JJEE
Volume 10 | Number 3 | September 2024 | Pages 431-442

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

Clustering Performance Analysis of the K-Medoids Algorithm for Improved Fingerprint-Based Localization

Abdulmalik Shehu Yaro[1,2*], Maly Filip[3], Karel Maly[4], Pavel Prazak[5]

[1, 3, 4, 5] Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové, Czech Republic.
Email: abdulmalik.yaro@uhk.cz
[2] Department of Electronics and Telecommunications Engineering, Ahmadu Bello University, Zaria, Nigeria

Received: Dec 22, 2023 *Revised: Jan 09, 2024* *Accepted: Jan 22, 2024* *Available online: Jun 23, 2024*

***Abstract*—** Fingerprint-based localization, which uses received signal strength (RSS) measurements from spatially deployed wireless access points (APs), is a popular technique for indoor positioning. The size of the fingerprint database has a significant impact on the accuracy of localization. The higher the density of the fingerprint database, the more accurate the localization, but the longer the localization time. Clustering is one of the techniques used such systems to improve localization accuracy and reduce localization time. To cluster fingerprints, the majority of clustering techniques employ a distance-based fingerprint similarity metric. However, the choice of distance metric has a significant impact on the performance of the clustering algorithm. Using four publicly available RSS-based fingerprint databases, this paper investigates the clustering performance of the k-medoids algorithm using six distance metrics, namely Euclidean, Manhattan, cosine, Mahalanobis, Chebyshev, and Canberra distance. Using the silhouette score as a performance metric, the cosine and Euclidean distance metrics outperform the others, with the highest silhouette score values of about 0.38, 0.43, 0.34, and 0.31 on the SEUG_IndoorLoc, IIRC_IndoorLoc, MSI_IndoorLoc, and IPIN_2019_PIEP_UM databases, respectively. It demonstrates that on these four databases, using Euclidean distance as well as the angle between fingerprint measurement vectors is the best option for generating efficient clusters that will result in high localization accuracy and low localization time.

***Keywords*—** K-Medoids; Distance metric; Received signal strength; Silhouette score; Clustering; Fingerprint; Indoor localization.

1. INTRODUCTION

Indoor localization is becoming increasingly important and has applications in a variety of areas, such as navigation assistance and asset tracking. Due to signal attenuation and multipath propagation, conventional localization techniques, such as global positioning systems (GPS), are unreliable for indoor localization [1]. Fingerprint-based localization, which uses received signal strength (RSS) or channel state information (CSI) measurements from spatially deployed wireless access points (APs), has emerged as a promising alternative for indoor localization [2]. To determine the location of an indoor user, the fingerprint-based indoor localization system employs a two-phase process, namely the offline phase and the online phase [1, 3]. The offline phase involves the acquisition of radio frequency (RF) signals from the wireless APs, the determination of the RSS or CSI measurements from the received RF signals, and the creation of an RSS or CSI-based fingerprint measurement database [1]. A fingerprint measurement is a vector containing RSS or CSI measurements that were collected from several wireless APs at a given reference location (RL). The fingerprint measurement database is a database that contains fingerprint measurements mapped to the RL from where

* Corresponding author
DOI: https://doi.org/10.5455/jjee.204-1703256698

they are obtained. The online phase of the fingerprint-based localization system involves determining the location of an indoor user using the instantly acquired fingerprint measurement by scanning through the fingerprint database using a localization matching algorithm such as the k-nearest neighbor (k-NN)[4], support vector machine (SVM) [5], and Gaussian mixture model (GMM) [6].

The localization performance of the fingerprint-based localization system is dependent on several factors, one of which is the density of the fingerprint database [2, 7]. Fingerprint database density refers to the number of RLs used in the generation of the database, and the more RLs used, the higher the density. To increase localization accuracy, several researchers have reported using a high-density fingerprint database. The use of a high-density database results in a high localization time; that is, it takes longer to determine the localization of an indoor user using the instantly acquired fingerprint measurement. However, low or near-real-time localization time and high localization accuracy are two of the primary objectives of any wireless-based localization system. Researchers proposed the use of clustering techniques to overcome the trade-off between localization time and accuracy [7–11]. Fingerprint database clustering is the process of grouping fingerprint measurements into clusters based on a common parameter known as the similarity metric. There are several commonly used fingerprint database clustering techniques, some of which are k-means [12], c-means [13], affinity propagation clustering (APC) [3], and density-based spatial clustering (DBSCAN) [14]. The k-means is the most commonly used clustering algorithm due to its simplicity and moderate clustering performances [15, 16]. In this paper, an improved version of the k-means, known as the k-medoids, is considered [17–19]. Unlike the k-means, which use the mean average of all the fingerprints within a cluster as the representative of that cluster, known as the cluster centroid, the k-medoids use an actual fingerprint within the cluster.

The performance of any clustering technique is dependent on the fingerprint similarity metric used [17, 20, 21], and the most used fingerprint similarity metric is the distance-based similarity metric. Different distance-based similarity metrics capture the similarity between fingerprint measurements in different ways. However, in general, the distance-based similarity metric quantifies the similarity between two fingerprint measurements by measuring the distance between their RSS vectors. The smaller the distance, the more similar the fingerprints are considered to be. Some examples of distance-based similarity metrics are Euclidean distance, Manhattan distance, Minkowski distance, Mahalanobis distance, Canberra distance, and Cosine similarity [8]. Since the similarity metric plays an important role in the performance of the clustering algorithm, it is important to investigate the impact of different distance-base fingerprint similarity metrics on the clustering performance. Thus, in this paper, the clustering performance of the k-medoids algorithm is determined and compared considering different distance-based similarity metrics using an RSS-based fingerprint database. The contribution of this paper is the investigation of the impact of some of the commonly used distance-based fingerprint similarity metrics on the k-medoids clustering performance using several RSS-based fingerprint databases of varying fingerprint density.

The paper reminder is organized as follows: Section 2 provides an overview of the k-medoids clustering algorithm process as well as a review of related work. This is followed by a mathematical description of the distance-based similarity metrics that will be used with the k-medoids algorithm in Section 3. Section 4 contains the simulation and result discussions, and Section 5 contains the conclusion and recommendations for future works.

2. OVERVIEW OF K-MEDOIDS CLUSTERING ALGORITHM AND REVIEW OF RELATED WORKS

As earlier stated, this paper considered the k-medoids algorithm to cluster RSS-based fingerprint databases, and in this section of the paper, an overview of the k-medoids algorithm clustering process is first presented, followed by a review of related works.

2.1. K-Medoids Algorithm Clustering Process

The K-medoids algorithm is a robust and efficient clustering algorithm that has been widely used to improve the localization accuracy of fingerprint-based localization systems. The K-medoids algorithm partitions the fingerprint database into a predetermined number of clusters (k). Unlike k-means clustering, which uses the average of fingerprints in each cluster as its cluster centroid, k-medoids use one of the fingerprints within the cluster as its representative, which is known as a cluster medoid. Below is a summary of the steps taken to implement the k-medoids algorithm [17, 22]. A graphical description of the clustering process of the k-medoids algorithm is shown in Fig. 1.

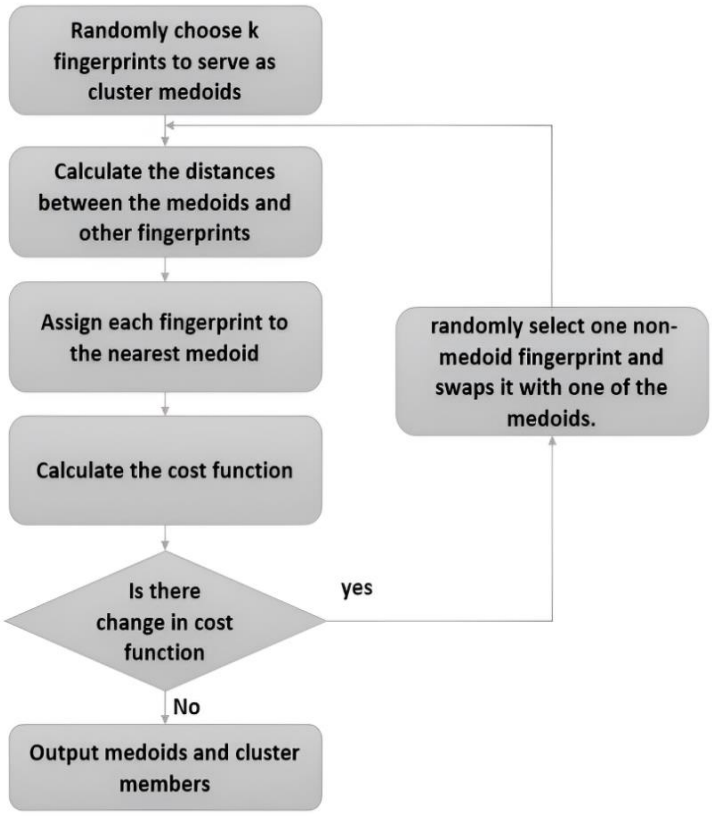

Fig. 1. K-medoids algorithm clustering process.

Step-1: Initialize medoids.

As the initial medoids, choose k fingerprints at random from the fingerprint database. These medoids will serve as the centers of the clusters.

Step-2: Cluster assignments.

Assign the reminder of the fingerprints to the nearest medoid. This can be accomplished with any distance-based metric, such as the Euclidean distance or the Manhattan distance.

Step-3: Cost calculation.

Calculate the cost function, which measures the overall dissimilarity between fingerprints and the medoids to which they are assigned.

This cost function represents the overall quality of clustering.

Step-4: Optimize medoids.

Continuously swap medoids with non-medoid fingerprints as the cost function decreases. This swapping process aims to find better medoids that can improve the clustering structure while lowering overall costs.

Step-5: Return cluster assignments and medoids.

Return the final cluster assignments and medoids for each cluster once the algorithm converges or the specified number of iterations has been reached. These outcomes represent the clustering solution.

Steps 1–5 summaries the steps taken to implement the k-medoids algorithm. A review of works related to the performance comparison of the k-medoids algorithm using different distance-based fingerprint similarity metrics is presented in the following subsection.

2.2. Review of Related Works

Several works have been published on the use of the k-medoids algorithm using different distance-based similarity metrics [17–19, 22–24], but most of the works in these categories used Euclidean distance as the similarity measure metric. Fewer works have been published on the clustering performance comparison of the k-medoids algorithm, considering more than one distance similarity measure metric. For example, the authors of [23] compared the performance of the k-means and k-medoids clustering algorithms using only two types of distance-based similarity metrics, Manhattan and Euclidean. In addition, the authors of [17] performed a k-medoids performance comparison using Manhattan, Euclidean, and Chebyshev distances as similarity measure metrics, while in [24], Euclidean, Canberra, and Chebyshev distances are used. A summary of the comparison of work related to k-medoids algorithm clustering performance considering multiple distance similarity measure metrics is shown in Table 1.

Table 1. Comparison of works on k-medoids clustering performance considering different distance-based similarity metrics.

Related work	Distance metric	No. of databases	Density	
			Points/RLs	Attributes
[17]	Manhattan Euclidean Chebyshev	1	647	5
[23]	Manhattan Euclidean	1	983	10
[24]	Euclidean Chebyshev Canberra	1	147679	6
Current work	Euclidean	4	49	10
	Chebyshev		194	4
	Mahalanobis		4973	11
	Manhattan Canberra Cosine		1000	8

Table 1 shows that most of the published works used only one database with two to three of the most used distance metrics to evaluate the performance of the k-medoid algorithm.

Furthermore, the databases considered in earlier published works can be classified as dense. The effect of any distance similarity metric on the clustering performance of the k-medoids algorithm varies with databases and their densities. While some distance metrics may work well for large databases, they may be inefficient or perform poorly when applied to smaller databases.

As a result, it is critical to investigate the performance of a clustering algorithm using various distance metrics on databases of various density sizes, which is what this study aims to do. The performance of the k-medoids algorithm was evaluated using six different distance metrics on four different databases, ranging in density size from small to large.

The distance metrics considered are Euclidean, Chebyshev, Mahalanobis, Manhattan, Canberra, and cosine. The mathematical descriptions of each of the distance metrics considered in this paper are presented in the following section.

3. MATHEMATICAL DESCRIPTION OF DISTANCE-BASED FINGERPRINT SIMILARITY METRIC FOR K-MEDOID CLUSTERING

The mathematical description of the distance metrics considered in determining the performance of the k-medoids algorithm is presented in this section of the paper.

As mentioned earlier, the distance metrics considered are Euclidean, Chebyshev, Mahalanobis, Manhattan, Canberra, and Cosine.

3.1. Euclidean Distance based Similarity Metric

This is a straightforward and most used distance-based similarity metric by most clustering algorithms. It calculates the shortest straight-line distance between two fingerprints.

Given two fingerprint measurement vectors $\mathbf{f}_1$ and $\mathbf{f}_2$, which contain RSS measurements obtained from N wireless APs, as shown in Eqs. (1) and (2):

$$\mathbf{f}_1 = [rss_1^1, rss_2^1, rss_3^1 \dots rss_N^1] \quad (1)$$

$$\mathbf{f}_2 = [rss_1^2, rss_2^2, rss_3^2 \dots rss_N^2] \quad (2)$$

The Euclidean distance between fingerprints $\mathbf{f}_1$ and $\mathbf{f}_2$ can be obtained using Eq. (3) [17].

$$d_{Euclidean}(\mathbf{f}_1, \mathbf{f}_1) = \sqrt{\sum_{i=1}^{N} \left(rss_i^1 - rss_i^2\right)^2} \quad (3)$$

3.2. Chebyshev Distance based Similarity Metric

The Chebyshev distance, also known as the maximum norm, is another distance-based similarity metric considered in this paper. It is useful for calculating distances in high-dimensional spaces where other metrics, such as the Euclidean distance, may be affected by outliers.

The Chebyshev distance between the two fingerprint measurements in Eqs. (1) and (2) is calculated mathematically as [17]:

$$d_{Chebyshev}(\mathbf{f}_1, \mathbf{f}_1) = \max\left\{\left|rss_1^1 - rss_1^2\right|, \left|rss_2^1 - rss_2^2\right|, \dots, \left|rss_3^1 - rss_3^2\right|\right\} \quad (4)$$

The Chebyshev distance, based on Eq. (4), finds the maximum absolute difference between the corresponding RSS measurements in the two fingerprints, measuring the distance along the coordinate where the fingerprints are farthest apart.

3.3. Mahalanobis Distance based Similarity Metric

Another distance-based similarity metric considered in the performance analysis is the Mahalanobis distance. It is a generalization of the Euclidean distance that takes the covariance structure of the entire fingerprint database into account. The Mahalanobis distance is a more robust metric than the Euclidean distance in the presence of outliers. It is also unaffected by the size of the differences in fingerprint measurements. The Mahalanobis distance between the two fingerprint measurements in Eqs. (1) and (2) is calculated as follows [25]:

$$d_{Mahalanobis}(\mathbf{f}_1, \mathbf{f}_2) = \sqrt{(\mathbf{f}_1 - \mathbf{f}_2)^T \mathbf{S}^{-1} (\mathbf{f}_1 - \mathbf{f}_2)} \quad (5)$$

where $\mathbf{S^{-1}}$ is the inverse of the covariance matrix "$\mathbf{S}$" **and** ^{T} denotes the transpose of a matrix or vector.

3.4. Manhattan Distance based Similarity Metric

The Manhattan distance, also known as the city-block distance, is another commonly used distance-based similarity metric considered in this paper. It is calculated by adding the absolute differences in RSS measurements between each fingerprint. The Manhattan distance is a straightforward and computationally efficient metric that works well with large fingerprint databases. It is also resistant to outliers, making it a versatile choice for a variety of applications. The Manhattan distance between the two fingerprint measurements in Eqs. (1) and (2) is calculated as follows [17]:

$$d_{Mahalanobis}(\mathbf{f}_1, \mathbf{f}_2) = \sum_{i=1}^{N} \left| rss_i^1 - rss_i^2 \right| \quad (6)$$

3.5. Canberra Distance based Similarity Metric

The Canberra distance, also known as the L1-norm distance, is a distance-based similarity metric used to calculate the distance between two fingerprint measurements. Because it is unaffected by the magnitude of differences between fingerprints, it is a reliable metric for detecting outliers. As a result, it is an excellent choice for applications where the fingerprint database may contain noisy or incorrect values. Other metrics, such as the Euclidean distance, are more sensitive to the overall structure of the fingerprints in the database than the Canberra distance. The Canberra distance between fingerprints $\mathbf{f}_1$ and $\mathbf{f}_2$ can be calculated mathematically as [24]:

$$d_{Canberra}(\mathbf{f}_1, \mathbf{f}_2) = \sum_{i=1}^{N} \frac{|rss_i^1 - rss_i^2|}{|rss_i^1| + |rss_i^2|} \quad (7)$$

Eq. (7) computes the sum of the absolute differences between the corresponding fingerprint measurements, normalised by the sum of the absolute values of the RSS measurements. This normalisation helps to mitigate the impact of large values.

3.6. Cosine Distance based Similarity Metric

The cosine distance, also known as cosine similarity, is the final distance-based similarity metric considered in this paper. It measures the similarity between two fingerprints using the angle between them, regardless of their magnitude. The angle between the two fingerprint measurement vectors is defined as the cosine distance and is mathematically expressed as [8]:

$$d_{Cosine}(\mathbf{f}_1, \mathbf{f}_2) = 1 - \frac{\sum_i^N (rss_i^1 \times rss_i^2)}{\sqrt{\sum_i^N (rss_i^1)^2} \times \sqrt{\sum_i^N (rss_i^2)^2}} \quad (8)$$

The distance metrics discussed in this section will be used in Step 2 of the k-medoids algorithm clustering processing described in Subsection 2.1. That is, once the cluster medoids have been identified, the distance between the cluster medoids and the remaining fingerprints in each cluster is calculated using the distance metrics listed from 1 to 6. The performance of the k-medoids algorithm using each of these distance metrics is evaluated in the following section using four publicly available RSS-based fingerprint databases of varying fingerprint density sizes.

4. SIMULATION RESULT AND DISCUSSION

The performance of the k-medoids algorithm using each of the distance-based similarity metrics presented in Section 3 is determined and compared in this section using four publicly available RSS-based fingerprint databases. The simulation setup and parameters are presented first, followed by a clustering performance evaluation and comparison.

4.1. Simulation Setup and Parameters

The clustering performances of the k-medoids algorithm using the six distance-based similarity metrics are determined and compared using four publicly available RSS-based fingerprint databases with characteristics shown in Table 2.

Table 2. Characteristics of the four RSS-based fingerprint databases considered.

Databases	Wireless technology	Database characteristics	
		Number of APs	Number of fingerprints
IIRC_IndoorLoc [26]	Zigbee	4	194
SEUG_ IndoorLoc [27]	Wi-Fi	3	49
MSI_IndoorLoc [28]	Wi-Fi	11	4973
IPIN_2019_PIEP_UM [29]	Wi-Fi	8	1000

The databases MSI_IndoorLoc and IPIN_2019_PIEP_UM were used at the International Conferences on Indoor Positioning and Indoor Navigation (IPIN) in 2017 and 2019, respectively. These two databases can be considered to be dense. The MSI_IndoorLoc database has a total of 4973 fingerprint measurement vectors containing RSS measurements obtained from four Wi-Fi-based wireless APs, all of which are collected within an indoor environment with a coverage area of 1000 m^2.

The IPIN_2019_PIEP_UM database is generated within an indoor environment with a coverage area of 1000 m^2 and contains about 1000 fingerprint measurements generated using eight Wi-Fi-based wireless APs. The databases IIRC_IndoorLoc and SEUG_IndoorLoc are smaller in size, with a total of 194 and 49 fingerprints, respectively. The IIRC_IndoorLoc database was generated using four Zigbee-based wireless APs, while the SEUG_IndoorLoc database was generated using three Wi-Fi-based wireless APs.

The total indoor coverage areas for the IIRC_IndoorLoc and SEUG_IndoorLoc databases are 161 m^2 and 33 m^2, respectively.

The silhouette score value is used as the clustering performance metric in this paper to evaluate the clustering performance of the k-medoids algorithm using various distance metrics.

4.2. Clustering Performance Comparison

The silhouette score is a metric for assessing the quality of clusters produced by a clustering algorithm. It measures how well clusters are separated from one another and how well fingerprints are assigned to their respective clusters. Silhouette scores range from -1 to 1, with a higher score value of 1 indicating a better clustering result. When a clustering algorithm with any similarity metric has a silhouette score of 0.7 or higher, it is considered to have good clustering performance. This indicates that all the clusters are reasonably well-separated and that the fingerprints are generally well-assigned to their respective clusters. Table 3 shows the silhouette score values for the different distance metrics used by the k-medoids algorithm to cluster the four RSS-based fingerprint databases. A graphical representation of the results in Table 3 can be seen in Fig. 2.

Table 3. Silhouette score for each distance-based similarity metric.

Distance metric	Silhouette score			
	SEUG_IndoorLoc	IIRC_IndoorLoc	MSI_IndoorLoc	IPIN_2019_PIEP_UM
Euclidean	0.37	0.42	0.34	0.31
Chebyshev	0.31	0.38	0.28	0.24
Mahalanobis	0.29	0.42	0.09	0.13
Manhattan	0.32	0.40	0.30	0.29
Canberra	0.29	0.32	0.25	0.28
Cosine	0.38	0.43	0.31	0.30

The silhouette scores obtained by the k-medoids algorithm with all distance-based similarity metrics for all four databases are very low, less than 0.5, indicating that the fingerprint clusters were poorly generated. This indicates that the clusters are not well-separated, and fingerprints may be misassigned to other clusters. The result discussion, on the other hand, will disregard the poorly generated clusters and compare the silhouette scores to determine which of the six fingerprint similarity metrics considered is slightly better.

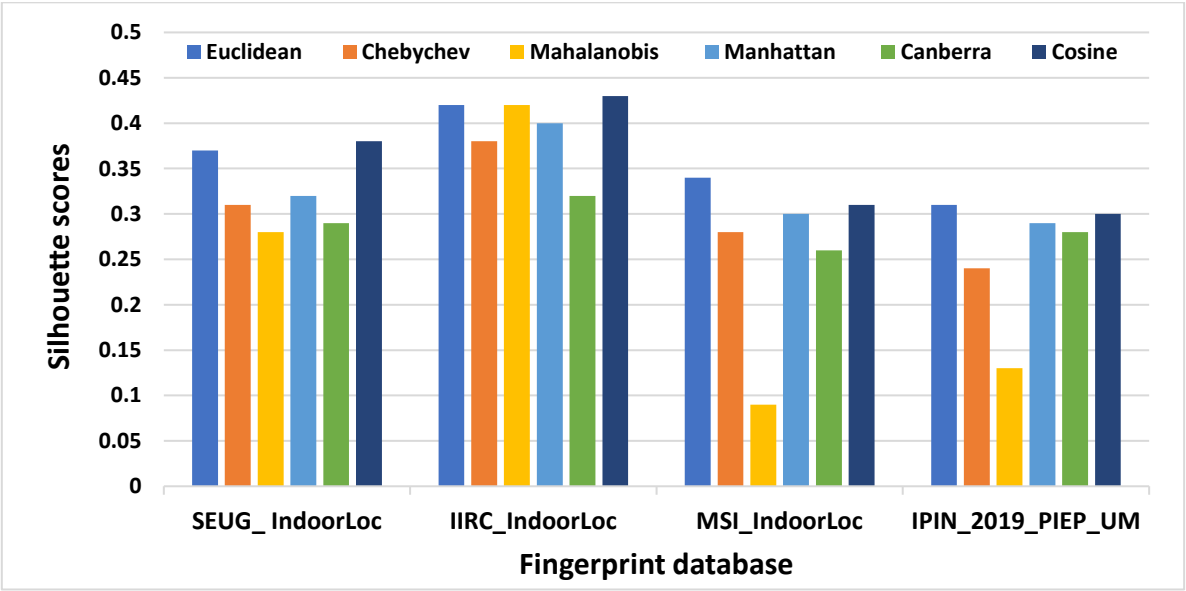

Fig. 2. Silhouette score comparison for different fingerprint databases.

Looking at the silhouette scores for the SEUG_IndoorLoc and IIRC_IndoorLoc databases, which are considered to be small in fingerprint density, it can be seen that the cosine distance has the highest silhouette score values of 0.38 and 0.43, respectively. This is followed by the

Euclidean distance, with silhouette score values of 0.37 and 0.42, respectively. In the SEUG_IndoorLoc database, the Manhattan distance came in third with a silhouette score value of 0.32. However, on the IIRC_IndoorLoc database, Mahalanobis came in third with a silhouette score value of 0.42, which is the same as that of the Euclidean distance. The silhouette score value difference between the cosine and Euclidean distances is insignificant. This means that both are equally good choices to be used with the k-medoids algorithm to cluster the SEUG_IndoorLoc and IIRC_IndoorLoc databases. The Canberra distance metric is the worst metric to be used with the k-medoids algorithm, as it has the lowest silhouette score values of 0.29 and 0.32 on the SEUG_IndoorLoc and IIRC_IndoorLoc databases, respectively. This means clusters generated using the Canberra distance as a similarity metric will result in very poor localization accuracy. Also, it suggests that these two databases do not have fingerprint outliers

Extending the analysis to the MSI_IndoorLoc and IPIN_2019_PIEP_UM databases, which are considered to be relatively dense in size, the Euclidean distance has the highest silhouette score values of 0.34 and 0.31, respectively. This is followed by the cosine distance metric, with silhouette score values of 0.31 and 0.30, respectively. The Manhattan distance came in third with silhouette score values of 0.30 and 0.31, respectively. For the MSI_IndoorLoc database, the silhouette score value of the Euclidean distance is significantly higher than that of the cosine and Manhattan, making it the best choice for use with the k-medoid algorithm. As for the IPIN_2019_PIEP_UM database, the silhouette score values for the Euclidean, Cosine, and Manhattan are nearly the same, meaning that any of the three is a good choice to use with the k-medoids algorithm. The Mahalanobis distance metric in this case has the lowest silhouette score values in both databases. This means that clusters generated using the Mahalanobis distance as a fingerprint similarity metric with the k-medoids algorithm will result in very poor localization accuracy. It also suggests the Mahalanobis distance has a very high sensitivity to the density of the fingerprint database and that its performance is best on a fingerprint database with a low density.

Even though all six metrics considered generated poorly separated clusters, the Euclidean and cosine distance metrics appear to be slightly better choices when considering all four databases. The clusters they generated could result in a slight improvement in localization accuracy compared to the others. This is regardless of the density of the fingerprint database. On large fingerprint databases, as part of the Euclidean and cosine distance metrics, the Manhattan distance metric is also a good choice, as it has comparable silhouette scores close to those of the Euclidean and cosine distances. Unlike the Euclidean and Manhattan distances, which measure the similarity between two fingerprints using actual distances, the cosine distance uses the angle distance between fingerprint measurement vectors. For it to have equivalent performances close to the Euclidean distance, it shows that the fingerprint measurement vectors in these four databases are likely to have similar angles. As a result, the cosine distance metric was able to effectively identify clusters of fingerprints with similar angles.

In summary, based on the four databases considered, the Euclidean and cosine distance metrics performed better than the other four distance metrics considered. However, in the practical implementation of the k-medoids algorithm, there are several factors that need to be considered when choosing the best distance-based fingerprint similarity metrics. This is because each of these distance metrics is affected by specific indoor environmental factors. For instance, in an indoor environment, RSS measurements are known to fluctuate due to several

factors, such as the presence and absence of crowds and temporal and ambient conditions. This results in a noisy fingerprint database, and it is known that both cosine and Euclidean distance metrics are very sensitive to noisy fingerprint databases. Furthermore, considering a large fingerprint database, it is computationally intensive to calculate the pairwise distances or angles between fingerprints, which could result in a longer localization time. Thus, when choosing the appropriate distance metrics for the practical implementation of any clustering, it is critical to carefully consider the characteristics of the fingerprint database. Furthermore, fingerprint database pre-processing techniques like noise reduction and database dimensionality reduction could be used to reduce or mitigate some of the limitations of using cosine and Euclidean distances as fingerprint similarity metric measures.

5. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

In this paper, the clustering performance of the k-medoids algorithm using six different distance metrics, namely Euclidean, Chebyshev, Mahalanobis, Manhattan, Canberra, and cosine, is determined. The clustering performance of the k-medoids algorithm with the different distance metrics is determined using four publicly available RSS-based fingerprint databases with different fingerprint densities. Using silhouette scores as the clustering performance metrics, the results show that all six distance metrics had silhouette scores that were less than 0.5, indicating poorly generated clusters. However, the k-medoids algorithm with the Euclidean and cosine distance metrics generated the most clusters that were slightly better separated than the others, irrespective of the sizes of the databases. The Manhattan distance metric performance is similar to that of Euclidean and cosine distance metrics only on the larger density databases. A well-clustered fingerprint database increases localization accuracy and reduces the localization time of the system. Even though the clusters were poorly generated, fingerprint clusters generated using Euclidean, cosine, and Manhattan distances as fingerprint similarity metrics will have slightly better localization accuracy. As such, future work will focus on investigating how these poorly separated clusters will affect the overall localization performance of the fingerprint-based localization system.

Acknowledgement: The authors acknowledge the support provided by FIM SPEV 2023 and the Excellence projects at the Faculty of Informatics and Management, University of Hradec Kralove, Czech Republic. Special thanks go to Eng. Daniel Schmidt for his assistance with data collection and preparation.

REFERENCES

[1] A. Yaro, F. Maly, P. Prazak, "A survey of the performance-limiting factors of a 2-dimensional RSS fingerprinting-based indoor wireless localization system," *Sensors*, vol. 23, no. 5, p. 2545, 2023, doi: 10.3390/s23052545.

[2] P. Sadhukhan, "Performance analysis of clustering-based fingerprinting localization systems," *Wireless Networks*, vol. 25, no. 5, pp. 2497–2510, 2019, doi: 10.1007/s11276-018-1682-7.

[3] A. Yaro, F. Malý, K. Malý, "Improved indoor localization performance using a modified affinity propagation clustering algorithm with context similarity coefficient," *IEEE Access*, vol. 11, pp. 57341–57348, 2023, doi: 10.1109/ACCESS.2023.3283592.

[4] S. Shang, L. Wang, "Overview of WiFi fingerprinting-based indoor positioning," *IET Communications*, vol. 16, no. 7, pp. 725–733, 2022, doi: 10.1049/cmu2.12386.

[5] H. Yang, Y. Wang, C. Seow, M. Sun, M. Si, L. Huang, "UWB sensor-based indoor LOS/NLOS localization with support vector machine learning," *IEEE Sensor Journal*, vol. 23, no. 3, pp. 2988–3004, Feb. 2023, doi: 10.1109/JSEN.2022.3232479.

[6] M. Alfakih, M. Keche, H. Benoudnine, A. Meche, "Improved Gaussian mixture modeling for accurate Wi-Fi based indoor localization systems," *Physical Communication*, vol. 43, p. 101218, 2020, doi: 10.1016/j.phycom.2020.101218.

[7] A. Yaro, F. Malý, K. Malý, "A two-nearest wireless access point-based fingerprint clustering algorithm for improved indoor wireless localization," *Emerging Science Journal*, vol. 7, no. 5, pp. 1762–1770, 2023, doi: 10.28991/ESJ-2023-07-05-019.

[8] D. Xu, Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, 2015, doi: 10.1007/s40745-015-0040-1.

[9] J. Bi, H. Lu, H. Cao, G. Yao, W. Sang, J. Zhen, Y. Liu, "Improved indoor fingerprinting localization method using clustering algorithm and dynamic compensation," *International Journal of Geo-Information*, vol. 10, no. 9, p. 613, 2021, doi: 10.3390/ijgi10090613.

[10] E. Gomes, M. Fonseca, A. Lazzaretti, A. Munaretto, C. Guerber, "Clustering and hierarchical classification for high-precision RFID indoor location systems," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5141–5149, 2022, doi: 10.1109/JSEN.2021.3103043.

[11] S. Subedi, H. Gang, N. Ko, S. Hwang, J. Pyun, "Improving indoor fingerprinting positioning with affinity propagation clustering and weighted centroid fingerprint," *IEEE Access*, vol. 7, pp. 31738–31750, 2019, doi: 10.1109/ACCESS.2019.2902564.

[12] H. Zhao, "Design and implementation of an improved k-means clustering algorithm," *Mobile Information Systems*, vol. 2022, pp. 1–10, 2022, doi: 10.1155/2022/6041484.

[13] W. Jiang, X. Fang, J. Ding, "Gaussian kernel fuzzy c-means algorithm for service resource allocation," *Scientific Programming*, vol. 2020, pp. 1–6, 2020, doi: 10.1155/2020/8889480.

[14] J. Bi, H. Cao, Y. Wang, G. Zheng, K. Liu, N. Cheng, M. Zhao, "DBSCAN and TD integrated wi-fi positioning algorithm," *Remote Sensing*, vol. 14, no. 2, p. 297, 2022, doi: 10.3390/rs14020297.

[15] E. Oti, M. Olusola, F. Eze, S. Enogwe, "Comprehensive review of k-means clustering algorithms," *International Journal of Advances in Scientific Research and Engineering*, vol. 07, no. 08, pp. 64–69, 2021, doi: 10.31695/IJASRE.2021.34050.

[16] M. Ahmed, R. Seraj, S. Islam, "The k-means algorithm: a comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, 2020, doi: 10.3390/electronics9081295.

[17] A. Sunge, Y. Heryadi, Y. Religia, Lukas, "Comparison of distance function to performance of k-medoids algorithm for clustering," Proceedings of International Conference on Smart Technology and Applications, doi: 10.1109/ICoSTA48221.2020.1570615793.

[18] T. Madhulatha, "Comparison between k-means and k-medoids clustering algorithms," Proceedings of International Conference on Advances in Computing and Information Technology, doi: 10.1007/978-3-642-22555-0_48.

[19] N. Shamsuddin, N. Mahat, "Comparison between k-means and k-medoids for mixed variables clustering," Proceedings of the Third International Conference on Computing, Mathematics and Statistics, doi: 10.1007/978-981-13-7279-7_37.

[20] Y. Thakare, S. Bagal, "Performance evaluation of k-means clustering algorithm with various distance metrics," *International Journal of Computer Applications*, vol. 110, no. 11, pp. 12–16, 2015, doi: 10.5120/19360-0929.

[21] M. Gonzales, L. Uy, J. Sy, M. Cordel, "Distance metric recommendation for k-means clustering: a meta-learning approach," TENCON 2022 - 2022 IEEE Region 10 Conference, 2022, doi: 10.1109/TENCON55691.2022.9978037.

[22] H. Park, C. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009, doi: 10.1016/j.eswa.2008.01.039.

[23] S. Abbas, A. Aslam, A. Rehman, W. Abbasi, S. Arif, S. Kazmi, "K-Means and K-Medoids: cluster analysis on birth data collected in city Muzaffarabad, Kashmir," *IEEE Access*, vol. 8, pp. 151847–151855, 2020, doi: 10.1109/ACCESS.2020.3014021.

[24] S. Gultom, S. Sriadhi, M. Martiano, J. Simarmata, "Comparison analysis of k-means and k-medoid with Euclidian distance algorithm, Canberra distance, and Chebyshev distance for big data clustering," IOP Conference Series: Materials Science and Engineering, 2018, doi: 10.1088/1757-899X/420/1/012092.

[25] R. Maesschalck, D. Rimbaud, D. Massart, "The Mahalanobis distance," *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000, doi: 10.1016/S0169-7439(99)00047-7.

[26] T. Alhmiedat, "Fingerprint-based localization approach for WSN using machine learning models," *Applied Sciences*, vol. 13, no. 5, p. 3037, 2023, doi: 10.3390/app13053037.

[27] S. Sadowski, P. Spachos, K. Plataniotis, "Memoryless techniques and wireless technologies for indoor localization with the internet of things," *IEEE Internet Things Journal*, vol. 7, no. 11, pp. 10996–11005, 2020, doi: 10.1109/JIOT.2020.2992651.

[28] A. Moreira, I. Silva, F. Meneses, M. Nicolau, C. Pendao, J. Sospedra, "Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning," International Conference on Indoor Positioning and Indoor Navigation, 2017, doi: 10.1109/IPIN.2017.8115914.

[29] J. Sospedra, A. Moreira, G. Silva, M. Nicolau, M. Sanz, L. Silva, "Exploiting different combinations of complementary sensor's data for fingerprint-based indoor positioning in industrial environments," International Conference on Indoor Positioning and Indoor Navigation, doi: 10.1109/IPIN.2019.8911758.