Improving DBSCAN for Indoor Positioning Using Wi-Fi Radio Maps in Wearable and IoT Devices

Darwin Quezada-Gaibor[*,†], Lucie Klus[†,*], Joaquín Torres-Sospedra[‡,*],
Elena Simona Lohan[†], Jari Nurmi[†] 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—IoT devices and wearables may rely on Wi-Fi fingerprinting to estimate the position indoors. The limited resources of these devices make it necessary to provide adequate methods to reduce the operational computational load without degrading the positioning error. Thus, the aim of this article is to improve the positioning error and reduce the dimensionality of the radio map by using an enhanced DBSCAN. Moreover, we provide an additional analysis of combining DBSCAN + PCA analysis for further dimensionality reduction. Thereby, we implement a post-processing method based on the correlation coefficient to join "noisy" samples to the formed clusters with Density-based Spatial Clustering of Applications with Noise (DBSCAN). As a result, the positioning error was reduced by 10% with respect to the plain DBSCAN, and the radio map dimensionality was reduced in both dimensions, samples and Access Points (APs).

Index Terms—Clustering; DBSCAN; PCA; RSS; Wi-Fi fingerprinting.

I. INTRODUCTION

Nowadays, Internet of Things (IoT) technologies have become indispensable for many companies in different domains such as telecommunications [1], transport [2] and e-health [3], giving rise to new and complex networks with millions of devices connected through the Internet. Many more services (including positioning and localization) are offered to the end-users at the expense of a fast increase in network traffic and greater consumption of computational resources. An increasingly attractive sub-category of IoT technologies is the category of wearables, i.e., body-worn or hand-held devices, which can serve various functionalities from health status monitoring to contact tracing and activity detection. Many IoT devices and, in particular, many wearable devices are heavily used in indoor scenarios and are more and more requiring some indoor localization features to enable a variety of Location Based Services (LBSs).

Wi-Fi fingerprinting technique is one of the most broadly used techniques for Indoor Positioning Systems (IPSs), thanks to the fact that many Wi-Fi routers and APs are already deployed in public and private areas. However, it is not efficient

enough to run it in some devices, such as power-constraint wearables. Thus, some authors have proposed methods based on clustering (e.g. k-means, affinity propagation and c-means) [4], [5] or dimensionality reduction [6]–[9] in order to reduce the datasets or at least the search area, and therefore to decrease the computational load. Whereas the main objective of dimensionality reduction in Wi-Fi fingerprinting is to reduce the vector dimensionality (number of APs) by applying linear or non-linear transformations (e.g. PCA, GDA, LDA, t-SNE) [10]. Both methods can be applied –separately or jointly– to Wi-Fi fingerprinting to reduce the radio map in both samples and feature dimensions.

DBSCAN [11]–[13] is a clustering method used to split the radio map into high-density and low-density clusters, dividing it into n non-overlapping reduced radio maps. Then, in the operational phase, the search has two steps. A coarse search to identify the closest centroid –the centroids are computed for each cluster in this work, as they are not provided in DBSCAN– and a fine-grained search to obtain the closest reference fingerprints –within the selected closest cluster– to the operational one. This clustering method requires two parameters, *Eps*, which is the distance used to form the neighbourhood between samples, and *MinPts*, which determines the minimum number of samples to form a cluster [14]. In case a point(sample) is not part of a cluster, it will be considered as noise. Many fingerprints can be considered noisy when DBSCAN clusters the radio map, which might affect the position estimation.

In this article, we introduce a new DBSCAN post-processing method, which is applied when the number of noise samples exceeds 10% of the total size of the dataset. Thus, we establish some rules devoted to group noise points in clusters with a higher level of similarity or correlation.

The main contributions of this article are the following:

- A new variant to enhance the position estimation when DBSCAN is used in Wi-Fi fingerprinting.
- Dimensionality reduction of the Wi-Fi fingerprinting datasets by combining DBSCAN and Principal Component Analysis (PCA).
- Reduction in the time search and position estimation time in the online phase of Wi-Fi fingerprinting.

This article is organized by the following sections. Section II

provides a general overview of clustering and Wi-Fi finger-printing and related work. Section III describes the proposal modification of DBSCAN clustering and its integration with Wi-Fi fingerprinting. Section IV describes the procedure to execute the experiment and its results. Section V provides the conclusions raised from the main findings.

II. RELATED WORK

Wi-Fi fingerprinting is considered as one of the most used techniques for indoor positioning and localization [15]. Due to that Wi-Fi APs are already deployed in multiple environments, avoiding the cost of deploying new positioning technologies. However, this solution requires to have datasets with hundreds or thousands of Received Signal Strength (RSS) measurements. Some researchers use unsupervised ML methods to classify similar fingerprints into clusters to reduce the search area in the online phase of Wi-Fi fingerprinting. The most used algorithms are k-means, fuzzy c-means (soft k-means), k-medoids, affinity propagation, among others. A few relevant approaches are detailed below.

Zhang *et al.* [16] proposed an algorithm based on hierarchical classification and k-means. The improved k-means is used to divide the indoor environment into overlapping zones. In contrast to k-means, which performs a partitioning of the data space into non-overlapping Voronoi cells, this new algorithm allows having a Wi-Fi fingerprint in more than one cluster. As a result, they reduced the execution time (less than 1 second) and improved the position accuracy with a low average positioning error of 1.2m.

Abusara *et al.* [6] use a different method to reduce the radio map. This method is devoted to eliminating non-relevant AP, reducing the positioning error. The authors use the fast orthogonal search (FOS) method to identify relevant information in the radio map and keep the main characteristics of the dataset. Additionally, they propose a modified FOS (mFOS) which is oriented to estimate the user position instead of compressing the radio map. As a result, both FOS and modified FOS provide better performance and lower positioning error than PCA.

Jia *et al.* [7] based their work in supervised ML, using Gaussian Process Manifold Kernel Dimension Reduction (GPMKDR) in the offline phase to detect and extract the most relevant features in the radio map. Consequently, the authors got a mean positional error of 1.13m, which is lower than when the PCA-based method is used.

López-de-Teruel *et al.* [8] evaluate the quality of the radio map using dimensionality reduction techniques, and propose two new visualization methods. The dimensionality reduction or data compression is mainly based on three well-known methods PCA, t-SNE and Linear Discriminant Analysis (LDA). As a result, they obtained a natural visualization of the radio map, including overlapping zones and outliers.

The analysed research literature shows the importance of dimensionality reduction for indoor positioning and IoT. It provides efficient use of the computational resources and the improvements in the execution time. It is important to highlight that these analyses were done for WLAN-fingerprinting.

III. PROPOSED DBSCAN VARIANTS

DBSCAN clustering algorithm is used to find groups of samples of different shapes [9], in particular, to find high-density zones (clusters) and to separate them from low-density zones. In contrast with k-means clustering, DBSCAN doesn't need a predefined number of clusters, it forms the clusters based on two parameters: *Eps*, which determines the distance to form the neighbourhood and *MinPts* which is the minimum number of samples to create a cluster. Once the clusters are generated, there are some samples labelled as noise, and they are excluded from the clusters.

In this work, we propose an improved DBSCAN in order to minimize the error in the position estimation when DBSCAN is used, and we combine this approach with PCA for further reduction in the radio map dimensions, providing computational efficiency. Additionally, DBSCAN clustering is combined with k-nearest neighbors (kNN) as main core IPS to estimate the user position. Thus, DBSCAN is applied in the offline phase of Wi-Fi fingerprinting positioning technique, in such a way, that the operational fingerprint will be compared with a specific cluster with similar characteristics in the online phase.

When DBSCAN is applied to Wi-Fi fingerprinting the clustered radio map may contain many samples denoted as noise, which might degrade the accuracy of the position estimation in some cases. Under optimal conditions, DBSCAN is capable of detecting and excluding outliers from the clusters. However, due to the heterogeneity of the datasets the cluster distribution is not homogeneous, and therefore some relevant samples might be excluded from them. Considering this as a weakness, we propose the following DBSCAN post-processing method.

A. Step one - Establish the percentage of "noise" samples allowed for each dataset

The first step is to establish the percentage of noise (threshold) accepted in the analysed dataset. In general, noise samples are represented by 0 or -1 in the vector with the cluster indexes when DBSCAN is applied. The selected threshold may differ from one dataset to another.

B. Step two - Compute the correlation coefficient matrix

The correlation coefficient matrix is computed if the percentage of noise fulfils the condition $\% of noise \geq threshold$, where the percentage of noise is computed with regard to the total number of samples and the threshold is established in step one. This correlation coefficient matrix is computed from the distance matrix provided by DBSCAN. Thus, the strength of the relationship between each sample may be known.

C. Step three - Joining "noise" samples to the formed clusters

The noise samples are joined to a specific cluster in case they meet the condition $Correlation Coefficient > 0.10$. If the condition is true, we search a labelled sample (no noise) with a higher level of correlation between the two samples.

When the sample is found, the noise point is joined to the same cluster. This process is repeated with all the noise samples.

Algorithm 1 graphically describes the process mentioned in the previous three steps.

Algorithm 1: DBSCAN post-processing

1 Input: IDX, distance matrix
2 Output: IDX'
3 IDX' ← IDX (cluster indices)
4 if $\%of\,noise \geq threshold$ **then**
5   ccm ← Compute the correlation coefficient matrix from the distance matrix
6   vector ← Sum all the coefficients by row
7   **for** $i \leftarrow 1$ **to** *length IDX'* **do**
8   **if** *IDX'(i) == 0* **then**
9   Compute the percentage of vector(*i*) with respect to the maximum vector's value
10   **if** *percent > 0.10* **then**
11   j ← 1
12   **while** $j \leq length\ IDX'$ **do**
13   sc ← sort ccm(i,:) DESC
14   **if** *IDX'(sc(j)) != 0* **then**
15   IDX'(i) ← IDX'(sc(j))
16   break
17   **else**
18   j ← j + 1
19   **end**
20   **end**
21   **end**
22   **end**
23   **end**
24 end

IV. EXPERIMENTS AND RESULTS

A. Setup and Procedure

This experiment is performed by using 12 Wi-Fi finger-printing radio maps from Tampere University [17]–[19], University Jaume I [20], University of Mannheim [21], [22], and University of Minho [23]. Supplementary materials, with method implementation and dataset explanation, are available at Zenodo [24] for research reproducibility.

The current analysis combines kNN, DBSCAN post-processing method and PCA in order to reduce the dimensionality of the radio maps and estimate the user position.

To compare our approach we run a plain kNN with $k = 1$, positive data representation, and cityblock distance metric (see [25]). This simple configuration is used as the baseline for the analysis performed in this paper.

The hyperparameters for kNN, and DBSCAN are listed in Table I. These hyperparameter values provide the best error in the position estimation for every dataset. Additionally, the table shows the data representation (powed, positive, and exponential) used for each dataset [25], the k value for kNN, the values of *Eps* and *MinPts* for DBSCAN. It is important to

highlight that there is no data normalization or standardization applied in plain DBSCAN. However, data normalization is applied for the combination of DBSCAN with PCA.

TABLE I
OPTIMAL PARAMETERS TO RUN DBSCAN

Database	Data Rep.	k (kNN)	Plain DBSCAN		with PCA 90%	
			Eps	*MinPts*	*Eps*	*MinPts*
DSI 1	Powed	11	0.170	3	0.005 00	2
DSI 2	Positive	9	150	2	0.015 00	2
LIB 1	Positive	11	160	4	0.010 00	2
LIB 2	Positive	9	66	2	0.010 00	2
MAN 1	Exponential	11	0.031	3	0.006 00	3
MAN 2	Exponential	11	0.033	3	0.001 11	3
SIM	Exponential	11	0.012	2	0.000 60	2
TUT 1	Positive	3	290	2	0.009 21	2
TUT 2	Powed	1	0.330	2	0.005 10	2
TUT 3	Positive	3	290	2	0.021 00	2
TUT 4	Positive	3	135	2	0.009 00	2
TUT 5	Positive	3	180	2	0.004 10	2

* DSI and SIM - U. of Minho, LIB - U. Jaume I, MAN - U. of Mannheim, TUT - Tampere U.

Once the data normalization is applied, we use PCA to reduce the dataset. Furthermore, with the aim of keeping most of the variance in the dataset, we chose 90% of variance explained. As a result, we obtained the number of principal components which satisfy the percentage of variance required.

The next step is to determine *Eps* and *MinPts*. Thus, to have a better approximation of the optimal *Eps* value, we use the algorithm proposed by [26] to find the elbow point, then multiple values of *Eps* and *MinPts* were tested to achieve the lowest positioning error for each dataset.

Finally, we apply the plain kNN, kNN with DBSCAN, kNN with DBSCAN and the post-processing method, kNN with DBSCAN, the post-processing method and PCA 90%.

To run the experiments, we used a computer with the following characteristics: Intel® Core™ i7-8700T @ 2.40GHz and 16 GB of RAM, the operating system is Fedora Linux and the software used is Octave v5.0.2.

B. Results

To analyse the results obtained through this experiment, we use the parameters and notation shown in Table II:

TABLE II
PARAMETERS AND NOTATION

δ	is the number of samples in the dataset
γ	is the number of APs
$\tilde{\gamma}$	is the reduced number of APs
ϵ_{2D}	represents the mean 2D positioning error
τ	is the execution time required to estimate the position
$\tilde{\epsilon}_{2D}$	represents the normalized 2D positioning error. The benchmark is the result of plain kNN.
$\tilde{\tau}$	is the normalized execution time
ψ	represents the number of clusters
ϕ	is the number of samples labelled as noise

Table III shows the main results of dimensionality reduction, the application of the post-processing method, execution time,

TABLE III
MAIN RESULTS OF APPLYING DBSCAN IN DIFFERENT DATASET

Database	Plain kNN						DBSCAN				DBSCAN MOD				DBSCAN + PCA 90%				
	δ	γ	ϵ_{2D}	τ	$\tilde{\epsilon}_{2D}$	$\tilde{\tau}$	ψ	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\tau}$	ψ	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\tau}$	$\tilde{\gamma}$	ψ	ϕ	$\tilde{\epsilon}_{2D}$	$\tilde{\tau}$
DSI 1	1369	348	4.95	14.03	1	1	252	13	1.08	0.002	252	13	1.08	0.022	64	297	12	3.83	0.009
DSI 2	576	348	4.95	5.90	1	1	123	140	1.05	0.140	123	30	1.06	0.113	61	117	37	3.00	0.028
LIB 1	576	3120	3.01	53.97	1	1	3	0	0.91	0.394	3	0	0.91	0.401	31	67	9	1.38	0.024
LIB 2	576	3120	4.02	56.57	1	1	51	378	1.31	0.026	51	36	1.14	0.046	55	73	19	1.27	0.030
MAN 1	14300	460	2.82	186.19	1	1	289	12181	1.25	0.002	289	4430	1.03	0.019	16	1	0	3.28	0.984
MAN 2	1300	460	2.47	17.54	1	1	118	0	0.91	0.022	118	0	0.91	0.022	8	99	10	4.34	0.073
SIM	10710	8	3.14	303.21	1	1	608	9224	1.04	0.001	608	1084	0.81	0.001	5	2	20	3.03	0.834
TUT 1	1476	490	8.61	22.56	1	1	73	431	0.92	0.040	73	0	0.98	0.036	51	57	139	2.64	0.084
TUT 2	584	176	12.66	3.27	1	1	129	100	1.09	0.018	129	0	0.90	0.014	56	111	0	4.10	0.019
TUT 3	697	3951	8.92	98.33	1	1	39	461	1.75	0.015	39	4	1.46	0.030	108	40	1	3.60	0.014
TUT 4	3951	697	6.10	99.37	1	1	186	2974	2.11	0.022	186	24	1.48	0.030	188	64	154	5.73	0.023
TUT 5	446	982	6.39	14.32	1	1	49	216	1.51	0.046	49	64	1.63	0.079	49	26	43	8.82	0.072
Average					1	1			1.24	0.061			1.12	0.068				3.75	0.183

and error in the positioning estimation of the four above-mentioned methods.

The first group in the table of results shows the parameters used to execute the plain kNN. Here the dataset is used in its original size without modifications and normalization, these results are used as the baseline for our analysis.

The second group shows the results of executing DBSCAN and kNN. Here we can see that the error in the position estimation ($\tilde{\epsilon}_{2D}$) increased with respect to the baseline in most of the cases. However, the matching time and position estimation time ($\tilde{\tau}$) decreased significantly in the online phase of Wi-Fi fingerprinting, after using DBSCAN clustering.

The third group is the modified DBSCAN or post-processing method + kNN. Here we can see the error in the position estimations is reduced by 10% (approx.) in comparison to the plain DBSCAN, but the time required to search the closest fingerprints and estimate the position is slightly increased, yet it is still significantly lower than the time required in case of using the plain kNN.

The fourth group is the combination of the modified DBSCAN or post-processing method + PCA 90%. The results of the positioning error are considerably higher in some datasets such as TUT 4 and TUT 5. However, the number of APs were "compressed" from 697 to 188 and from 982 to 49 in case of TUT 4 and TUT 5, respectively. This represents a considerable reduction of the dataset.

Regarding the formed clusters, we can observe that the distribution throughout the clusters is not equal in all the datasets, obtaining clusters of different sizes. Fig. 1 shows the distribution of clusters in TUT 2 training dataset. The x-axis represents the number of clusters and the y-axis the number of samples assigned to the cluster. The first plot shows the formed clusters after applying DBSCAN without any modification. The second plot (middle graph) shows how the noise samples are redistributed throughout the remaining clusters when we apply DBSCAN post-processing. Finally, the last plot shows the distribution of clusters after applying DBSCAN post-processing and PCA.
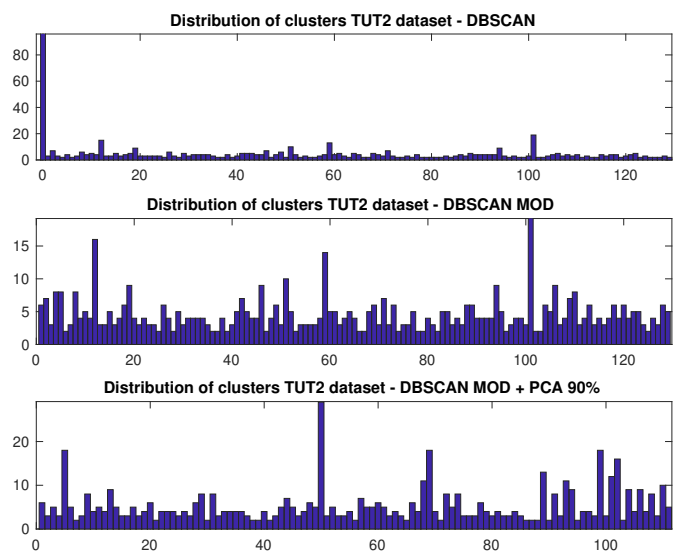

Fig. 1. TUT2 - Distribution of clusters after running DBSCAN, DBSCAN post-processing method and DBSCAN + PCA

C. Discussion

After conducting a search in Web Of Science with the following query *"TS=(DBSCAN AND indoor AND (position* OR localization OR location OR tracking))"*, we can notice that only a few researchers are working with DBSCAN and Wi-Fi fingerprinting for IPS (13 results). Their work coincides in asserting that DBSCAN is an efficient clustering algorithm to detect outliers in datasets. Additionally, if we discard the noise samples, we can see a reduction in the dimensionality of the radio map (reducing the number of samples).

DBSCAN, as other clustering methods, does not guarantee that all reference samples are equally distributed. Nevertheless, in some cases, we have detected that the original method tends to include a very large number of samples in just one cluster (see Fig.1 top), which is not computationally effective, thus, the need of improving it.
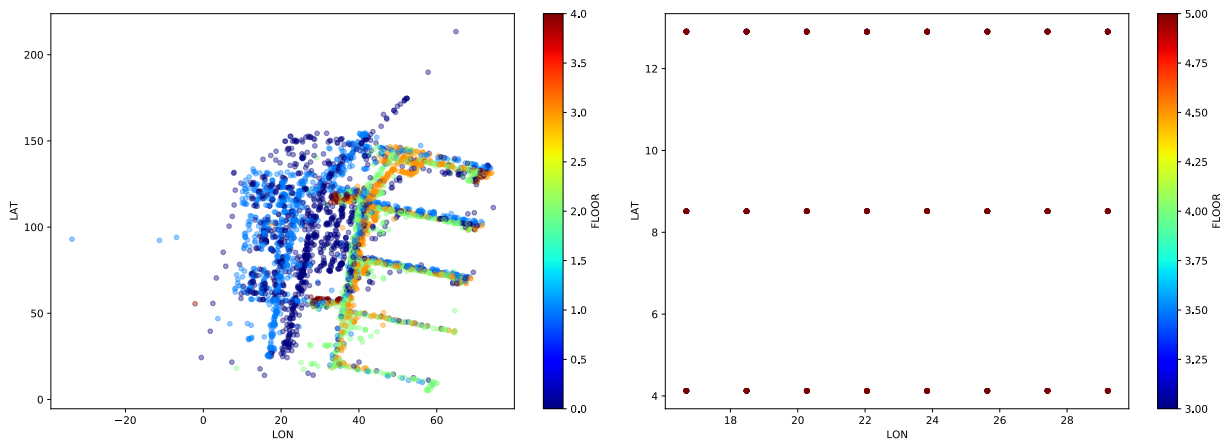
Fig. 2. TUT4 and LIB1/LIB2 dataset – training points representation

After applying the proposed post-processing method or DBSCAN modification, we can observe that the results obtained are slightly better than while using a plain DBSCAN. The error in the position estimation decreased by almost 10% after applying our method and injecting some noisy samples to the clusters. In the operational phase, the computational time is not significantly altered with respect to the original DBSCAN. Resulting improvements in the position estimation plus the reduction in the processing time could be good enough for some wearable devices with intermediate capabilities. However, we consider that this research line needs further improvements if we target to use it in very low profile devices.

Although DBSCAN provides a lower matching time and position computation time compared with the plain kNN in the online phase, it might require a large quantity of time to form the clusters and then, to compute the correlation coefficient matrix in the offline stage. This processing time is especially long in large datasets, i.e. those covering large multi-building and/or multi-floor operational areas. In our experiments, clustering of the largest datasets took more than 10 hours. Thus, the proposed method is valid in those radio maps which remain unaltered for a large period of time as it is not feasible if the radio map is regularly (hourly or daily) updated.

Regarding the variation in the results obtained after the post-processing method, we can observe that the error in the position estimation does not change significantly after the dimensionality reduction in some datasets. This is the case of the LIB 1 and LIB 2 (Fig. 2 right), the result was expected due to its distribution and the methodology of taking samples. However, we expected a better performance for TUT 4 (Fig. 2 left) and TUT 5 since DBSCAN is widely used to detect outliers and exclude them from the cluster, but the error increased approximately 8 times in comparison with the plain kNN and in both implementations of DBSCAN. Here it is important to mention that the methodology used to find the optimal parameters for *Eps* and *MinPts* is very important to avoid discarding useful samples.

V. CONCLUSIONS

This article provides a novel DBSCAN post-processing method to be used in IPS, in order to reduce the error in the position estimation when DBSCAN clustering is applied in Wi-Fi fingerprinting radio maps. Moreover, our method helps to reduce the dataset dimensionality by keeping relevant samples. This post-processing method is applied in the offline phase of Wi-Fi fingerprinting to join important samples denoted as noise to the formed clusters. This method is based on the correlation coefficient, which is needed to determine in which cluster is the sample with a higher level of relationship.

As a result of the experiment, we obtained a reduction of approximately 10% in the positioning error compared with the original DBSCAN. Also, the matching time and position estimation time in the online phase of Wi-Fi fingerprinting is significantly reduced than when using a plain kNN. However, the time used to form the clusters and then to compute the correlation coefficient matrix is considerably high in the offline phase, which should be considered in the implementation of DBSCAN.

Additionally, we combined the proposed post-processing method with PCA analysis for further reduction of the dimensionality of the radio map. Although the radio map dimension was considerably reduced, the positioning error increased in all the cases.

The results obtained of the positioning error and dimensionality reduction in datasets permit to use this method in middle profile IoT and wearable devices with the same characteristics when a high level of positioning accuracy is not necessary.

To sum up, we consider this implementation as a good starting point to work with DBSCAN for IPS on power-constraint wearables due to the fact that only a few researchers are working with the exposed combination for IPS (to our best knowledge) and it has shown promising first results. The next step is to research new ways to decrease the execution time of DBSCAN post-processing method in the offline phase and reduce the error when the dimension of the dataset is compressed by using PCA analysis. For the research reproducibility, we have also provided the link to all datasets on Zenodo [24].

REFERENCES

[1] H. Rahimi, A. Zibaeenejad, and A. A. Safavi, "A novel iot architecture based on 5g-iot and next generation technologies," in *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, 2018, pp. 81–88.

[2] S. Vidyasagaran, S. R. Devi, A. Varma, *et al.*, "A low cost iot based crowd management system for public transport," in *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017, pp. 222–225.

[3] K. Monteiro, É. Rocha, É. Silva, *et al.*, "Developing an e-health system based on iot, fog and cloud computing," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*, 2018, pp. 17–18.

[4] T. Wang, T. Sui, X. Liu, *et al.*, "Wifi positioning algorithm in tunnel based on fuzzy c-means clustering and knn algorithm," in *2019 Chinese Automation Congress (CAC)*, 2019, pp. 567–571.

[5] Y. Zhong, F. Wu, J. Zhang, *et al.*, "Wifi indoor localization based on k-means," Jul. 2016, pp. 663–667.

[6] A. Abusara, M. Hassan, and M. Ismail, "Rss fingerprints dimensionality reduction in wlan-based indoor positioning," Apr. 2016, pp. 1–6.

[7] B. Jia, B. Huang, H. Gao, *et al.*, "On the dimension reduction of radio maps with a supervised approach," in *2017 IEEE 42nd Conference on Local Computer Networks (LCN)*, 2017, pp. 199–202.

[8] P. López-de-Teruel, Ó. Cánovas, and F. J. García Clemente, "Using dimensionality reduction techniques for refining passive indoor positioning systems based on radio fingerprinting," *Sensors*, vol. 17, Apr. 2017.

[9] M. Ester, H.-P. Kriegel, J. Sander, *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96, Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[10] A. Sarveniazi, "An actual survey of dimensionality reduction," *American Journal of Computational Mathematics*, vol. 04, pp. 55–72, Jan. 2014.

[11] S. Li, "An improved dbscan algorithm based on the neighbor similarity and fast nearest neighbor query," *IEEE Access*, vol. 8, pp. 47 468–47 476, 2020.

[12] S. Chen, X. Liu, J. Ma, *et al.*, "Parameter selection algorithm of dbscan based on k-means two classification algorithm," *The Journal of Engineering*, vol. 2019, no. 23, pp. 8676–8679, 2019.

[13] 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.

[14] J. Dromard, G. Roudière, and P. Owezarski, "Online and scalable unsupervised network anomaly detection method," *IEEE Transactions on Network and Service Management*, vol. 14, no. 1, pp. 34–47, 2017.

[15] S. Xia, Y. Liu, G. Yuan, *et al.*, "Indoor fingerprint positioning based on wi-fi: An overview," *ISPRS International Journal of Geo-Information*, vol. 6, p. 135, Apr. 2017.

[16] C. Zhang, N. Qin, Y. Xue, *et al.*, "Received Signal Strength-Based Indoor Localization Using Hierarchical Classification," *SENSORS*, vol. 20, no. 4, Feb. 2020.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] D. Quezada-Gaibor, L. Klus, J. Torres-Sospedra, *et al.* (Aug. 2020). Supplementary Materials for "Improving DBSCAN for Indoor Positioning using Wi-Fi radio maps in wearable and IoT devices", [Online]. Available: https://zenodo.org/deposit/3983634.

[25] J. Torres-Sospedra, R. Montoliu, S. Trilles, *et al.*, "Comprehensive analysis of distance and similarity measures for Wi-Fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.

[26] V. Satopaa, J. Albrecht, D. Irwin, *et al.*, "Finding a "kneedle" in a haystack: Detecting knee points in system behavior," in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 166–171.