Accurate and Efficient Wi-Fi Fingerprinting-Based Indoor Positioning in Large Areas

Moises Ramires*, Joaquín Torres-Sospedra†, and Adriano Moreira†
**Centro de Computação Gráfica*, Guimarães, Portugal
† *Algoritmi Research Center*, *University of Minho*, Guimarães, Portugal

***Abstract*—The core of fingerprinting is based on the uniqueness of the RF signature in a given location over time. In the offline phase, the fingerprints –the set of RSSI values from different anchors– are collected at given locations generating a radio map. In the online phase, a matching algorithm retrieves the most similar fingerprints from the radio map and computes the position estimate for every operational fingerprint. However, computing the similarities to all the samples in the radio map may be inefficient and not scale in those cases where the radio map is large. Previous attempts to alleviate the computational load rely on the segmentation of the radio map through smart clustering in the offline stage, and a two-step estimation process in the online stage. However, most of the clustering models applied are generic without any consideration about signal propagation and relevant fingerprints are often filtered, resulting in a higher positioning error. This paper introduces Strongest AP Set (SAS), a clustering model conceived for RSSI-based fingerprinting. The results show that SAS is not only able to reduce the computational cost, but also to provide better accuracy than the full model without clustering.**

***Index Terms*—Clustering; Scalability; Positioning; RSSI; Fingerprinting**

I. Introduction

Fingerprint-based positioning has become very popular in the last years with around 1800 published works according to Scopus. The model introduced in RADAR [1], with an offline and an online phase, is still used by many fingerprint-based solutions. Despite being originally conceived for Wi-Fi, fingerprinting is gaining popularity with ZigBee, Bluetooth Low Energy (BLE), LoRaWAN and SigFox [2]–[5].

The core idea of Received Signal Strength Indicator (RSSI) fingerprinting relies on collecting fingerprints in several known locations (i.e., the reference dataset or radio map) in the offline phase. The online phase usually relies on a matching algorithm, which retrieves the most similar fingerprints from the radio map and then computes the position estimate based on them. This matching algorithm, which in essence is a custom implementation of the k-Nearest Neighbour (k-NN) algorithm, requires to compute the similarity between operational fingerprint and all the reference fingerprint in the radio map. In large deployments, when the number of reference samples and detected Access Points (APs) is large, k-NN may not be efficient and scalability problems may arise.

Some works have reduced the computational cost at the operational stage, including filtering by common or strongest APs or clustering. However, Torres-Sospedra *et al.* [6] showed that reducing the computational cost is usually at the expense of increasing the positioning error, being K-Means clustering an alternative providing a good trade-off between both metrics.

Clustering models are generic and they are applied without any specific knowledge about radio signal propagation. We consider that the accuracy provided by the k-NN can not only be kept but also improved by focusing only on the part of the radio map which is relevant to the operational fingerprint.

In this paper, we propose the SAS clustering method, which exploits the concept that the strongest APs are indicating the region where the user is located. Therefore, the idea behind SAS is to cluster the fingerprints in the radio map based on the set of the N strongest APs. Our contributions include:

- A clustering model specific for RSSI-based positioning grounded on the set of strongest APs.
- A sophisticated approach to select the most similar clusters not based on distances in the RSSI feature space.
- Experiments over 10 datasets using the positioning error, time to generate the clusters, and computational cost at the operational stage as benchmarking metrics.

II. Related Work

A popular strategy to reduce the computational cost of fingerprinting relies on clustering. In general, clustering methods make groups of similar objects. Thus, in fingerprinting, each cluster contains similar fingerprints [7], [8]. In the online phase, the search for the most similar fingerprints is done in two steps: first determining which is the most relevant cluster and, then, searching for the most similar fingerprints within that cluster.

The most popular clustering model is K-Means, which splits the RSSI feature space into Voronoi cells, and it has been widely applied to fingerprinting. Anuwatkun *et al.* [8] applied it in combination with the difference of RSSIs, whereas Lee *et al.* [9] developed an algorithm to find the optimal value of K for K-Means, decreasing the positioning error by 20%with respect the same positioning algorithm without clustering.

Variants of K-Means include Fuzzy c-Means (FCM) and K-Medoids. FCM allows a fingerprint to belong to multiple clusters [10]–[12], giving rise to overlapped areas. K-Medoids provides a dataset partition and a cluster selection suitable for fingerprinting [13], being able to detect/exclude outliers [14].

Corresponding Author: M. Ramires (`Moises.Ramires@ccg.pt`)

The authors gratefully acknowledge funding from FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

Affinity Propagation Clustering (APC) has been used in positioning applications. It provides better accuracy than traditional clustering models with less number of features [15]. Caso *et al.* [16] adapted APC for fingerprinting, improving its accuracy and efficiency. APC does not require setting the number of clusters, but the computational costs and memory requirements are prohibitive in large radio maps.

DBSCAN was originally conceived as a density-based clustering model, and it is more robust to outliers present on the reference dataset than other traditional clustering models. Despite its usage in fingerprinting is less common, it has been used to improve the performance and to detect those sub-regions in the operational area where the performance is low [17]. However, DBSCAN seems to be very sensitive to the radio map, ending in low performance in several cases.

A previous comparison showed that clustering models and other optimization rules reduced the computational cost at the expense of a slightly higher positioning error [18]. This work introduces a novel clustering model for fingerprinting, not only grouping similar samples, but also exploiting the information about the current region to reduce the positioning error.

III. Materials and Methods

A. Basics of Clustering in fingerprinting

Fingerprinting requires two phases (see Algorithm 1). In the offline phase, reference fingerprints (s^t) are collected in a set of locations whose position is known in advance, generating a radio map ($\mathcal{T}$). In the online phase, the operational fingerprints (s^v) are compared to the fingerprints stored in the radio map. Their position is estimated using the locations of the most similar fingerprints in the radio map, usually computing their centroid. Therefore, the cost of estimating the position for any operational fingerprint in the test set, $|\mathcal{V}|$, is $O(|\mathcal{T}|)$, which may not be efficient in, for instance, large radio maps.

Clustering adds a new step to the offline phase (see ① in Algorithm 1), which is devoted to generating groups of similar fingerprints and a representative sample for each cluster. This step is just run once per radio map, and it has no impact on the operational time. However, some clustering models, such as Affinity Propagation, are very demanding at this stage. In those cases where the radio map is updated regularly, the time to generate the clusters is a critical factor.

Clustering also modifies the operational phase to perform the two-step search. First, the most relevant cluster is identified (see ② in Algorithm 1). In most of the clustering models, a cluster is represented by its centroid or a "popular" sample. Thus, the coarse search corresponds to finding the most similar cluster representative. Once, the centroid is selected, the fine-grained search is done over the samples belonging to that cluster (see ③ in Algorithm 1), being the reduced radio map $\hat{\mathcal{T}}$ much smaller that the full radio map $\mathcal{T}$ ($|\hat{\mathcal{T}}| << |\mathcal{T}|$). The most significant gains in terms of estimation time are expected to come from the use of a much smaller radio map.

The changes introduced by clustering in the fingerprinting method are highlighted with ①–③ in Algorithm 1.

Algorithm 1 Pseudocode of k-NN for positioning

```
input $\mathcal{T}$, $\mathcal{V}$, $k$
① Offline pre-processing of training datasets
for $i = 1$ to $|\mathcal{V}|$ do
    ② Identify most relevant cluster
    ③ Generate reduced radio map, $\hat{\mathcal{T}}$, using $\mathcal{T}$ and $\mathbf{s}_i^v$
    for $j = 1$ to $|\hat{\mathcal{T}}|$ do
        Compute distance similarity $\mathbf{s}_i^v$ and $\mathbf{s}_j^t$
    end for
    Sort similarities in RSS space
    Select the $k$ closest candidates
    Estimate building, floor and position
end for
Return: Estimated positions, floors, buildings
```

B. The Strongest AP Set (SAS) Clustering

Although traditional clustering models have been able to tackle indoor positioning, the efficiency has often been achieved at the cost of worse positioning accuracy [6]. Some knowledge-based rules can narrow the problem, helping in generating better clusters and providing better ways to identify the most relevant cluster for an operational fingerprint.

In order to take into account the challenges of RSSI-based positioning in clustering, we have developed Strongest AP Set (SAS). SAS exploits the link between the strongest AP and the sub-region in the operational area when creating a cluster. i.e., the set of strongest APs in an operational fingerprint is the key to indicate the coarse-grained region where it was collected since those strong signals cannot be measured anywhere else.

To adapt to any scenario, SAS uses 2 hyperparameters, N and P. The former represents the length of the set of strongest APs in a sample, and the latter indicates the minimum number of common strongest APs between two samples to belong to the same cluster. Finally, SAS follows the algorithmic structure of the traditional fingerprint-based model [1] (see Algorithm 1), where the k-NN model is applied over the reduced radio map, $\hat{\mathcal{T}}$, to estimate the position.

1) Creating the clusters: In the offline phase, SAS clusters the radio map by exploiting the strongest APs according to Algorithm 2. First, the set of strongest APs is identified for every reference sample in the radio map as follows:

1) For each reference fingerprint in the radio map, $\mathcal{T}$, get the identifiers of the N strongest APs and the strongest absolute RSSI value;
2) The APs with undetected RSSI values are excluded from the strongest AP set and the identifier is replaced with the value -1; this happens when the number of observed APs in the reference fingerprint is shorter than N;
3) Fingerprints with P or less detected APs in their strongest AP set, are considered noisy samples and, therefore not included in the filtered radio map $\dot{\mathcal{T}}$;
4) Sort fingerprints in descending order according to the strongest RSSI value and provide sorted filtered radio map $\ddot{\mathcal{T}}$ and corresponding sets of strongest APs SAS.

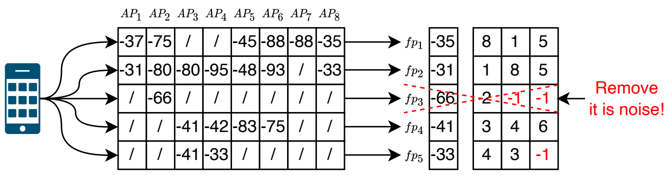


Fig. 1. Identification of the Strongest AP Sets for cluster generation (N=3, P=1)

A visual example is provided in Fig.1 with N=3 and P=1.

Generating the clusters requires to iterate through the sorted list of fingerprints. Marked fingerprints, already members of at least one cluster, cannot form their own cluster. In the iterative process, if a fingerprint is not marked, then:

1) It is marked and it starts its own cluster, i, and its set of strongest APs is used as the cluster i representative.
2) Its set of strongest APs will be compared to the sets of strongest APs of all the other reference fingerprints.
3) Those reference fingerprints that have more than P APs in common with the representative of the created cluster (APs with identifier -1 are not considered) are also marked and assigned to the newly created cluster, i.

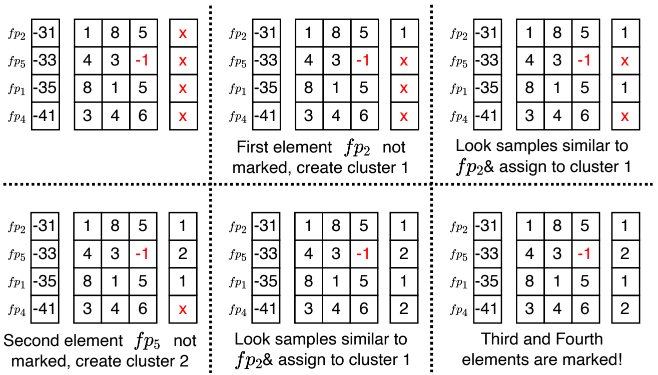
Let us define S_i as the set of the N strongest AP identifiers in s_i^t: ap_1, ap_2, …, ap_N, and Ω as the set of all S_i values. For clarification, we provide a graphical example in Fig.2. The full algorithmic description of the clustering process is provided in Algorithm 2.

2) Cluster identification in the operational phase: In the operational phase, when a new fingerprint is collected, the system should identify which is the most relevant cluster (or set of clusters) for that sample. The corresponding procedure is given by Algorithm 3 and is made of the following steps:

1) Get the identifier of the N strongest APs of the current operational fingerprint, the APs with undetected RSSI values are excluded from set and the identifier is replaced with -2.
2) Compare the set of strongest APs to all cluster representatives and get how many APs they have in common, again undetected APs are not considered.
3) Select the cluster with the highest similarity (highest number of common APs). In the case of a tie, select all the clusters with the highest similarity.

3) Generate the reduced radio map: Once the most similar cluster (or clusters) are identified, the fingerprints to generate the reduced radio map, $\hat{\mathcal{T}}$, can be easily retrieved. Then, the k-NN algorithm is applied to estimate the position.

As the cluster identification may provide multiple "*most similar*" clusters for an operational fingerprint, the fingerprints of each cluster have to be inserted into the reduced radio map. This raises a question about those fingerprints that belong to multiple clusters. In SAS, multiple instances of the same fingerprint are not allowed in the reduced radio map and only one instance is included.


fp_2 -31 1 8 5
fp_5 -33 4 3 -1
fp_1 -35 8 1 5
fp_4 -41 3 4 6
x x x x
First element fp_2 not marked, create cluster 1
1 x x x
Look samples similar to fp_2& assign to cluster 1
1 x 1 x
Second element fp_5 not marked, create cluster 2
1 2 1 x
Look samples similar to fp_2& assign to cluster 1
1 2 1 2
Third and Fourth elements are marked!
1 2 1 2

$$C[1] = \{1,3\} \qquad CR[1] = [1,8,5]$$
$$C[2] = \{2,4\} \qquad CR[2] = [4,3,-1]$$

Fig. 2. Iterative process to generate the Clusters with SAS

Algorithm 2 SAS Clustering

```
Input: $\mathcal{T}$ (ordered in descendent order of strongest AP), $\Omega$, $N$, $P$
// Initialize Variables
$C = \{\}$ // Initialize set of clusters
$CR = \{\}$ // Initialize cluster representatives
$N_{cl} = 0$ // Initialize number of current clusters
for $i = 1$ to $|\mathcal{T}|$ do
    // Process fingerprint if it doesn't belong to any cluster
    if $s_i^t \notin \bigcup_k C[k]$ then
        $N_{cl} = N_{cl} + 1$
        $C[N_{cl}] = \{s^t \in \mathcal{T} : |S_j \cap S_i| > P, \forall j \in \{1, 2, \dots, |\Omega|\}\}$
        $CR[N_{cl}] = S_i$
    end if
end for
Return: $C$, $CR$
```

Algorithm 3 SAS Cluster Identification

```
Input: $CR$, $S^{\mathcal{V}}$
$D = \{\}$ //is the set of selected clusters
$E = \{\}$ // is the set of similarities
for $i = 1$ to $|CR|$ do
    $E[i] = \{|S^{\mathcal{V}} \cap CR[i]|, i\}$
end for
$\dot{E}$ =sort E in descending order of the first element (similarity)
$\ddot{E}$ =select the first element of $\dot{E}$ and all other elements with the same similarity
$D = \bigcup_k C[\ddot{E}[k, 2]]$
Return: $D$
```

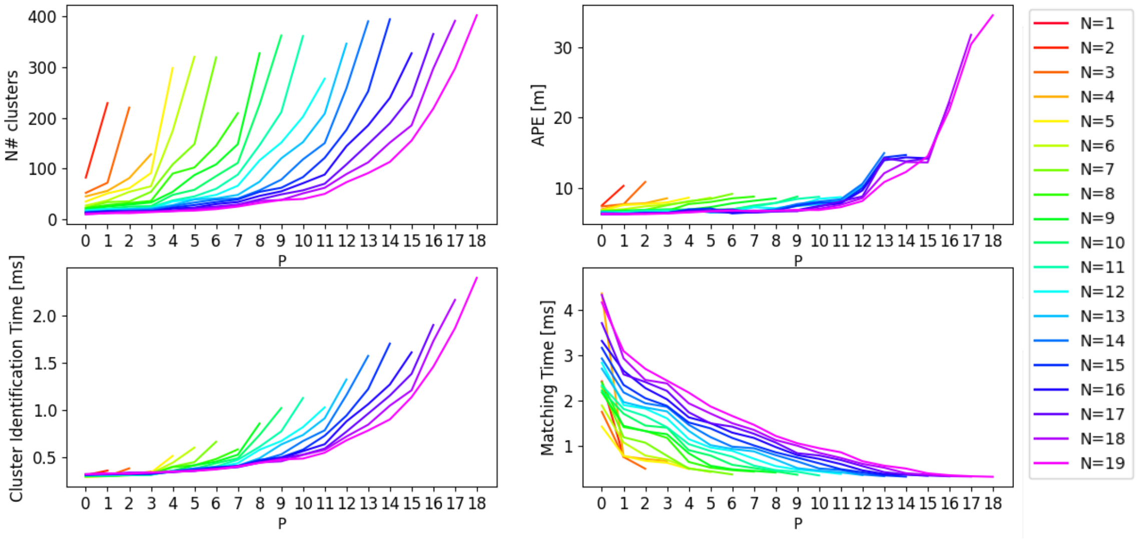


Fig. 3. Plots to visualize the relation between N and P for dataset TUT5. Top-left: Number of clusters. Top-right: APE. Bottom-left: Cluster Identification Time. Bottom-right: Matching Time.

IV. Experiments and Results

A. *Experimental Setup*

In order to assess SAS, we have compared it to the plain k-NN baseline and K-Means as it is a well-known efficient clustering model. Previous tests showed us other clustering models may not be robust or require prohibitive resources [6]. The comparison includes 10 Wi-Fi datasets, similar to [19].

The selected datasets are DSI1, LIB1&2, MAN1, MINT1, UJI1, SAH1, UTS1, TUT5&6 [6], [20]–[23]. For each dataset, we set the hyperparameters' values providing a good trade-off between the positioning error and the computational after running the models multiple times (see Table I). For SAS, the number of generated clusters, $\#C$, is also provided.

TABLE I
Hyperparameters fot the executed models

		DSI1	LIB1	LIB2	MAN1	MINT1	UJI1	SAH1	TUT5	TUT6	UTS1
	$\lvert\mathcal{T}\rvert$	1369	3120	576	14300	4973	19369	9274	442	3107	9108
k-NN	k	30	11	24	14	12	5	4	4	1	24
K-Means	K	2	26	30	3	31	19	5	32	2	30
SAS	N	10	12	6	6	4	3	3	18	10	10
	P	3	7	1	2	1	0	0	0	0	0
	$\#C$	13	58	7	6	5	71	108	10	30	30

The metrics selected to evaluate our solution consist of the Averaged Positioning Error (APE) [24], the clustering time, the cluster identification time, and the matching time. The clustering time refers to the time required to generate the clusters of a radio map. The cluster identification time refers to the average time required to get the most similar cluster(s) for each fingerprint in the operational phase. The matching time corresponds to the process of retrieving the most similar fingerprints from the reduced radio map (full radio map in plain k-NN) and providing an individual position estimation.

The experiments were run in a computer with Intel® Core™ i7-4710MQ CPU 2.50 GHz and Python 3.9.4. Data and code are available in https://github.com/moisesramires/SAS [25]. Simple data cleansing was applied to all datasets, where distant APs and void fingerprints were removed.

B. *SAS hyperparameters*

SAS has 2 hyperparameters, N and P (see Section III-B) and understanding their relationship and impact in the deployment of SAS is crucial. So, for every dataset, several combinations of values for N and P were evaluated. As a result, four graphs were generated for each dataset. In the four graphs, each line represents a value of N, where the x-axis corresponds to values of P. The y-axis corresponds to the APE, the clustering identification time, the matching time, and the number of generated clusters, respectively. Fig.3 shows the plots for dataset TUT5 as an illustrative example, the plots for all datasets are available in [25].

It is worth noting that all datasets report similar plots, and the best trade-off between positioning error and Matching time happens when $P \leq \frac{N}{2}$ in all datasets. Another observation is that the Matching time is going down as P increases because the samples must have a lot more of strongest APs in common and we have more strict clusters as a result. This also means a bigger number of small clusters, and, therefore, the cluster identification time and the number of clusters go up while the Matching time goes down.

By analyzing the data we can conclude that selecting the best N and P combination is challenging. P must be small in comparison to N to achieve the lowest positioning errors, but these solutions will not be the best in terms of computational cost. Selecting the optimal combination of hyperparameters required to find a middle ground between this trade-off.

C. *Results*

Tables II–V present the main results in terms of Averaged Positioning Error (APE) [24], clustering time, averaged cluster identification time and averaged matching time, respectively.

TABLE II
MAIN RESULTS: AVERAGE POSITIONING ERROR [m].

	DSI1	LIB1	LIB2	MAN1	MINT1	UJI1	SAH1	TUT5	TUT6	UTS1
Plain k-NN	4.10	2.44	2.92	2.25	2.50	8.84	6.25	6.25	2.08	7.51
SAS	**4.05**	2.36	**2.78**	**2.24**	2.48	**8.21**	**5.87**	**6.22**	**2.04**	**7.14**
K-Means	4.25	**2.33**	2.81	**2.24**	**2.45**	8.87	6.04	6.45	2.09	7.41

First, the results show the diversity of datasets in terms of APE, with errors ranging from 2.04 m to 8.81 m. K-Means is providing worse results than the plain k-NN in four datasets and slightly better results on the remaining six. K-Means splits the radio map into Voronoi cells, so fingerprints near the cluster boundaries may have less information available and, therefore, worse results. In contrast, SAS is always providing the best results than plain k-NN and K-Means.

It is worth mentioning the outstanding performance of SAS in UJI1, UTS1, SAH1, and TUT5, where the APE provided by SAS is significantly lower than plain k-NN and/or K-Means.

TABLE III
MAIN RESULTS: CLUSTERING TIME [s].

	DSI1	LIB1	LIB2	MAN1	MINT1	UJI1	SAH1	TUT5	TUT6	UTS1
SAS	0.09	0.52	0.18	0.27	0.08	3.47	2.77	0.09	0.75	2.26
K-Means	0.54	0.87	0.96	2.65	4.49	6.72	2.93	0.17	0.74	0.57

Concerning clustering time, plain k-NN does not run this step, so only K-Means and SAS are assessed. SAS provides better results than K-Means in all datasets, except for TUT6, where they have a difference of 0.01 s and in UTS1, where K-Means is the winner. In contrast to Affinity Propagation Clustering (APC), which requires tight memory and computation resources [6], SAS scales to large datasets.

TABLE IV
MAIN RESULTS: CLUSTER IDENTIFICATION TIME [ms].

	DSI1	LIB1	LIB2	MAN1	MINT1	UJI1	SAH1	TUT5	TUT6	UTS1
SAS	0.14	0.29	0.18	0.54	0.16	0.56	0.85	0.32	0.48	0.61
K-Means	0.50	0.30	0.30	0.40	0.28	0.50	0.57	0.42	2.93	0.52

In relation to the averaged cluster identification time, plain k-NN also does not run this step. K-Means and SAS provides similar results throughout the datasets. The identification time depends on the dataset, being in the worst case 3 ms. An important aspect is that despite the time being lower for K-Means, this trend is the opposite in the smallest datasets.

TABLE V
MAIN RESULTS: MATCHING TIME [ms].

	DSI1	LIB1	LIB2	MAN1	MINT1	UJI1	SAH1	TUT5	TUT6	UTS1
Plain k-NN	98.81	48.24	46.65	130.85	36.70	612.42	482.46	15.29	120.83	349.90
SAS	5.28	6.86	18.20	76.18	18.36	29.44	18.75	4.33	15.61	52.46
K-Means	11.45	2.86	2.66	43.61	2.11	50.74	146.97	1.31	79.52	19.87

Regarding the matching time, the plain k-NN does not scale. In the largest dataset (UJI1), it requires more than 0.6 s to provide a position estimate. SAS is providing a matching time below 53 ms in all datasets, except MAN1. MAN1 is the dataset with the highest density of samples with 110 fingerprints per reference point and many samples will be located in the sub-region dominated by the set of strongest APs. In the largest datasets (UJI1, SAH1, TUT6), with the exception of UTS1, the computational cost of SAS is not only satisfactory but better than K-Means. It seems that SAS is promising for large operational areas.

According to all presented results, it seems that K-Means is more efficient than SAS in terms of computational cost in the operational phase. This is in part due to the cluster identification in SAS being more sophisticated. SAS generates clusters that may overlap and multiple clusters can be assigned as the "*most similar cluster*" in the operational phase of SAS. Still, the computational cost of SAS is significantly lower with respect to plain k-NN, especially in the datasets covering very large areas. In contrast, SAS is providing better accuracy than K-Means, and always improving in relation to plain k-NN.

In order to analyze the trade-off between the computational costs and positioning error, we provide a scatter plot in Fig.4 (top), where Averaged Positioning Error (APE) and Averaged Execution Time (AET) are compared for each dataset. AET is the cluster identification time plus the matching time. As the datasets are diverse, it is hard to see a pattern. Thus, both metrics are normalized with respect to a baseline for each dataset resulting in the scatter plot shown in Fig.4 (bottom) with relative values with respect to the plain k-NN.

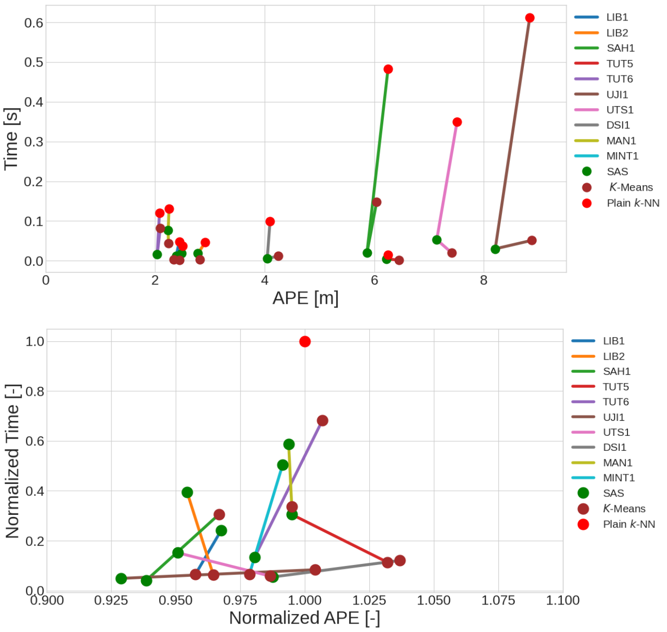


Fig. 4. Efficiency vs. Accuracy: absolute values (top), relative values (bottom)

Fig. 4 identifies 3 scenarios, the first one is where SAS is worse than K-Means both in terms of average error and in terms of computational cost, and this happens for the datasets LIB1, MAN1 and MINT1. Should be noted that even if SAS is losing, it still has an improvement in the average error with respect to the plain k-NN. The second scenario is when K-Means has a better computational cost, but SAS has a better average error, and this is can be observed in the datasets LIB2, TUT5 and UTS1. The third scenario is when SAS beats K-Means in both fields, and this can happens in the datasets that cover large areas (UJI1, SAH1 and TUT6) and DSI1. In particular, Fig. 4 shows 3 main outputs.

1) K-Means is computationally more efficient than SAS in the operational stage at the expense of having slightly worse performance (APE) than the plain k-NN.
2) SAS provides a good efficiency/accuracy trade-off, giving good efficiency without sacrificing accuracy.
3) SAS has not only reduced the computational cost to a minimum expression in the two largest datasets, but also the positioning error has been significantly decreased in around 7.5–12.5%. It seems that SAS is a promising method for datasets involving large operational areas.

V. Discussion & Conclusions

This work introduces the Strongest AP Set (SAS) clustering model, a novel approach in order to split the radio maps into smaller pieces. SAS considers the radio signal properties, focusing on building a scalable Indoor Positioning System (IPS) without any loss in positioning performance.

To assess SAS, we have performed a comprehensive analsysis over 10 datasets and 4 performance metrics. SAS has been compared to the k-NN model without clustering and the K-Means clustering model.

The results show that the proposed SAS clustering model not only reduces the computational costs but also provides a good positioning accuracy. SAS is better than the plain k-NN model for all datasets in terms of positioning error and execution time. In large datasets, the improvements of SAS are outstanding. K-Means, in contrast, provided a worse positioning error than the plain k-NN in 4 datasets. i.e., SAS is scalable without any loss in positioning performance.

SAS filters noisy samples and focuses only on the strongest APs for cluster selection. The results show that the fingerprint-based models should look at trustworthy information instead of focusing on the whole picture.

It is worth mentioning that SAS does not rely on any random initialization, getting the same radio map partition run over run. Despite K-Means providing good averaged results, the initial clusters are randomly generated, which introduces variability in the results from run to run without any metric reporting the quality of the partition done to the radio map.

This work brings new avenues for research and development of reliable and scalable fingerprint-based positioning. The next steps would be focused on finding a lightweight approach to select the most similar clusters and to improve the efficiency on large datasets.

References

[1] P. Bahl and V. Padmanabhan, "Radar: An in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000*, vol. 2, 2000, 775–784 vol.2.

[2] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[3] T. J. Bihl, K. W. Bauer, and M. A. Temple, "Feature selection for rf fingerprinting with multiple discriminant analysis and using zigbee device emissions," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 8, pp. 1862–1874, 2016.

[4] M. Aernouts *et al.*, "Sigfox and lorawan datasets for fingerprint localization in large urban and rural areas," *Data*, vol. 3, no. 2, 2018.

[5] G. G. Anagnostopoulos and A. Kalousis, "A reproducible comparison of rssi fingerprinting localization methods using lorawan," in *th Workshop on Positioning, Navigation and Communications*, 2019.

[6] J. Torres-Sospedra *et al.*, "A comprehensive and reproducible comparison of clustering and optimization rules in wi-fi fingerprinting," *IEEE Transactions on Mobile Computing*, vol. 21, no. 3, pp. 769–782, 2022.

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

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

[9] S. G. Lee and C. Lee, "Developing an improved fingerprint positioning radio map using the k-means clustering algorithm," in *Int. Conf. on Information Networking*, 2020, pp. 761–765.

[10] H. Zhou and N. Van, "Indoor fingerprint localization based on fuzzy c-means clustering," Jan. 2014, pp. 337–340.

[11] D. J. Suroso *et al.*, "Fingerprint-based technique for indoor localization in wireless sensor networks using fuzzy c-means clustering algorithm," in *International Symposium on Intelligent Signal Processing and Communications Systems*, 2011.

[12] C. Zhang *et al.*, "Received signal strength-based indoor localization using hierarchical classification," *Sensors*, vol. 20, 2020.

[13] J. Cheng *et al.*, "A new three-dimensional indoor positioning mechanism based on wireless lan," *Mathematical Problems in Engineering*, 2014.

[14] H. Lin and L. Chen, "An optimized fingerprint positioning algorithm for underground garage environment," in *Int. Conf. on Information Networking*, 2016, pp. 291–296.

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

[16] 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, 2015.

[17] B. Wang *et al.*, "An Improved WiFi Positioning Method Based on Fingerprint Clustering and Signal Weighted Euclidean Distance," eng, *Sensors*, vol. 19, no. 10, 2019.

[18] J. Torres-Sospedra *et al.*, "New cluster selection and fine-grained search for k-means clustering and wi-fi fingerprinting," in *Int. Conf. on Localization and GNSS (ICL-GNSS)*, 2020.

[19] N. Saccomanno, A. Brunello, and A. Montanari, "What you sense is not where you are: On the relationships between fingerprints and spatial knowledge in indoor positioning," *IEEE Sensors Journal*, 2021.

[20] A. Moreira *et al.*, *Wi-Fi Fingerprinting dataset with multiple simultaneous interfaces*, version 1.0, Zenodo, Sep. 2019.

[21] E. S. Lohan, J. Torres-Sospedra, and A. Gonzalez, *WiFi RSS measurements in Tampere University multi- building campus, 2017*, version 1, Zenodo, Aug. 2021.

[22] T. King *et al.*. "CRAWDAD dataset mannheim/compass (v. 2008-04-11)." (2008), [Online]. Available: https://crawdad.org/mannheim/compass/20080411.

[23] T. King, T. Haenselmann, and W. Effelsberg, "On-demand fingerprint selection for 802.11-based positioning systems," in *Int. Symposium on a World of Wireless, Mobile and Multimedia Networks*, 2008.

[24] "ISO/IEC 18305:2016 Information technology — Real time locating systems — Test and evaluation of localization and tracking systems," Standard, 2016.

[25] M. Ramires, *SAS GitHub Repository*, Available: https://github.com/moisesramires/SAS, 2022.