (0.21)	0.29 (0.31)	0.86 (0.23)	0.41 (0.37)
FengYuOpt75%	1.02 (0.24)	0.24 (0.31) *	0.94 (0.28)	0.37 (0.57)
Prcntil25	0.97 (0.04)	0.27 (0.17)	0.74 (0.14)	0.46 (0.36)
Prcntil50	0.98 (0.03)	0.45 (0.25)	0.74 (0.14)	0.80 (0.59)
Prcntil75	0.98 (0.04)	0.54 (0.27)	0.74 (0.14)	0.97 (0.70)
Grid0050	1.45 (0.83)	0.39 (0.34)	1.52 (0.67)	0.40 (0.33)
Grid0100	1.44 (0.82)	0.36 (0.31)	1.51 (0.66)	0.38 (0.32)
Grid0250	1.40 (0.56)	0.27 (0.26)	1.48 (0.52)	0.27 (0.26)
Grid0500	1.73 (0.79)	0.19 (0.21)	1.87 (0.83)	0.21 (0.21)
Grid1000	2.47 (1.82)	0.13 (0.10)	2.85 (2.07)	0.16 (0.16)
BE-Grid	1.31 (0.55)	0.28 (0.25)	1.35 (0.51)	0.28 (0.23)
BT-Grid	2.40 (1.87)	0.10 (0.10)	2.53 (1.83)	0.11 (0.10)
GridOverl0050	1.45 (0.83)	0.69 (0.65)	1.52 (0.67)	0.69 (0.65)
GridOverl0100	1.41 (0.82)	0.64 (0.62)	1.48 (0.66)	0.64 (0.61)
GridOverl0250	1.42 (0.82)	0.47 (0.50)	1.43 (0.57)	0.47 (0.49)
GridOverl0500	1.64 (0.67)	0.34 (0.42)	1.76 (0.71)	0.35 (0.41)
GridOverl1000	2.16 (1.37)	0.20 (0.20)	2.56 (1.75)	0.22 (0.20)
BE-GridOverl	1.28 (0.54)	0.45 (0.44)	1.32 (0.51)	0.46 (0.44)
BT-GridOverl	2.11 (1.41)	0.18 (0.21)	2.24 (1.38)	0.19 (0.21)
Kushki001	1.02 (0.03)	1.18 (0.37)	0.77 (0.15)	1.83 (1.02)
Kushki002	1.02 (0.03)	1.13 (0.37)	0.78 (0.15)	1.76 (1.02)
Kushki003	1.04 (0.06)	1.08 (0.38)	0.80 (0.17)	1.68 (0.99)
Kushki004	1.04 (0.09)	1.04 (0.38)	0.81 (0.18)	1.61 (0.96)
Kushki005	1.04 (0.09)	1.00 (0.38)	0.82 (0.18)	1.54 (0.93)
Kushki006	1.04 (0.06)	0.96 (0.37)	0.82 (0.17)	1.47 (0.88)
Kushki007	1.03 (0.05)	0.94 (0.36)	0.81 (0.16)	1.41 (0.84)
Kushki008	1.03 (0.04)	0.91 (0.35)	0.81 (0.15)	1.35 (0.80)
Kushki009	1.02 (0.03)	0.90 (0.34)	0.80 (0.15)	1.31 (0.76)
Kushki010	1.02 (0.03)	0.88 (0.33)	0.80 (0.14)	1.26 (0.71)
Kushki011	1.02 (0.03)	0.86 (0.32)	0.79 (0.13)	1.22 (0.67)
Kushki012	1.03 (0.03)	0.84 (0.32)	0.80 (0.12)	1.17 (0.62)
Kushki013	1.03 (0.03)	0.82 (0.32)	0.81 (0.12)	1.11 (0.59)
Kushki014	1.02 (0.02)	0.80 (0.32)	0.81 (0.11)	1.07 (0.55)
Kushki015	1.02 (0.02)	0.79 (0.32)	0.81 (0.11)	1.04 (0.52)
KushkiBE	1.00 (0.01)	1.07 (0.35)	**0.74 (0.13)**	1.75 (1.00)
KushkiBT	1.05 (0.08)	0.71 (0.35)	0.84 (0.14)	0.93 (0.55)
5-MeansTrad	1.12 (0.14)	0.25 (0.06)	0.97 (0.27)	0.48 (0.24)
10-MeansTrad	1.12 (0.17)	0.15 (0.03) *	0.96 (0.25)	0.27 (0.14)
15-MeansTrad	1.12 (0.16)	0.11 (0.03) *	1.06 (0.45)	0.19 (0.10) *
20-MeansTrad	1.17 (0.41)	0.09 (0.03) *	1.18 (0.69)	0.15 (0.08) *
25-MeansTrad	1.15 (0.37)	0.09 (0.03) *	1.14 (0.68)	0.14 (0.07) *
rfp1-MeansTrad	1.22 (0.43)	**0.07 (0.04)** *	1.11 (0.50)	0.11 (0.08) *
rfp2-MeansTrad	1.18 (0.31)	0.08 (0.04) *	1.07 (0.35)	0.11 (0.08) *
BE-cMeansTrad	1.04 (0.06)	0.15 (0.08) *	0.86 (0.16)	0.31 (0.18)
BT-cMeansTrad	1.22 (0.43)	0.07 (0.04) *	1.06 (0.38)	0.10 (0.08) *
5-MeansAlt	1.05 (0.10)	0.35 (0.13)	0.80 (0.12)	0.65 (0.34)
10-MeansAlt	1.04 (0.04)	0.21 (0.08) *	0.83 (0.12)	0.39 (0.21)
15-MeansAlt	1.03 (0.03)	0.15 (0.05) *	0.85 (0.10)	0.27 (0.14)
20-MeansAlt	1.03 (0.03)	0.13 (0.04) *	0.86 (0.10)	0.22 (0.12) *
25-MeansAlt	1.03 (0.04)	0.11 (0.04) *	0.87 (0.09)	0.18 (0.10) *
rfp1-MeansAlt	1.05 (0.06)	0.08 (0.05) *	0.89 (0.09)	0.13 (0.09) *
rfp2-MeansAlt	1.06 (0.08)	0.09 (0.05) *	0.92 (0.09)	0.13 (0.09) *
BE-cMeansAlt	1.01 (0.03)	0.27 (0.19)	0.80 (0.12)	0.59 (0.33)
BT-cMeansAlt	1.05 (0.06)	0.08 (0.05) *	0.91 (0.09)	0.12 (0.09) *
APCSpaTrad	1.10 (0.08)	0.10 (0.05) *	0.98 (0.12)	0.11 (0.06) *
APCSpaAlt	1.13 (0.17)	0.10 (0.05) *	1.05 (0.19)	0.11 (0.06) *
Marques10	1.05 (0.16)	0.11 (0.10) *	0.89 (0.22)	0.15 (0.13) *
Moreira1st	1.15 (0.31)	0.07 (0.07) *	1.00 (0.31)	**0.10 (0.09)** *
Moreira3st	1.02 (0.12)	0.13 (0.13) *	0.83 (0.18)	0.19 (0.17) *
MoreiraS06	0.97 (0.11)	0.12 (0.09) *	0.84 (0.17)	0.19 (0.19) *
MoreiraS12	**0.95 (0.08)**	0.17 (0.17) *	0.76 (0.14)	0.26 (0.26)

* stands for $\epsilon_{3D} < 1.25$ and $\tau_{3D} < 0.25$

Bold typeset figures mean highest ϵ_{3D} and lowest τ_{3D} in the two configurations

averaged normalized error and computational costs of 1 for the Simple Configuration, whereas the mean normalized error is 0.74 (26 % lower than the baseline) and the mean normalized cost is 1.76 (76 % higher than the baseline) for the Best Configuration in Table 5. Figure 5 visualizes these results in a scatter plot. Selecting optimal parameters tends to decrease the positioning error, usually at the expense of higher computational costs. This negative slope can be clearly seen in three blocks: Kushki, methods based on grid-based clustering and other methods. The last group – including c-Means, Affinity Propagation and the majority rules– contains the models that achieve a good trade-off between the positioning error and the computational time.

Only c-Means, Affinity Propagation Clustering and the methods based on the strongest AP consistently achieved a normalized accuracy below 1.25 and a normalized computational burden below 0.25 (marked with * in the Table 5).

The computational cost of c-means decreases as the value of c increases, but the accuracy also decreases as shown in Figure 6. The trends reported in the figure also show that the parameters of c-Means (Traditional/Alternative generation of clustering and the value of c) have an impact in the normalized error and computational time. In general, the normalized accuracy of the traditional c-means is worse than the plain k-NN algorithm, whereas the results of the alternative c-means are similar to the plain k-NN method. Among all the methods based on c-means clustering, rfp1-MeansAlt provides a remarkably good compromise between accuracy improvement and time reduction, while also having a low variability.

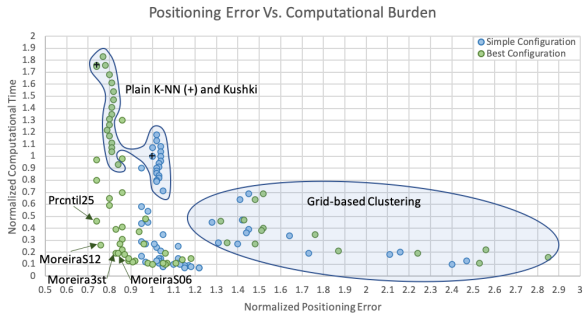

Fig. 5. Relation between the normalized positioning error and normalized computation time of the methods reported in Table 5.

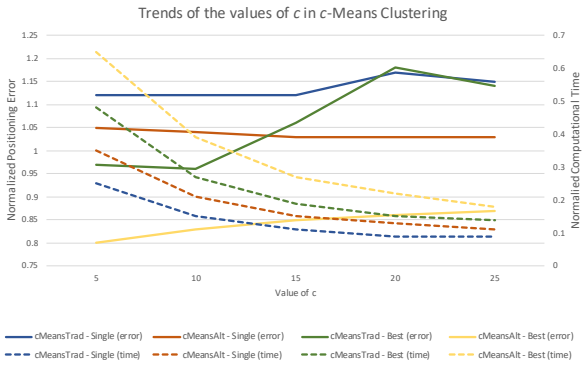

Fig. 6. Trends on c-Means Clustering through different c values.

The Affinity Propagation Clustering, while providing a notable reduction of the computational cost, may require significant memory and computational resources for the clustering stage. We had to use the Sparse implementation provided by its developers because of memory shortage problems for large datasets. Even using the Sparse implementation, the time required to compute the centroids was very high, lasting many hours for the largest databases.

The methods based on the strongest AP proposed by Marques and Moreira provided remarkable time reductions while commonly improving the positioning accuracy. The method proposed by Moreira that filters by one AP (Moreira1St) provides the best overall computation time, at the expense of slightly increasing the normalized positioning error, to 1.15 and 1.00 for the Simple and Best Configuration, respectively. The more APs are used in the filtering stage (1 or 2 in Marques10, 3 in Moreira3St), the more the positioning error reduces at the expense of a higher computation time. For the version of Moreira that is based on the similarity to the strongest AP, the positioning error and the computation time are well balanced when the threshold value is set to 6 dB (MoreiraS06). The accuracy is close to that of the plain k-NN but the normalized time is reduced by factor 8.

Other methods miscarry either in the error or time improvement. FengYu failed because the radio map reduction was entirely done during the operational phase without any off-line pre-processing. The methods based on the percentile rule kept the positioning error but did not accomplished a notable time reduction. The grid-based clustering methods performed very poorly in terms of error. The clustering proposed by Kushki led to an unexpected high computational time. The possible causes are explained in Section 4.2.2.

5 Discussion

Some attempts to reduce computational costs were identified in a literature review on FP in the major research data sets. Although most of them relied on k-Means clustering (c-Means in this paper) and Affinity Propagation, other interesting rules were found. We implemented and evaluated the reproducible ones, those that provided enough implementation details. The proposed evaluation framework allowed a deep analysis, which led to the following observations.

The election of some parameters of the FP method, like the distance function to compare the fingerprints, are usually only based on the positioning accuracy. According to our experience and the empirical results, we encourage to ponder the computational cost when selecting the most appropriate distance metrics. The Sørensen distance provides good accuracy with a reasonable increase of the computational costs. In contrast, the metrics based on probabilistic log-Gaussian distance provide the best accuracy in some cases but they have three times the computational cost of the Manhattan distance.

The computations in the on-line stage of clustering and optimization rules mainly depend on the complexity required to reduce the radio map and the resulting number of fingerprints. In some cases, the operations required to reduce the training set can be done off-line, which alleviates the computational costs during the on-line phase as we demonstrated with the optimized FengYu method.

Although some methods promise to reduce the computational load while barely affecting the accuracy, their assumptions are not applicable to many situations. Examples of these assumptions are, for instance, having multiple fingerprints in every reference point or a uniform distribution of fingerprints over the space. Usually, the methods' original evaluation fits the requirements and the system does not present any anomaly. However, they might provide a poorer accuracy when the evaluation is more realistic and includes features breaking those assumptions.

Grid-based clustering is not suggested. Averaging fingerprints only works when the distance between RPs is larger than the cell size and every RP has several fingerprints taken in the same conditions. Averaging should never be used neither when the local fingerprint density is very low not when it includes samples from different sources.

The methods based on the strongest AP proposed by Marques and Moreira, c-Means clustering and Affinity Propagation Clustering (APC) appear to be universally eligible to reduce the computational cost in the on-line stage of FP positioning while keeping an accuracy similar to plain k-NN. However, the computational costs and resources required to generate the clusters with APC might be prohibitive in large datasets.

Regarding c-Means clustering, the variant presented with the alternative initialization creates effective clusters that better group similar fingerprints with respect the traditional initialization. The two-stage search is effective as in the coarse search, the right cluster is selected and, then, the fine-grained search provides the right fingerprints to compute the position estimation. As explained in Section 4.1, the computational cost decreases as the number of clusters increases, which find the best scenario around the proposed heuristic $rfp1$ (square root of reference samples). However, as a side effect, the positioning error increases. As the number of clusters increases, the probability that the operational fingerprint falls near a boundary between clusters also increase. Selecting the wrong cluster and having reference fingerprints from the same reference position scattered in different clusters are the main causes of this side effect of c-Means with large c values.

In general, none of the methods guarantees that the reduced training sets have all the same size. The generated clusters are generated to group similar fingerprints in the feature (RSS) space, but they may not be equally distributed as the groups depend on distribution of APs, localization of the RPs, the devices used to collect the data and the noise in signal propagation. Similarly, in the optimization rules, the size of the reduced radio map depends also on the coverage of the APs detected on the operational phase. In other words, the computational costs vary depending on the reduced radio map linked to the operational fingerprint.

To sum up, most of the analyzed methods apply an off-line pre-processing stage [85]. It is devoted to create supporting data, e.g. clusters and reduced radio maps, for the on-line phase. That pre-processing is highly relevant for production systems, since it might avoid unnecessary calculations in the on-line phase that degrade a system's scalability. However, this pre-processing stage is not negligible, like in the case of Affinity Propagation Clustering where it took several hours for large radio maps on our hardware.

Finally, some of the analyzed methods could not be implemented due to the lack of details in the publications introducing them. Lack of details or procedures to set some parameters are the most common issues we faced when implementing the methods found in the literature. The Indoor Positioning community should promote the diffusion and communication of new methods ensuring reproducible research, e.g., publishing the code and data in a public repository. Also, different scenarios (e.g., through datasets) should be considered for generalization purposes.

6 Conclusion

This paper presents a comparison of different clustering and optimization rules to reduce the computational burden of Wi-Fi fingerprinting (FP) methods. Although researcher have already published partial results, they cannot be compared as the evaluation scenarios and metrics differ. Also, the results are usually restricted to one deployment (research facilities or small area) and cannot be generalized. The Indoor Positioning community needs an evaluation framework similar to the one used in Machine Learning with multiple datasets.

To the best of our knowledge, no other work has used an evaluation framework as comprehensive as the one we have presented here, which includes two aggregated normalized metrics and 16 datasets. We implemented 15 methods, testing several parameter values for some of them, to perform an empirical comprehensive comparison. An evaluation framework with heterogeneous datasets not only allowed us to generalise better on the evaluation metrics, but also enables the research community to compare new methods against a large set of deployments through datasets.

Balancing the general accuracy and the general computational costs, MoreiraS06 could be appointed as the best model within the analysed methods. Moreover, this model somehow benefits from the fact that two fingerprints sharing the same strongest AP with similar RSS value should be close in the space, which reduces the computational costs of FP without degrading its accuracy. In general, the methods based on the strongest AP proposed by Marques and Moreira, c-Means clustering and Affinity Propagation Clustering appear to work well in all considered scenarios, and thus are likely to be universally eligible. However, the cluster generation of Affinity Propagation is the most demanding one, requiring several hours or facing some execution issues in the largest datasets. The problem of averaging fingerprints with different features makes the methods based on clustering to be the best choice for single-device datasets with large local density, whereas the methods based on the strongest AP are mostly suited for datasets with low local density or collected by diverse devices.

The efficacy of reducing the computational costs of the on-line stage depends not only on the clustering or optimization method itself but also on the number APs and the spatial distribution of the fingerprints. A developer of a FP system should keep this in mind. Bold assumptions and requirements during the evaluation stage are not encouraged. Moreover, the way of implementing an algorithm may affect the computational costs. We should move as much as possible computation to the off-line pre-processing stage.

As further work we will proceed on the implementation of the FP models in other computer languages, including C (assembler) which is used for efficient implementations in embedded devices with few resources, to test, for instance, their feasibility for mobile apps. Moreover, our long-term objective is to settle the best practices for evaluating indoor positioning systems with multiple scenarios and datasets in order to ensure reproducible research. Our contribution to this goal starts here by making available the datasets and open-source code used in this work. The community can extend the proposed evaluation setup with their datasets that consider more realistic propagation models, devices not included in this work (new smartphones or computing devices) or, even, disruptive data collection strategies.

ACKNOWLEDGMENTS

This project have been funded by: Ministerio de Ciencia, Innovación y Universidades – INSIGNIA project (Programa Torres-Quevedo, PTQ2018-009981), Academy of Finland – PRISMA project (#313039); FCT – Fundação para a Ciência e a Tecnologia within the R&D Units Project Scope: UIDB/00319/2020; UJI's research programme (PREDOC/2016/55 and POSDOC-B/2018/12).

REFERENCES

[1] L. Yin, Q. Ni, and Z. Deng, 'A GNSS/5g integrated positioning methodology in d2d communication networks,' *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 2, pp. 351–362, 2018.

[2] P. Richter and M. Toledano-Ayala, 'Ubiquitous and seamless localization: Fusing GNSS pseudoranges and WLAN signal strengths,' *Mobile Information Systems*, vol. 2017, p. 16, 2017, *Mobile Geospatial Computing Systems for Ubiquitous Positioning*.

[3] W. Jiang, Y. Li, and C. Rizos, 'Optimal data fusion algorithm for navigation using triple integration of PPP-GNSS, INS, and terrestrial ranging system,' *IEEE Sensors Journal*, vol. 15, no. 10, pp. 5634–5644, Oct. 2015.

[4] N. Klepeis, W. Nelson, W. Ott, *et al.*, 'The national human activity pattern survey (nhaps): A resource for assessing exposure to environmental pollutants,' *Journal of Exposure Analysis and Environmental Epidemiology*, vol. 01, pp. 231–252, 2001.

[5] G. Shtar, B. Shapira, and L. Rokach, 'Clustering wi-fi fingerprints for indoor–outdoor detection,' *Wireless Networks*, vol. 25, no. 3, pp. 1341–1359, Apr. 2019.

[6] S. He and S. Chan, 'Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons,' *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, Firstquarter 2016.

[7] P. Bahl and V. N. Padmanabhan, 'Radar: An in-building RF-based user location and tracking system,' in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings.*, vol. 2, 2000, pp. 775–784.

[8] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, 'Kernel-based positioning in wireless local area networks,' *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 689–705, 2007.

[9] R. Berkvens, H. Peremans, and M. Weyn, 'Conditional entropy and location error in indoor localization using probabilistic wi-fi fingerprinting,' *Sensors*, vol. 16, no. 10, 2016.

[10] E. Mok and B. K. S. Cheung, 'An improved neural network training algorithm for wi-fi fingerprinting positioning,' *ISPRS Int. Journal of Geo-Information*, vol. 2, no. 3, pp. 854–868, 2013.

[11] A. B. Adege, L. Yen, H. Lin, *et al.*, 'Applying deep neural network (dnn) for large-scale indoor localization using feed-forward neural network (ffnn) algorithm,' in *2018 IEEE International Conference on Applied System Invention (ICASI)*, 2018, pp. 814–817.

[12] M. Nowicki and J. Wietrzykowski, 'Low-effort place recognition with wifi fingerprints using deep learning,' in *Automation 2017*, R. Szewczyk, C. Zieliński, and M. Kaliczyńska, Eds., Cham: Springer International Publishing, 2017, pp. 575–584.

[13] D. Lymberopoulos and J. Liu, 'The microsoft indoor localization competition: Experiences and lessons learned,' *IEEE Signal Processing Magazine*, vol. 34, no. 5, pp. 125–140, 2017.

[14] C. Feng, W. S. A. Au, S. Valaee, *et al.*, 'Received-signal-strength-based indoor positioning using compressive sensing,' *IEEE Trans. on Mobile Computing*, vol. 11, no. 12, pp. 1983–1993, Dec. 2012.

[15] G. P. Gempita, D. Wilasari, P. Kristalina, *et al.*, 'Implementation of k-nn fingerprint method on receiving server for indoor mobile object tracking,' in *2019 International Electronics Symposium (IES)*, 2019, pp. 411–416.

[16] J. Yoo, 'Change detection of rssi fingerprint pattern for indoor positioning system,' *IEEE Sensors Journal*, vol. 20, no. 5, pp. 2608–2615, 2020.

[17] J. Torres-Sospedra, A. R. Jiménez, S. Knauth, *et al.*, 'The smartphone-based offline indoor location competition at IPIN 2016: Analysis and future work,' *Sensors (Switzerland)*, vol. 17, no. 3, pp. 1–17, 2017.

[18] J. Torres-Sospedra, A. Jiménez, A. Moreira, *et al.*, 'Off-Line Evaluation of Mobile-Centric Indoor Positioning Systems: The Experiences from the 2017 IPIN Competition,' *Sensors*, vol. 18, no. 2, p. 487, 2018.

[19] 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.

[20] N. Soltanieh, Y. Norouzi, Y. Yang, *et al.*, 'A review of radio frequency fingerprinting techniques,' *IEEE Journal of Radio Frequency Identification*, pp. 1–1, 2020.

[21] J. Luo and L. Fu, 'A smartphone indoor localization algorithm based on wlan location fingerprinting with feature extraction and clustering,' *Sensors*, vol. 17, no. 6, 2017.

[22] H. Shin and H. Cha, 'Wi-fi fingerprint-based topological map building for indoor user tracking,' in *International Conference on Embedded and Real-Time Computing Systems and Applications*, 2010.

[23] A. Moreira, M. J. Nicolau, F. Meneses, *et al.*, 'Wi-fi fingerprinting in the real world - RTLS@UM at the EvAAL competition,' in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, Oct. 2015.

[24] Y. Chen, D. Lymberopoulos, J. Liu, *et al.*, 'Indoor localization using FM signals,' *IEEE Transactions on Mobile Computing*, vol. 12, no. 8, pp. 1502–1517, 2013.

[25] W. Chen, Q. Chang, H.-t. Hou, *et al.*, 'A novel clustering and kwnn-based strategy for wi-fi fingerprint indoor localization,' in *2015 4th International Conference on Computer Science and Network Technology (ICCSNT)*, vol. 01, Dec. 2015, pp. 49–52.

[26] N. Marques, F. Meneses, and A. Moreira, 'Combining similarity functions and majority rules for multi-building, multi-floor, WiFi positioning,' in *2012 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, IEEE, Nov. 2012.

[27] F. Yu, M. Jiang, J. Liang, *et al.*, '5G WiFi signal-based indoor localization system using cluster k-nearest neighbor algorithm,' *International Journal of Distributed Sensor Networks*, vol. 10, no. 12, p. 247 525, Jan. 2014.

[28] V. T. Haute, D. E. Poorter, I. Moerman, *et al.*, 'Comparability of rf-based indoor localisation solutions in heterogeneous environments: An experimental study,' *IJAHUC*, pp. 92–114, 2016.

[29] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, *et al.*, 'UJIIndoorLoc: A new multi-building and multi-floor database for WLAN fingerprint-based indoor localization problems,' in *Proceedings of the Fifth Conference on Indoor Positioning and Indoor Navigation*, Database available in https://archive.ics.uci.edu/ml/datasets/UJIIndoorLoc, 2014, pp. 261–270.

[30] Z. Iqbal, D. Luo, P. Henry, *et al.*, 'Accurate real time localization tracking in a clinical environment using Bluetooth low energy and deep learning,' *PLOS ONE*, vol. 13, no. 10, Oct. 2018.

[31] 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.

[32] Widyawan, M. Klepal, and D. Pesch, 'Influence of predicted and measured fingerprint on the accuracy of rssi-based indoor location systems,' in *2007 4th Workshop on Positioning, Navigation and Communication*, 2007, pp. 145–151.

[33] F. Lemic, V. Handziski, G. Caso, *et al.*, 'Enriched training database for improving the wifi rssi-based indoor fingerprinting performance,' in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, 2016, pp. 875–881.

[34] G. Caso and L. De Nardis, 'Virtual and oriented WiFi fingerprinting indoor positioning based on multi-wall multi-floor propagation models,' *Mobile Networks and Applications*, vol. 22, no. 5, pp. 825–833, Oct. 1, 2017.

[35] G. G. Anagnostopoulos and A. Kalousis, 'A reproducible analysis of rssi fingerprinting for outdoor localization using sigfox: Preprocessing and hyperparameter tuning,' in *2019 International Conference on Indoor Positioning and Indoor Navigation*, Sep. 2019.

[36] S. Khandker, R. Mondal, and T. Ristaniemi, 'Positioning error prediction and training data evaluation in rf fingerprinting method,' in *2019 International Conference on Indoor Positioning and Indoor Navigation*, Sep. 2019.

[37] B. Li, J. Salter, A. G. Dempster, *et al.*, 'Indoor positioning techniques based on wireless LAN,' in *First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, 2007, pp. 13–16.

[38] A. Baniukevic, D. Sabonis, C. S. Jensen, *et al.*, 'Improving wi-fi based indoor positioning using Bluetooth add-ons,' in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, Jun. 2011, pp. 246–255.

[39] P. Bolliger, 'Redpin - adaptive, zero-configuration indoor localization through user collaboration,' in *Proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environments*, ACM, 2008, pp. 55–60.

[40] J. Ledlie, J.-G. Park, D. Curtis, *et al.*, 'Molé: A scalable, user-generated wifi positioning engine,' *Journal of Location Based Services*, vol. 6, no. 2, pp. 55–80, 2012.

[41] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, *et al.*, 'Zee: Zero-effort crowdsourcing for indoor localization,' in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12, ACM, 2012, pp. 293–304.

[42] W. Zhao, S. Han, R. Q. Hu, *et al.*, 'Crowdsourcing and multi-source fusion-based fingerprint sensing in smartphone localization,' *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3236–3247, 2018.

[43] Y. Gu, C. Zhou, A. Wieser, *et al.*, 'Trajectory estimation and crowdsourced radio map establishment from foot-mounted imus, wi-fi fingerprints, and gps positions,' *IEEE Sensors Journal*, vol. 19, no. 3, pp. 1104–1113, Feb. 2019.

[44] P. Richter and M. Toledano-Ayala, 'Revisiting gaussian process regression modeling for localization in wireless sensor networks,' *Sensors*, vol. 15, no. 9, pp. 22 587–22 615, Sep. 2015.

[45] N. Hernández, M. Ocaña, J. M. Alonso, *et al.*, 'Continuous space estimation: Increasing WiFi-based indoor localization resolution without increasing the site-survey effort,' *Sensors*, vol. 17, 2017.

[46] T. Wang, P. Tseng, Y. Chan, *et al.*, 'A ray-tracing based fingerprinting for indoor positioning,' in *IEEE Intl Conf on Scalable Computing and Communications and Its Associated Workshops*, 2015.

[47] M. M. Atia, A. Noureldin, and M. J. Korenberg, 'Dynamic online-calibrated radio maps for indoor positioning in wireless local area networks,' *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1774–1787, Sep. 2013.

[48] C. Pendão and A. Moreira, 'Fast graph - organic 3D graph for unsupervised location and mapping,' in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018.

[49] A. Arya, P. Godlewski, and P. Melle, 'A hierarchical clustering technique for radio map compression in location fingerprinting systems,' in *71st IEEE Vehicular Technology Conference*, 2010.

[50] T. J. Gallagher, B. Li, A. G. Dempster, *et al.*, 'A sector-based campus-wide indoor positioning system,' in *2010 International Conference on Indoor Positioning and Indoor Navigation*, 2010.

[51] M. A. Youssef, A. Agrawala, and A. U. Shankar, 'WLAN location determination via clustering and probability distributions,' in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, IEEE Comput. Soc, 2003.

[52] T. King, T. Butter, M. Brantner, *et al.*, 'Distribution of fingerprints for 802.11-based positioning systems,' in *2007 International Conference on Mobile Data Management*, May 2007, pp. 224–226.

[53] B. Li, I. Quader, and A. G. Dempster, 'On outdoor positioning with wi-fi,' *Journal of Global Positioning Systems*, vol. 7, 2008.

[54] B. Wang, X. Liu, B. Yu, *et al.*, 'An improved wifi positioning method based on fingerprint clustering and signal weighted euclidean distance,' *Sensors*, vol. 19, no. 10, 2019.

[55] A. Razavi, M. Valkama, and E.-S. Lohan, 'K-means fingerprint clustering for low-complexity floor estimation in indoor mobile localization,' in *IEEE Globecom Workshops (GC Wkshps)*, 2015.

[56] B. Altintas and T. Serif, 'Improving rss-based indoor positioning algorithm via k-means clustering,' in *17th European Wireless 2011 - Sustainable Wireless Technologies*, Apr. 2011, pp. 1–5.

[57] G. Caso, L. De Nardis, and M.-G. Di Benedetto, 'A mixed approach to similarity metric selection in affinity propagation-based wifi fingerprinting indoor positioning,' *Sensors*, vol. 15, no. 11, pp. 27 692–27 720, 2015.

[58] W. Liu, X. Fu, and Z. Deng, 'Coordinate-based clustering method for indoor fingerprinting localization in dense cluttered environments,' *Sensors*, vol. 16, no. 12, 2016.

[59] A. Moreira, I. Silva, and J. Torres-Sospedra. (Apr. 2020). The DSI dataset for Wi-Fi fingerprinting using mobile devices. version 1.0, [Online]. Available: https://doi.org/10.5281/zenodo.3778646.

[60] T. King, T. Haenselmann, and W. Effelsberg, 'On-demand fingerprint selection for 802.11-based positioning systems,' in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks*, Jun. 2008, pp. 1–8.

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

[62] J. Torres-Sospedra and A. Moreira, 'Analysis of sources of large positioning errors in deterministic fingerprinting,' *Sensors*, vol. 17, no. 12, 2017.

[63] S. Shrestha, J. Talvitie, and E. S. Lohan, 'Deconvolution-based indoor localization with WLAN signals and unknown access point locations,' 2013.

[64] E.-S. Lohan, J. Torres-Sospedra, H. Leppäkoski, *et al.*, 'Wi-fi crowdsourced fingerprinting dataset for indoor positioning,' *MDPI Data*, vol. 2, no. 4, Oct. 2017, Database available in https://zenodo.org/record/889797.

[65] P. Richter, E. S. Lohan, and J. Talvitie. (Jan. 2018). WLAN (WiFi) rss database for fingerprinting positioning, [Online]. Available: https://zenodo.org/record/1161525.

[66] Lohan. (May 2020). Additional TAU datasets for Wi-Fi fingerprinting- based positioning. version v1, 11.05.2020, [Online]. Available: https://doi.org/10.5281/zenodo.3819917.

[67] J. Rojo, G. M. Mendoza-Silva, G. Ristow Cidral, *et al.*, 'Machine learning applied to wi-fi fingerprinting: The experiences of the ubiqum challenge,' in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1–8.

[68] J. Torres-Sospedra, A. J. C. Moreira, S. Knauth, *et al.*, 'A realistic evaluation of indoor positioning systems based on wi-fi fingerprinting: The 2015 evaal-etri competition,' *JAISE*, vol. 9, no. 2, pp. 263–279, 2017.

[69] J. Torres-Sospedra, P. Richter, E. S. Lohan, *et al.* (2020). Supporting Materials for 'A Comprehensive and Reproducible Comparison of Clustering and Optimization Rules in Wi-Fi Fingerprinting'. https://doi.org/10.5281/zenodo.3968503.

[70] S. Lloyd, 'Least squares quantization in pcm,' *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[71] D. Arthur and S. Vassilvitskii, 'K-means++: The advantages of careful seeding,' in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007, pp. 1027–1035.

[72] B. J. Frey and D. Dueck, 'Clustering by passing messages between data points,' *Science*, vol. 315, p. 2007, 2007.

[73] Probabilistic and Statistical Inference Group. (2014). Tools & software, [Online]. Available: https://psi.toronto.edu/tools/.

[74] J. Han, J. Pei, and M. Kamber, *Data mining: Concepts and techniques*, 3rd ed. Waltham, MA 02451, USA: Elsevier, 2011.

[75] J. Machaj, P. Brida, and R. Piché, 'Rank based fingerprinting algorithm for indoor positioning,' in *2nd International Conference on Indoor Positioning and Indoor Navigation*, 2011.

[76] K. Kaemarungsi and P. Krishnamurthy, 'Properties of indoor received signal strength for wlan location fingerprinting,' in *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, 2004, pp. 14–23.

[77] W. Njima, I. Ahriz, R. Zayani, *et al.*, 'Deep cnn for indoor localization in iot-sensor systems,' *Sensors*, vol. 19, no. 14, 2019.

[78] A. Alsmady and F. Awad, 'Optimal wi-fi access point placement for rssi-based indoor localization using genetic algorithm,' in *2017 8th International Conference on Information and Communication Systems (ICICS)*, 2017, pp. 287–291.

[79] J. Bi, Y. Wang, Z. Li, *et al.*, 'Fast radio map construction by using adaptive path loss model interpolation in large-scale building,' *Sensors*, vol. 19, no. 3, 2019.

[80] S. H. Cha, 'Comprehensive survey on distance/similarity measures between probability density functions,' *International Journal of Mathematical Models and Methods in Applied Sciences*, vol. 1, no. 4, pp. 300–307, 2007.

[81] J. Torres-Sospedra, R. Montoliu, S. Trilles, *et al.*, 'Comprehensive analysis of distance and similarity measures for Wi-Fi finger-

printing indoor positioning systems,' *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.
[82] K. Gilad and L. Antsfeld, *Cluster-based fingerprinting algorithms*, US Patent 8,180,367, May 2012.
[83] A. Cramariuc, H. Huttunen, and E. S. Lohan, 'Clustering benefits in mobile-centric WiFi positioning in multi-floor buildings,' in *2016 International Conference on Localization and GNSS*, 2016.
[84] G. Caso, L. De Nardis, F. Lemic, *et al.*, 'Vifi: Virtual fingerprinting wifi-based indoor positioning via multi-wall multi-floor propagation model,' *IEEE Transactions on Mobile Computing*, 2019.
[85] Y. Zhang, S. Zhang, R. Li, *et al.*, 'WiFi fingerprint positioning based on clustering in mobile crowdsourcing system,' in *Int. Conf. on Computer Science and Education*, 2017, pp. 252–256.

Joaquín Torres-Sospedra received his PhD on Ensembles of Neural Networks and Machine Learning from Universitat Jaume I in 2011. In April 2013, he joined the Institute of New Imaging Technologies (INIT, Universitat Jaume I) where he led Indoor Positioning projects. Since January 2020, he is the Scientific Coordinator of UBIK Geospatial Solutions and still collaborates with the INIT, as well as other international research institutions. He has authored more than 120 articles in journals and conferences. His current research interests include indoor positioning solutions based on Wi-Fi & BLE, Machine Learning and Evaluation. He has supervised 5 Master and 2 PhD Students. Currently, he is supervising 6 PhD students. He is the chair of the IPIN International Standards Committee and IPIN Smartphone-based off-site Competition.

Philipp Richter received his Dipl.-Ing. degree in Electrical Engineering from Technische Universität Darmstadt, Germany, and his D.Eng. from Universidad Autónoma de Querétaro, Mexico, in 2008 and 2016, respectively. From 2009 to 2012 he worked as an research associate at Fraunhofer IIS in Nuremberg. Until fall 2019, he worked as a post-doctoral researcher at Tampere University, Finland, and he is currently with u-blox, Finland. His research interests lie in the design and analysis of robust signal processing, Bayesian inference, machine learning algorithms and their application in positioning, currently particularly in satellite-based positioning.

Adriano Moreira is an Associate Professor, with Habilitation, at University of Minho, and a researcher at the Algoritmi Research Centre. He received the "Licenciatura" degree in Electronics and Telecommunications Engineering and the PhD degree in Electrical Engineering, respectively in 1989 and 1997, from the University of Aveiro. He co-founded the Computer Communications and Pervasive Media research group, and is the Director of the MAP-tele doctoral program in Telecommunications. His research activities have been taking place within the ubicom@uminho research sub-group, which has been focusing in the creation of technologies for smart places. In the past few years he participated in many research projects funded by national and EU programs. He is the author of several scientific publications in conferences and journals, and the author of one patent in the area of computational geometry. Together with his colleagues, won the 1st prize on the off-site track of the EvAAL-ETRI Indoor Localization Competition (IPIN 2015 and 2017) and the 2nd prize of the corresponding competition in 2016.

Elena-Simona Lohan received her MSc degree from Polytechnics University of Bucharest (1997), a DEA degree in Econometrics at Ecole Polytechnique, Paris (1998), and a PhD degree in Telecommunications from Tampere University of Technology (2003). She is now an Assoc. Prof. at Tampere University and a Visiting Professor at Universitat Autonoma de Barcelona. She is a co-editor of the first book on Galileo satellite system (Springer "Galileo Positioning technology"), co-editor of the Springer book "Multi-technology positioning", and author/co-author of more than 185 international peer-reviewed publications, 6 patents and inventions. In the past 4 years, she has been a Principal Investigator in 5 national and 4 EU projects. She is also an associate Editor for RIN Journal of Navigation and for IET Journal on Radar, Sonar, and Navigation.

Germán Martín Mendoza-Silva has a Bachelor in Computer Science from the University of Oriente, Cuba in 2005 and a Msc. in Geospatial Technologies from WWU (Germany), UNL (Portugal) and UJI (Spain) in 2015. Currently, he is a PhD. student in the Institute of New Imaging Technologies at UJI, Spain, focused on WLAN-based indoor positioning, indoor navigation, machine learning and GIS applications.

Sergio Trilles received his PhD in Integration of Geospatial Information from the Jaume I University in 2015. He is author of more than fifty journal and conference peer-reviewed publications. He the opportunity to work four months as researcher in the Digital Earth and Reference Data Unit of the European Commission's Joint Research Centre (JRC). Currently he is a post-doc researcher at the GEOTEC group.

Miguel Matey-Sanz has a bachelor's degree in Computer Science from Universitat Jaume I. Currently, he is studying a master's degree in Intelligent Systems. He joined the GEOTEC group (Institute of New Imaging Technologies) via the Study and Research program, that allow students to start their research career while they are studying their bachelor's degree.

Joaquín Huerta is full professor at the Department of Information Systems from University Jaume I in Spain, where he teaches several courses related to GIS and Internet Technologies. His current research interests are indoor positioning, smart cities, mobile and web GIS applications and augmented reality. He is the head of the GEOTEC Research Group, Director of the Erasmus Mundus Master of Science in Geospatial Technologies degree program, run jointly with the universities of Münster and Nova de Lisboa. He is and has been principal investigator of several research projects including EU projects as A-WEAR, GEO-C, EUROGEOSS. In addition to academic activities he is founding member of UBIK Geospatial Solutions (http://ubikgs.com)