Lightweight Wi-Fi Fingerprinting with a Novel RSS Clustering Algorithm

Lightweight Wi-Fi Fingerprinting with a Novel RSS Clustering Algorithm

Darwin Quezada-Gaibor[*,†], Joaquín Torres-Sospedra[‡],
Jari Nurmi[†], Yevgeni Koucheryavy[†], and Joaquín Huerta[*]
[*]*Institute of New Imaging Technologies*, *Universitat Jaume I*, Castellón, Spain
[†]*Electrical Engineering Unit*, *Tampere University*, Tampere, Finland
[‡]*UBIK Geospatial Solutions S.L.*, Castellón, Spain

***Abstract*—Nowadays, several indoor positioning solutions support Wi-Fi and use this technology to estimate the user position. It is characterized by its low cost, availability in indoor and outdoor environments, and a wide variety of devices support Wi-Fi technology. However, this technique suffers from scalability problems when the radio map has a large number of reference fingerprints because this might increase the time response in the operational phase. In order to minimize the time response, many solutions have been proposed along the time. The most common solution is to divide the data set into clusters. Thus, the incoming fingerprint will be compared with a specific number of samples grouped by, for instance similarity (clusters). Many of the current studies have proposed a variety of solutions based on the modification of traditional clustering algorithms in order to provide a better distribution of samples and reduce the computational load. This work proposes a new clustering method based on the maximum Received Signal Strength (RSS) values to join similar fingerprints. As a result, the proposed fingerprinting clustering method outperforms three of the most well-known clustering algorithms in terms of processing time at the operational phase of fingerprinting.**

***Index Terms*—Indoor Positioning, Wi-Fi fingerprinting, Clustering, Computing Efficiency**

I. Introduction

The rapid increase of mobile and Internet of Things (IoT) devices, which use localisation services for indoor and outdoor environments, demands efficient technologies and methods which provides high positioning accuracy while having low power consumption. This requirement has lead to the development of several smart technologies in the past decade, that includes Bluetooth Low Energy (BLE) [1], ultra-wideband (UWB) [2], Visible light communication (VLC) [3], among others. However, many Indoor Positioning System (IPS) still use the well-know technology Wi-Fi in order to reuse the infrastructure (Access Point (AP) or Wi-Fi routers) deployed in the environment, specially in smartphone-based applications but also for robots and autonomous vehicles driving. Additionally, the most common technique used with Wi-Fi is fingerprinting.

Corresponding Author: D. Quezada Gaibor (quezada@uji.com)

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Sklodowska Curie grant agreement No. 813278 (A-WEAR: A network for dynamic wearable applications with privacy constraints, http://www.a-wear.eu/).; J. Torres-Sospedra gratefully acknowledge funding from Ministerio de Ciencia, Innovación y Universidades (INSIGNIA, PTQ2018-009981)

Fingerprinting is divided into two phases, off-line and online phase. In the off-line phase, the reference fingerprints are collected, generating a radio map. Different techniques are used to reduce the data set dimensionality and/or form clusters [4]. The online phase is devoted to find the most similar fingerprint between the incoming fingerprint at an unknown position and the reference fingerprints with known positions.

The major limitation of Wi-Fi fingerprinting is related to the number of fingerprints stored in the data set, given that the performance of this technique might be affected when there are hundreds or thousands of fingerprints in the data set (radio map). It reduces the time response in the online phase of Wi-Fi fingerprinting, due to the incoming fingerprint has to be compared with each fingerprint in the radio map. As a result, it is not an efficient approach for real-time indoor positioning applications. In order to provide fast response in the online phase of Wi-Fi fingerprinting, some authors have proposed a wide range of solutions such as the use clustering methods (k-Means, Density-based Spatial Clustering of Applications with Noise (DBSCAN), C-Means, affinity propagation, etc.) [5]–[7], dimensionality reduction, data compression [8], or rules based on the signal propagation [9], [10].

Different research articles have proposed some alternative to the existing clustering algorithms in order to improve the execution time. For instance, [11] implemented some modifications to C-Means clustering in order to reduce the position error without affecting the radio map integrity. Similarly, [12] used k-Means to group similar points according to the real position of the user instead of using the signal distance. As a result, the improved in the position accuracy is demonstrated.

In contrast with traditional clustering algorithms, we propose a new clustering algorithm, namely fingerprinting clustering (FPC) which is based on the maximum RSS values and the APs or routers in common between samples. In general, some clustering algorithms base their classification on specific distance metrics such as Euclidean distance, city block (Manhattan) or statistical methods to join similar points. The proposed method attempts to group fingerprints taking into account the RSS received in a specific point. The proposed clustering algorithm does not require a pre-defined number of clusters to form the groups. The number of clusters thus is given by the maximum RSS values of each sample, and the corresponding AP of that RSS value.

FPC was designed to be a lightweight clustering algorithm that requires minimal computational resources, providing fast time response with a tolerable rate of positioning error. FPC is conceived to be used in low-profile devices (e.g., smartwatches or basic smartphones). Thus, we also provide a comparison between FPC algorithm and three traditional clustering algorithms (k-Means, C-Means and DBSCAN) in this paper. This comparison was carried out considering the time required to form the clusters, the full time to estimate the user position by combining these algorithms with k-nearest neighbors (k-NN), and the average number of samples per cluster in each data set.

The main contributions of this work are the following:

- A new clustering method based on the RSS values and similar APs in common between samples.
- An extended analysis of the proposed algorithm in multiple open-source Wi-Fi radio maps.

This article is divided into four sections as follow. Section II provides a general overview of clustering algorithms and related work. Section III describes the proposal clustering algorithms for Wi-Fi fingerprinting. Section IV describes the experiments carried out in this work and their results. Section V offers a brief discussion of the results. Finally, section VI provides the main conclusions raised from the findings.

II. Related work

Clustering algorithms have been used to group data with similar characteristics into some classes (clusters). These clustering algorithms are widely used in number of applications, they have also been adopted for indoor positioning, especially with Wi-Fi fingerprinting. As a result of applying clustering, the IPSs were improved in different dimensions.

For instance, Xue *et al.* [12] proposed a new method to reduce the error in the position estimation for fingerprint-based systems. This method is devoted to join the nearest fingerprints to the reference points using the real physical distance between points. Hence, this new clustering algorithm uses both k-NN and k-Means in different processes to enhance the classes' accuracy. As a result, the authors have demonstrated the significant reduction of the positioning error in comparison with the original algorithm (k-Means).

Tao *et al.* [13] used a clustering algorithm based on k-medoids to detect outliers and find the correct cluster for the operational fingerprint in the online phase. Unlike other clustering algorithms (e.g., k-Means), their proposal does not need a pre-defined number of cluster to form the classes. Additionally, the authors propose a new distance metric for clustering. Thus, that study achieved a low positioning error by combining the clustering method with the system proposed.

Cui *et al.* [14] combined weighted k-nearest neighbor (WKNN) and affinity propagation clustering (APC) to estimate the user position. Thus, APC is used to determine the best centroid and the similarities between the samples according to two messages (responsibility and availability) given by the centroid. Thus, this research proposed a robust method to form the radio map based on crowd-sourcing data collection.

Wang *et al.* [15] proposed a novel method based on signal weighted euclidean distance (SWED) and position label-assisted (PL-assisted) clustering. Thereby, the authors obtained better performance than the traditional k-Means, and also, the positioning error reported was lower than WKNN using Manhattan and euclidean distance.

Ren *et al.* [11] provided a novel clustering method based on C-Means. This clustering method aims to optimize the radio map. The clustering algorithm proposed by the authors is capable of creating a common area between fingerprints. Thus, the proposed algorithm a better performance in terms of position accuracy and time response.

Wang *et al.* [7] developed a new algorithm, namely DBSCAN-KRF, combining three different approaches, the first one is DBSCAN, k-NN and random forest algorithm. Therefore, using DBSCAN the authors were able to remove noisy samples from the radio map. The user position is estimated using k-NN or random forest according to the following criteria. k-NN is used when the region is sensitive, which means that the clusters contain a low level of samples of different reference points. In contrast, if the cluster contains multiple samples of other reference points, a random forest will be used as the main core IPS. The authors thus achieved a low positioning error (error $<$ 1.5m).

Recently, we have also modified the k-Means algorithm to reduce the computational cost without degrading the performance in the positioning accuracy [4]. We provided different rules to not only reduce the time required to select the optimal cluster, but also to reduce the number of reference fingerprints within the winning cluster on-the-fly. The rules applied were based on signal propagation properties.

As we can see, the clustering algorithms used in the analyzed studies are based on traditional clustering approaches such as k-Means, C-Means, APC or DBSCAN. Although the clustering methods mentioned in earlier paragraphs provided an acceptable positioning accuracy, the computational load required to execute each of the proposed algorithm is not mentioned in any of these articles. As most of the traditional clustering algorithms were not developed for indoor positioning purposes, they need to be modified or improved to provide better performance in IPS applications.

III. Algorithm

Given that Wi-Fi fingerprinting data sets may contain hundreds or thousands of fingerprints, which are used in order to estimate the user or device position. These data sets might increase over the time when new environments are added or update through crowdsourced data collection technique.

In order to reduce the match time in the online phase of Wi-Fi fingerprinting, we propose a new clustering method based on the radio signal intensity received in a specific point. Unlike other clustering methods, this new approach is not based on any distance metric, it takes into account RSS, and similar APs in view in a particular position.

As each AP or router cover a specific area (radius of coverage), which is given by the configuration and the standard

(802.11a/b/g/n) used in the access point, the coverage area can be different in each environment. Thus, [16] provided a comprehensive study of the empirical coverage area and the RSS in it. The authors mentioned multiple factors that might affect the signal propagation indoors, such as walls, antenna direction, and people. The presence of people is a significant source of attenuation and absorption [17]. Hence, indoor is a challenging environment for signal propagation, and therefore, for indoor positioning based on radio frequency technologies. Despite this limitation, Wi-Fi technology and fingerprinting technique are still widely used in many IPSs.

The fingerprints for the radio map are collected in the offline phase. Each fingerprint is referenced with its position (x, y, z, floor and building). This radio map will be used in the online phase to match the operational fingerprint (unknown position) with the reference fingerprints and estimate the user position.

Generally, a radio map data set (Ψ) is formed by m number of samples and n number of APs ($m \times n$). Each position in the radio map belong to a RSS value (ψ_{ij}), where $i = 1, 2, ..., m$ and $j = 1, 2, ..., n$.

$$\Psi = \begin{bmatrix} \psi_{11} & \dots & \psi_{1n} \\ \vdots & \ddots & \vdots \\ \psi_{m1} & \dots & \psi_{mn} \end{bmatrix}$$

Thus, in order to divide the radio map into x number of clusters, we propose a new clustering method described in the following steps:

A. Data Preparation

1) Data Representation: In [18], the authors used three main data representations *exponential*, *positive* and *powed*. The data representation is linked with each radio map. For instance, *powed* data representation was used for UJI 1, TUT 2 and DSI 1. In this article, we used the same data representation for each data set.

2) Remove features with zero values: In this step, if the feature does not contain any valid RSS value, it is removed from the data set. It will avoid delays in the data processing.

B. Initial clusters

Once the data is prepared, the next step is to form the initial clusters. These clusters are formed according to the maximum RSS value in each fingerprint, which is related to a specific AP. Thus, the clusters will be formed if the maximum RSS value of each fingerprint (sample) are in the same AP (feature) ($\Psi_i \in C_j \Leftrightarrow MAX\Psi_i \in AP_j$, where C_j represents a j-th cluster). More restrictive clusters can be formed by using more than one maximum RSS value (unique maximum RSS values). When the initial clusters are available, we compute the centroid for each initial cluster (i.e., the mean of all objects in the cluster), similar to k-Means.

C. Joining small clusters

Suppose the number of fingerprints in some clusters are less than the average number of fingerprints of all the clusters. In that case, we compute the degree of relationship between the centroids of the small clusters by using the correlation coefficient (see Eq.1). Therefore, only those clusters with a high level of similarity are joined in only one cluster (see Figure 1). This process will end when the number of fingerprints of the new clusters are greater than or equal to the average number of fingerprints. Finally, the centroid of each new cluster is computed. Algorithm 1 describes how the FPC works.

$$\rho(Csc_A, Csc_B) = \frac{cov(Csc_A, Csc_B)}{\sigma_{Csc_A}\sigma_{Csc_B}} \quad (1)$$

where, ρ is the correlation coefficient between two centroids (Csc_A and Csc_B), Csc represents the centroid of an small cluster, σ is the standard deviation, and $conv$ is the covariance between two centroids.

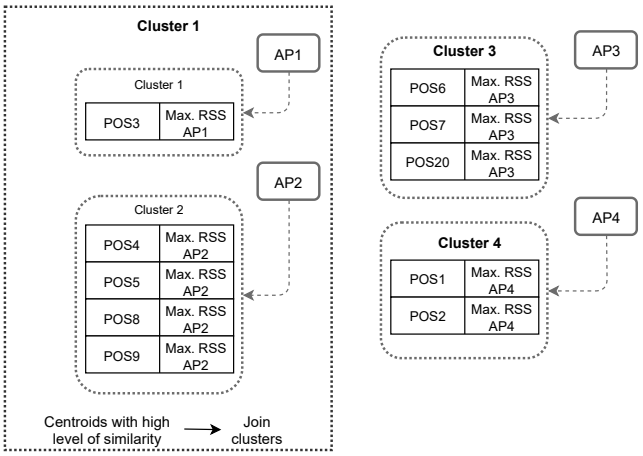

Fig. 1. Basic operation of the proposed clustering algorithm

As it is given, the proposed clustering algorithm requires three parameters: the data set, the data representation, and the number of maximum RSS values used to form the cluster. If the user does not specify the data representation, the positive data representation is selected by default. Similarly, the default number of maximum RSS values is set to 1. FPC returns the clusters indexes (*IDX*) and the *centroids* of each cluster.

IV. Experiments and Results

A. Experiment setup

The proposed algorithm has been evaluated over 17 Wi-Fi and 3 BLE open-access data sets. The advantage of a comprehensive asssesment is given by the heterogeneity in both scenarios and devices, which permit reaching more general conclusions about the proposed algorithm [4], [8], [18]. The datasets we use were collected by: U. Jaume I (LIB 1–2 [19], UJI 1–2 [20], SIM [18]), Tampere U. (TUT 1–7 [21]–[26]), Mannheim U.(MAN 1–2 [27], [28]), U. of Minho (DSI 1–2 [29], MINT 1 [30]), U. of Extremadura (UEXB 1-3 [31]).

Additionally, we have compared the FPC with three traditional clustering algorithms which are: k-Means, C-Means and DBSCAN. These traditional clustering algorithms have

Algorithm 1: RFP Clustering

Input: $dataset, dataRep, numMaxRss$
Output: $IDX, centroids$
Function `RfpClustering` $(dataset, dataRep, numMaxRss)$ **:**
  // Data preparation ▷ First process
  $dataset \longleftarrow data\_rep(dataset)$
  $dataset \longleftarrow remove\_zero\_features(dataset)$
  // Initial clusters ▷ Second process
  $\Psi_i \in C_j \Leftrightarrow MAX\Psi_i \in AP_j$
  // Each cluster is denoted by a cluster index (IDX)
  // Compute the centroid of each cluster (mean of each cluster)
  $\overline{C_j} = \frac{1}{|C_j|} \sum_{\Psi_i \in C_j} \Psi_i$
  // Join small clusters ▷ Third process
  // Compute the average number of samples in the cluster
  $\overline{x} = \frac{1}{M} \sum_{j=1}^{M} C_j$
  // Compute the correlation coefficient between the centroids of small clusters
  C_j is small $\Leftrightarrow$ the number of samples in $C_j < \overline{x}$
  ccm = $correlation\_coeff(centroids\_small\_clusters)$
  if *small_cluster_x has high level of correlation with small_cluster_y* **then**
    $Cnew_j$ = small_cluster_x $\cup$ small_cluster_y
    update centroid
  end
  return $IDX, centroids$
End Function

been used in several research articles in the domain of indoor positioning (see section II) and other research areas, acquiring different levels of accuracy. Our proposal is compared with these clustering algorithms in terms of execution time and positioning accuracy. The hyperparameters (*Best configuration*) for *k* in k-NN were obtained from [4], [30], [31] and the *Eps* and *MinPts* for DBSCAN from [32].

Given that the proposed algorithm does not require to establish the number of clusters to form the groups, we executed FPC to get the *k* for k-Means and the initial value for C-Means. Due to that DBSCAN form the clusters based in the two parameters (*Eps* and *MinPts*) it is difficult to set a specific number of cluster. However, it is not a limitation to compare DBSCAN with FPC. Table I shows data sets with their number of fingerprints, APs and the number of maximum RSS values used to form the clusters.

Once the hyperparameters were obtained for each clustering algorithm, we executed them along with k-NN for each data set in order to get the positioning error (2D and 3D) and the execution time. The experiments were carried out using a computer with the following characteristics: Intel® Core™ i7-8700T @ 2.40GHz and 16 GB of RAM, the operating system is Windows 10, and the software used is Matlab.

TABLE I
PARAMETERS TO RUN FPC

Database	δ	γ	Num. Max. RSS
Wi-Fi data sets			
DSI 1	1369	157	2
DSI 2	576	157	3
LIB 1	576	174	1
LIB 2	576	197	1
MAN 1	14 300	28	3
MAN 2	1300	28	4
SIM	10 710	8	4
TUT 1	1476	309	1
TUT 2	584	354	2
TUT 3	697	992	1
TUT 4	3951	992	2
TUT 5	446	489	2
TUT 6	3116	652	4
TUT 7	2787	801	3
UJI 1	19 861	520	3
UJI 2	20 972	520	4
MINT 1	4973	11	3
BLE data sets			
UEXB 1	417	30	2
UEXB 2	552	30	2
UEXB 3	240	30	1

B. Results

We have defined three main parameters to carry out the comparison between the algorithms. The first one is ϵ_{2D} which represents the positioning error *2D* (meters), ϵ_{3D} the positioning error *3D* (meters) and τ the time in seconds of executing k-NN plus the clustering algorithm. In this research work we report the normalized values of the previous parameters, $\tilde{\epsilon}_{2D}$, $\tilde{\epsilon}_{3D}$, $\tilde{\tau}$. Furthermore, we have compare the execution time of each clustering algorithm without combining it with k-NN (see Figure 2).

Other parameters used in this research work are: δ, which represents the number of fingerprints in the training data set, γ the number of APs in the training data set, and ϕ is the number of clusters.

To run the FPC algorithm, we have used between 1 to 4 maximum RSS values to be compared in each fingerprint. However, this parameter can be different for each data set to reduce the time response or improve the positioning accuracy. Data sets like LIB 1 require one maximum RSS value to form the clusters, obtaining a positioning error of 2.68 m (baseline 2.49 m). However, other data sets require more than one maximum RSS values to group the most similar fingerprints into the cluster. For instance, in SIM, TUT 6, UJI 2 and MAN 2 needed the maximum *"Num. Max. RSS"* value, 4 RSSs, used to form the groups or clusters.

Table II shows the results after running each clustering algorithm with k-NN. In the first place, we have the number of fingerprints and APs of each training data set, then we have the results of executing k-NN with the *best configuration* provided in [4]. The next results are k-NN + k-Means, here we can observe that the *2D* and *3D* positioning error increased in comparison with the baseline. However, the execution time

TABLE II
COMPARISON k-NN AND CLUSTERING METHODS

Database	Baseline						k-NN+k-Means				k-NN+C-Means				k-NN+DBSCAN				k-NN+FPC			
	ϵ_{2D}	ϵ_{3D}		τ	$\tilde{\epsilon}_{2D}$	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}$	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}$	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}$	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}$	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\epsilon}_{3D}$	$\tilde{\tau}$
Wi-Fi data sets																							
DSI 1	3.791	3.791	5.906	1	1	1	69	1.231	1.231	0.455	13	2.602	2.602	0.450	253	1.410	1.410	0.757	69	1.216	1.216	0.286
DSI 2	3.804	3.804	3.953	1	1	1	126	1.280	1.280	0.277	6	5.89	5.89	3.68	124	1.364	1.364	0.285	126	1.326	1.326	0.198
LIB 1	2.475	2.478	17.688	1	1	1	16	1.148	1.151	0.150	16	1.101	1.116	0.313	3	1.109	1.197	0.484	16	1.071	1.083	0.127
LIB 2	2.265	2.267	34.719	1	1	1	16	1.398	1.443	0.108	16	1.237	1.272	0.237	52	2.329	2.743	0.032	16	1.403	1.723	0.106
MAN 1	2.057	2.057	60.172	1	1	1	48	1.111	1.111	0.159	43	1.105	1.105	0.353	290	1.717	1.717	2.671	48	1.102	1.102	0.087
MAN 2	1.856	1.856	6.891	1	1	1	24	1.162	1.162	0.102	24	1.110	1.110	0.231	1	1.000	1.000	1.091	24	1.106	1.106	0.129
SIM	2.411	2.411	94.688	1	1	1	28	1.033	1.033	0.037	28	1.043	1.043	0.097	609	1.354	1.354	0.960	28	1.070	1.070	0.061
TUT 1	4.232	4.446	107.063	1	1	1	86	1.405	1.713	0.084	4	1.149	1.167	0.692	74	1.872	2.481	0.042	86	1.932	2.154	0.021
TUT 2	7.804	8.095	1.094	1	1	1	139	1.844	2.139	3.028	13	2.502	3.941	3.185	130	1.771	2.191	0.671	139	1.384	1.626	0.728
TUT 3	8.167	8.552	39.375	1	1	1	120	1.082	1.110	0.312	2	1.149	1.231	1.492	40	1.916	2.524	0.069	120	1.502	1.691	0.085
TUT 4	5.07	5.398	713.531	1	1	1	571	1.211	1.223	0.526	2	1.259	1.467	1.427	187	2.535	2.564	0.237	571	1.874	2.335	0.050
TUT 5	5.254	5.259	78.344	1	1	1	158	1.126	1.210	0.063	2	1.250	1.398	0.813	50	1.839	2.323	0.021	150	1.570	1.736	0.017
TUT 6	1.901	1.908	283.672	1	1	1	1077	1.270	1.268	1.509	3	2.185	2.306	3.237	306	2.263	6.358	0.129	1076	3.162	3.167	0.331
TUT 7	2.065	2.24	251.547	1	1	1	946	1.103	1.132	1.099	17	1.583	1.822	1.948	227	2.707	3.621	0.206	896	2.214	2.869	0.190
UJI 1	6.173	6.556	261.797	1	1	1	1467	1.185	1.407	18.384	73	2.196	2.633	2.665	214	6.289	10.102	4.105	1467	1.427	1.781	0.714
UJI 2	5.603	6.089	1634.641	1	1	1	2972	1.317	1.284	4.486	62	1.656	1.601	1.215	205	7.053	4.604	0.796	2972	1.489	2.095	0.409
MINT1 1	2.46	2.46	35.438	1	1	1	8	1.017	1.017	0.146	8	0.977	0.977	0.205	251	1.163	1.163	0.554	8	1.037	1.037	0.116
BLE data sets																						
UEXB 1	3.439	3.534	0.469	1	1	1	51	1.065	1.084	0.400	35	1.210	1.225	5.430	2	5.866	5.841	0.800	51	1.434	1.442	0.300
UEXB 2	4.576	4.639	0.688	1	1	1	56	0.997	0.993	0.477	36	1.287	1.288	4.406	4	4.337	4.353	0.522	56	1.400	1.405	0.250
UEXB 3	7.393	7.55	0.156	1	1	1	19	0.918	0.947	1.202	4	1.011	1.020	4.607	24	1.988	2.034	2.704	19	1.200	1.242	1.002
Avg.				1	1	1		1.138	1.188	1.572		1.595	1.725	1.747		2.471	2.902	0.816		1.425	1.581	0.248

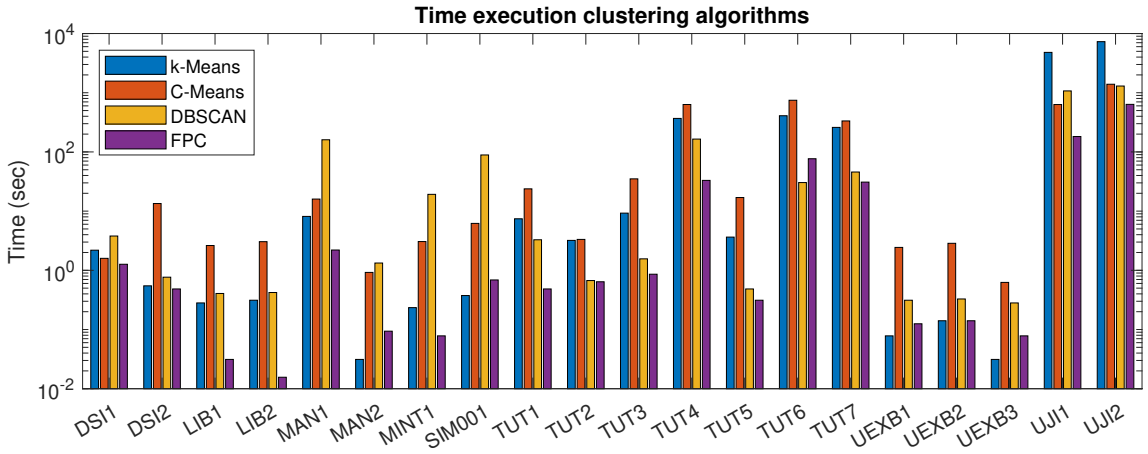

Fig. 2. Execution time clustering algorithm

is significantly reduced in most of the data sets with the exception of TUT 2, UJI 1 and UJI 2. For instance, in DSI 1 and DSI 2 the positioning error increased around 30%, but the execution time was reduced in more than 60% approximately.

Similarly, C-Means reduced the execution time in comparison with the baseline and increased the positioning error. However, the performance of this clustering algorithm is slightly worse than k-Means in most of the cases with the exception of LIB 1, LIB 2, MAN 1, MAN 2 and TUT 1 where the *2D* and *3D* positioning error was better than that of k-Means. DBSCAN also provides a better performance in terms of execution time in most of the data sets, but the positioning error increased in all the data sets with an average normalized *2D* ($\tilde{\epsilon}_{2D}$) and *3D* ($\tilde{\epsilon}_{3D}$) positioning error of 2.471 and 2.901 respectively. Despite the fact that the C-Means and DBSCAN are well-known and widely used, it is not appropriated to apply these algorithms as it is for fingerprinting. They need some modifications. That is the reason for the number of researchers proposed multiple changes to these algorithms to achieve better results when they are used in this research field.

In the case of k-NN + FPC, the average *2D* and *3D* positioning error increased $\approx$ 43% and 58% respectively in comparison with the baseline. With regards to k-NN + k-Means 25% ($\tilde{\epsilon}_{2D}$) and 33% ($\tilde{\epsilon}_{3D}$) approximately. However, FPC performs better

that C-Means, being the ($\tilde{\epsilon}_{2D}$) reduced in ≈11% and ($\tilde{\epsilon}_{3D}$) in ≈8%. Similarly, k-NN + FPC outperformed k-NN + DBSCAN in ≈58% ($\tilde{\epsilon}_{2D}$) and ≈55% ($\tilde{\epsilon}_{3D}$).

In order to provide an extended analysis, we applied this clustering algorithm to the aforementioned BLE data sets. The results are similar to that obtained in Wi-Fi fingerprinting databases. However, as these data sets are not as large as some of the Wi-Fi data sets used in this article, it is not highly relevant to use clustering algorithms on them unless required to detect some anomalies such as outliers.

The full execution time is significantly reduced in comparison with the baseline (≈ 75%), k-NN + k-Means (≈ 84%), k-NN + C-Means (≈ 85%) and k-NN + DBSCAN (≈ 70%).

Figure 2 shows the execution time of each clustering algorithm, i.e., the time that the algorithm takes to form the clusters and return the result. The *x-axis* represents the data sets used in this research work, and *y-axis* the average time to form the clusters. As we can see, FPC is faster than DBSCAN, k-Means and C-Means in 93% of the cases (approx.). For instance, k-Means in TUT 1 took 7.42 seconds to form the clusters, C-Means 23.78 seconds, DBSCAN 3.28 seconds, and FPC needed only 0.48 seconds to process 1476 fingerprints. k-Means were faster than FPC in UEXB 1, UEXB 3, MAN 2 and SIM.

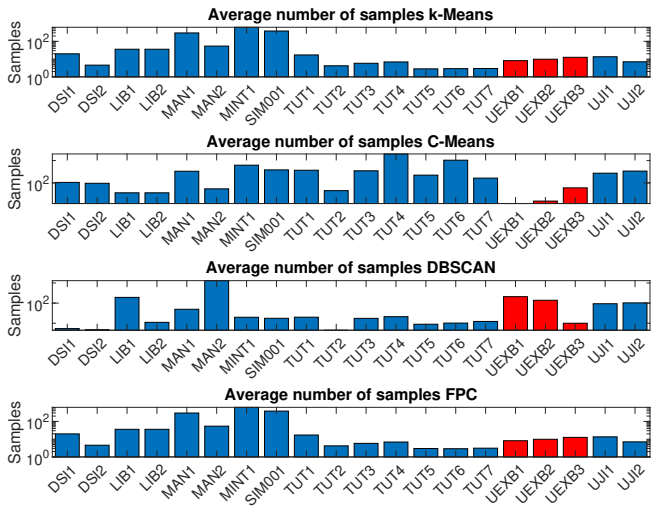

Fig. 3. Average number of fingerprints in each data set (Blue: WiFi Data sets, Red: BLE data sets)

Figure 3 shows the average fingerprints distribution in the clusters per data set. The *x-axis* represents the data sets used to test the algorithms, and *y-axis* is the average number of fingerprints in the clusters. Both k-Means and FPC share similar characteristics regarding the average number of fingerprints in the groups formed by these algorithms. However, it highly differs from C-Means and DBSCAN which have a different distribution of fingerprints between the cluster given their algorithms.

In the same fashion, Figure 4 depicts the standard deviation of the clusters in each data set, where the *x-axis* represents the data sets and the *y-axis* the standard deviation. Here we can observe that none of the clustering algorithms offer an equal distribution of fingerprints across the clusters.

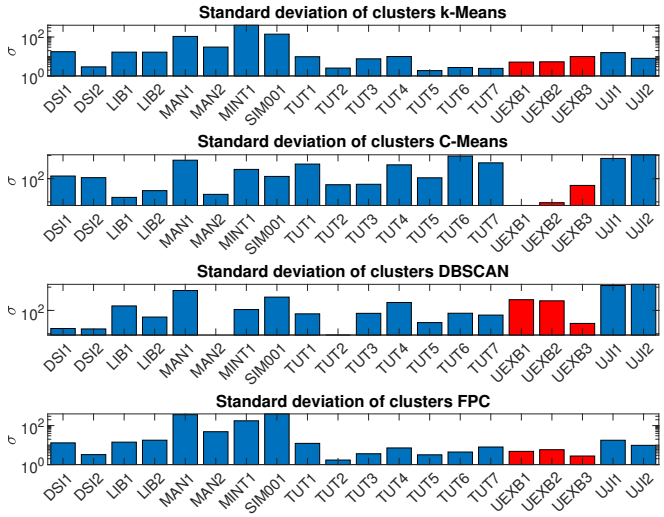

Fig. 4. Standard deviation of clusters in each data set (Blue: WiFi Data sets, Red: BLE data sets)

In fact, some data sets present a standard deviation similar to the average, which might be an indicator of a significant imbalance between classes, especially for C-Means and DBSCAN clustering algorithms. This behaviour should be avoided as the time required to obtain a position estimate will depend on the operational fingerprint and best-matched cluster. In some cases, the combination of a high average number of fingerprints and extremely low standard deviation indicate those cases where the clustering algorithm failed and only formed 1 cluster was generated.

V. Discussion

Considering that there is a trade-off between positioning accuracy and computation time, it is complex to develop lightweight algorithms which also provide a high level of accuracy. As discussed in the previous section (see Section IV), the traditional algorithms minimized the execution time in most of the cases, but the positioning error increased in those data sets or at least in most of them. In section II, we can see that different authors have offered various modifications to traditional algorithms to reduce the positioning error. However, these modifications add the cost additional processes, which increase the algorithm complexity, and therefore, the computational time.

Similar to other algorithms, FPC has both advantages and disadvantages. For instance, this clustering algorithm offers the lowest execution time if we compare it with DBSCAN, C-Means and k-Means. Therefore, FPC uses less computational resources than the compared traditional clustering algorithms in more than 93% of the cases (see Section IV-B). Similarly, if FPC is combined with k-NN, the full execution time (clustering + positioning estimation) is much lower than the other compared algorithms. However, the positioning error has slightly increased in all the data sets.

Although the proposed algorithm does not provide the lowest positioning error in all the cases, it is more efficient than C-Means and DBSCAN in terms of computational load as it is more than 66% faster and the positioning error is reduced by an 8%. However, the positioning error obtained with k-Means clustering is still better than FPC, C-Means and DBSCAN.

The results obtained after applying the proposed FPC with BLE data demonstrates that this algorithm can be used with other technologies with a similar performance than using it in Wi-Fi fingerprinting. However, it is preferable to use FPC in large databases where a fast response is required.

According to the results, FPC presents an acceptable performance in many of the data sets, despite their heterogeneity. Nevertheless, in some data sets, the 3D error is two or three times higher than the baseline and k-Means. It could be because of the methodology of data collection, for example, in systematic data collection (e.g., LIB 1 and LIB 2), the positioning error was lower than crowdsourced data collection (e.g., TUT 1, TUT 3, TUT 4, and UJI 2). Although FPC takes into account only the maximum RSS values to join samples with similar characteristics, the fluctuation of RSS and outliers can affect the positioning error. These adverse effects have to be examined in depth in future modification of this algorithm to enhance the positioning accuracy without a significant increment in the computation load.

Despite the fact that the efficiency of FPC in crowdsourced data sets is lower than other data sets, it could be used in devices where the computational efficiency is a must, and they do not require a high level of accuracy. Generally, low-profile devices require high efficient and lightweight algorithms to avoid the overload of computational resources and keep the battery life. Considering this fact, many mobile devices (including wearables and smartphones) use some positioning and localisation services, it is indispensable to develop or modify existing algorithms to provide computing efficiency at all the positioning steps.

VI. Conclusions

In this work, we presented a new lightweight clustering algorithm for Wi-Fi Fingerprinting, namely FPC. Unlike some traditional algorithms, which used different distance metrics such as euclidean distance, city block (Manhattan distance) or other methods, FPC is mainly based on the maximum RSS values to form the clusters. Therefore, if the maximum RSS values in different samples are in the same feature (AP), these samples are used to generate a cluster. Given that there could be small clusters (i.e., the number of samples in a cluster is less than the average number of samples in all the clusters), they are merged using the coefficient correlation matrix.

This new clustering algorithm outperforms DBSCAN, C-means and k-Means in terms of computational load, minimising the time to generate the clusters. Hence, FPC is faster than k-Means in $\approx 92\%$, $\approx 75\%$ in comparison with C-Means and $\approx 66\%$ in contrast with DBSCAN. However, the positioning accuracy was affected in all the data sets, increasing the positioning error (see section IV-B).

Given the computational efficiency of the proposed algorithm, it can be used in low-profile devices where it is indispensable lightweight algorithms. Therefore, IPSs can implement the proposed clustering algorithm if there is no stringent requirement of position accuracy.

Future work will analyse a method to identify outliers in the data sets in order to reduce the error. It will be useful in crowdsourced data sets where there are hundreds of outliers exist. This method has to be efficient enough to avoid the computational overload in the proposed clustering algorithm.

References

[1] K. Huang, K. He, and X. Du, "A hybrid method to improve the ble-based indoor positioning in a dense bluetooth environment," *Sensors*, vol. 19, no. 2, p. 424, 2019.

[2] Y. Cheng and T. Zhou, "Uwb indoor positioning algorithm based on tdoa technology," in *2019 10th International Conference on Information Technology in Medicine and Education (ITME)*, IEEE, 2019, pp. 777–782.

[3] Y. Li, Z. Ghassemlooy, X. Tang, *et al.*, "A vlc smartphone camera based indoor positioning system," *IEEE Photonics Technology Letters*, vol. 30, no. 13, pp. 1171–1174, 2018.

[4] J. Torres-Sospedra, D. Quezada-Gaibor, G. Mendoza Silva, *et al.*, "New cluster selection and fine-grained search for k-means clustering and wi-fi fingerprinting," Jun. 2020, pp. 1–6.

[5] P. A. Karegar, "Wireless fingerprinting indoor positioning using affinity propagation clustering methods," *Wireless Networks*, vol. 24, no. 8, pp. 2825–2833, 2018.

[6] A. Anuwatkun, J. Sangthong, and S. Sang-Ngern, "A diff-based indoor positioning system using fingerprinting technique and k-means clustering algorithm," in *2019 16th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, IEEE, 2019, pp. 148–151.

[7] K. Wang, X. Yu, Q. Xiong, *et al.*, "Learning to improve wlan indoor positioning accuracy based on dbscan-krf algorithm from rss fingerprint data," *IEEE Access*, vol. 7, pp. 72 308–72 315, 2019.

[8] L. Klus, D. Quezada-Gaibor, J. Torres-Sospedra, *et al.*, "Rss fingerprinting dataset size reduction using feature-wise adaptive k-means clustering," in *2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, 2020, pp. 195–200.

[9] 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.

[10] 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.

[11] J. Ren, Y. Wang, C. Niu, *et al.*, "A novel clustering algorithm for wi-fi indoor positioning," *IEEE Access*, vol. 7, pp. 122 428–122 434, 2019.

[12] W. Xue, X. Hua, Q. Li, *et al.*, "Improved clustering algorithm of neighboring reference points based on knn for indoor localization," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 2018, pp. 1–4.

[13] Y. Tao and L. Zhao, "A novel system for wifi radio map automatic adaptation and indoor positioning," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 683–10 692, 2018.

[14] H. Cui and K. Liu, "Indoor positioning and fingerprint updating based on affinity propagation clustering," in *2018 Eighth International Conference on Instrumentation Measurement, Computer, Communication and Control (IMCCC)*, 2018, pp. 226–230.

[15] 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, p. 2300, May 2019.

[16] S. Sendra, M. Garcia-Pineda, C. Turro, *et al.*, "Wlan ieee 802.11 a/b/g/n indoor coverage and interference performance study," *International Journal on Advances in Networks and Services*, vol. 4, Jan. 2011.

[17] S. Garcia-Villalonga and A. Perez-Navarro, "Influence of human absorption of wi-fi signal in indoor positioning with wi-fi fingerprinting," in *2015 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2015, pp. 1–10.

[18] J. Torres-Sospedra, P. Richter, A. Moreira, *et al.*, "A comprehensive and reproducible comparison of clustering and optimization rules in wi-fi fingerprinting," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2020.

[19] 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.

[20] 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 *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2014, pp. 261–270.

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

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] A. Moreira, I. Silva, and J. Torres-Sospedra, *The DSI dataset for Wi-Fi fingerprinting using mobile devices*, version 1.0, Zenodo, Apr. 2020.

[30] A. Moreira, M. J. Nicolau, I. Silva, *et al.*, *Wi-Fi Fingerprinting dataset with multiple simultaneous interfaces*, version 1.0, [available on-line] https://doi.org/10.5281/zenodo.3342526, Zenodo, Sep. 2019.

[31] F. J. Aranda, F. Parralejo, F. J. Álvarez, *et al.*, "Multi-slot ble raw database for accurate positioning in mixed indoor/outdoor environments," *Data*, vol. 5, no. 3, 2020.

[32] D. Quezada-Gaibor, L. Klus, J. Torres-Sospedra, *et al.*, "Improving dbscan for indoor positioning using wi-fi radio maps in wearable and iot devices," in *Twelfth International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT)*, Oct. 2020.