

RESEARCH ARTICLE

Enhancing DBSCAN Clustering for Fingerprint-Based Localization With a Context Similarity Coefficient-Based Similarity Measure Metric

ABDULMALIK SHEHU YARO^{1,2}, FILIP MALÝ¹, KAREL MALÝ¹, AND PAVEL PRAZAK¹

¹Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové, 500 03 Hradec Králové, Czech Republic

²Department of Electronics and Telecommunications Engineering, Ahmadu Bello University, Zaria 810106, Nigeria

Corresponding author: Abdulmalik Shehu Yaro (abdulmalik.yaro@uhk.cz)

This study is supported by the FIM Excellence project run at the Faculty of Informatics and Management, University of Hradec Kralove, Czech Republic.

ABSTRACT In fingerprint-based localization systems, clustering fingerprint databases is a proposed technique for improving localization accuracy while reducing localization time. Among various clustering algorithms, density-based spatial clustering of applications with noise (DBSCAN) stands out for its robustness to outliers and ability to accommodate fingerprint databases of various shapes. However, the clustering performance of the DBSCAN algorithm is heavily influenced by the type of similarity measure metric used, with most researchers using distance-based metrics. This paper aims to enhance DBSCAN clustering by using a pattern-based metric known as the context similarity coefficient (CSC) instead of distance-based metrics. The CSC metric examines received signal strength (RSS) measurement patterns that form fingerprint vectors and assesses both linear and non-linear relationships between these vectors to determine similarity. Four publicly available fingerprint databases were used to evaluate the clustering performance with silhouette scores as a performance metric. The performance of the DBSCAN algorithm with the CSC metric is determined and compared to Euclidean and Manhattan distances as similarity measure metrics. Simulation results indicate that achieving good clustering performance with the DBSCAN algorithm requires generating three or fewer clusters. The proposed CSC metric demonstrated the best clustering performance in two of four fingerprint databases and the second-best in another. However, computational complexity comparisons reveal that the CSC metric is highly computationally intensive and is suggested to be used on small to medium-sized fingerprint databases generated using an odd number of wireless APs deployed in a non-uniform or non-grid-like distribution.

INDEX TERMS Clustering, context similarity coefficient, DBSCAN, distance-based metrics, fingerprinting, RSS.

I. INTRODUCTION

Fingerprint-based localization is a type of wireless localization system that uses position-dependent signal parameters, such as received signal strength (RSS), with a localization matching algorithm to determine the location of the target [1], [2]. This type of system has become increasingly important, especially for indoor applications such as indoor

The associate editor coordinating the review of this manuscript and approving it for publication was Zahid Akhtar¹.

navigation and asset tracking, due to the high signal attenuation within the door environment associated with the global positioning system (GPS) [2]. The fingerprint-based localization system estimates the location of an indoor user using a two-phase process, namely, offline and online phases [3]. The offline phase involves the creation of a fingerprint database by first collecting RSS measurements from spatially deployed wireless access points (APs) at predetermined locations termed reference locations (RL) [2]. The RSS measurements collected at each RL form what is known as a

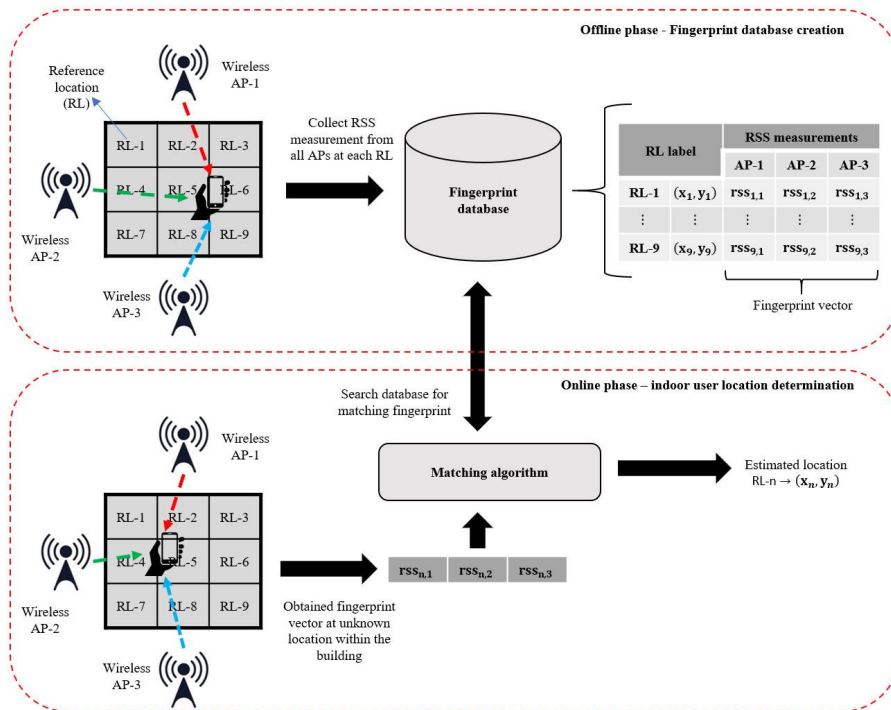


FIGURE 1. Fingerprint localization process with three wireless APs.

fingerprint vector and are stored in the database along with the corresponding RLs. In the online phase, first, the indoor user captures and generates a fingerprint vector. The system then compares the newly generated fingerprint vector to the fingerprint vectors in the database to identify the closest match. The indoor user location is estimated based on the RL of the closest-matched fingerprint vector in the database [4]. Figure 1 summarizes how fingerprint localization takes place using three wireless APs.

The fingerprint database, especially its density, plays a key role in determining the localization accuracy as well as the localization time of the fingerprint-based system [2], [5]. The density of the fingerprint database refers to the number of wireless APs and RLs used in its creation. The higher the density, i.e., the more wireless APs and RLs are used, the longer the localization time; that is, it takes longer to search through the database to identify the best matching fingerprint vector [5]. Fingerprint database clustering is introduced as a means to solve this trade-off. Clustering is the process of grouping fingerprint vectors into clusters using a common shared feature between the fingerprint vectors. This common sharing feature is known as the fingerprint similarity metric [6], [7]. Several clustering algorithms, such as k-means [8], fuzzy c-means (FCM) [9], and affinity propagation clustering (APC) [10], have been used to carry out fingerprint database clustering, and each has its advantages and disadvantages over the other. In this paper, the density-based clustering algorithm known as density-based spatial clustering of applications with noise (DBSCAN) is

considered due to its robustness to fingerprint vector outliers and ability to handle databases with arbitrary shapes [11].

The clustering performance of the DBSCAN algorithm is dependent on several factors, which include the determination of the minimum number of fingerprint vectors needed to form a cluster (MinPts), the similarity threshold between fingerprint vectors (epsilon, ϵ), and the fingerprint similarity measure metric used [6], [12]. The fingerprint similarity measure metric used plays a vital role in the accuracy of similarity determination between fingerprint vectors as well as the creation of distinct and well-separated clusters. This ultimately results in improved clustering performance. Most researchers used distance-based metrics as similarity metrics with the DBSCAN algorithm; however, in this paper, a pattern-based similarity metric known as context similarity coefficient (CSC) is proposed and used. Unlike the distance-based similarity metric, which uses the distance between fingerprint vectors to determine their similarity, the CSC metric uses RSS measurement patterns in each fingerprint vector to measure similarity [3], [13]. As such, this paper aims to improve the clustering performance of the DBSCAN algorithm for fingerprint-based indoor localization by using a pattern-based metric as a similarity measure. This paper makes the following contributions: (a) It investigates the feasibility of using the CSC metric as a fingerprint similarity measure with the DBSCAN algorithm and provides insights into its operational viability. This investigation aims to assess whether incorporating the CSC metric can enhance the algorithm’s efficiency and effectiveness in clustering fingerprints. (b) It assesses

the effectiveness of the CSC metric in comparison to traditional distance-based metrics to demonstrate its superior performance in clustering operations within the context of fingerprint-based indoor localization. This comparison seeks to determine the advantages of the CSC metric, including any improvements in accuracy and reliability that it may offer over conventional distance-based metrics.

The subsequent sections of the paper are organized as follows: Section II provides an overview of the DBSCAN algorithm clustering methodology as well as a review of related works. The mathematical description of the pattern-based similarity measure metric is presented in Section III, with the simulation results and discussion in Section IV. The conclusion is presented in Section V.

II. DBSCAN ALGORITHM CLUSTERING METHODOLOGY AND REVIEW OF RELATED WORK

As previously mentioned, this paper employs the DBSCAN algorithm for clustering the fingerprint database. In this section, the clustering methodology of the DBSCAN algorithm is first presented. This is followed by a review of related works on the different similarity measure metrics used with the DBSCAN algorithm.

A. DBSCAN ALGORITHM CLUSTERING METHODOLOGY

The DBSCAN algorithm is a popular unsupervised clustering algorithm known for its robustness to fingerprint vector outliers and ability to discover clusters of arbitrary shapes within the database. It has an advantage over the k-means algorithm as it is robust to fingerprint vector outliers, can identify clusters of any shape, and does not require the number of clusters to be generated to be predefined [14]. In comparison to the APC algorithm, the DBSCAN algorithm is easier to interpret. Furthermore, it requires tuning only two parameters (ϵ and MinPts) [15], unlike the APC algorithm, which requires setting preference values for all fingerprint vectors, making it more complex to configure.

The summary of the clustering methodology of the DBSCAN algorithm is presented below [14], [15]:

Step 1: Parameter setup and initialization

1. Define the two important parameters, namely, ϵ and MinPts.

Step 2: Identification of neighbors and core fingerprint vectors:

1. Determine the similarity value of all possible fingerprint vector pairs in the database.
2. For each fingerprint vector (\mathbf{f}_i), identify the fingerprint vectors with a similarity value less than or equal to ϵ .
3. If the number of fingerprint vectors (including \mathbf{f}_i) is greater or equal to MinPts, mark \mathbf{f}_i as a core fingerprint vector.

Step 3: Cluster expansion:

1. For each core fingerprint vector, \mathbf{f}_i , retrieve all its ϵ -neighbourhoods, including \mathbf{f}_i .

2. If the neighbouring fingerprint vector, \mathbf{f}_j is also a core fingerprint vector, recursively fine-tune and add its ϵ -neighbourhood fingerprint vectors to the cluster with \mathbf{f}_i as the core fingerprint vector.
3. If \mathbf{f}_j is not yet assigned to any cluster, assign it to the current cluster.

Step 4: Handling border fingerprint vectors:

1. If a fingerprint vector is in the ϵ -neighbourhood of a core fingerprint vector but itself is not a core fingerprint vector (i.e., it has fewer than MinPts neighbours), mark it as a border fingerprint vector.
2. Border fingerprint vectors are assigned to the cluster of a nearby core fingerprint vector when applicable.

Step 5: Identification of noise fingerprint vectors:

1. Fingerprint vectors that are neither core nor border fingerprint vectors are considered noise fingerprint vectors. These fingerprint vectors do not belong to any cluster.

Step 6: Formation of clusters:

1. As the algorithm progresses, a cluster is formed by each core fingerprint vector and its connected fingerprint vectors.
2. All noise fingerprint vectors remain unassigned.

Step 7: Termination and output:

1. The process continues until all fingerprint vectors have been assigned to clusters or identified as noise.
2. The algorithm outputs a set of clusters—groups of fingerprint vectors identified as dense regions—and a set of noise fingerprint vectors considered fingerprint vector outliers.

Steps 1 through 7 summarize the steps taken to perform clustering using the DBSCAN algorithm. During the identification of core fingerprint vector ϵ -neighborhoods in Step 2 above, the similarity measure metric is used. In the next section, a review of work on the different similarity measure metrics used with the DBSCAN algorithm is presented.

B. REVIEW OF RELATED WORKS

As mentioned earlier, the similarity measure metric used to determine the similarity of fingerprint vector pairs influences the clustering performance of the DBSCAN algorithm. Several research studies have used different similarity measure metrics to perform clustering with the DBSCAN algorithm [12], [16], [17], [18], [19], [20], [21], [22], and [23]. For instance, the authors in [16], [17], [18], [19], [20] used Euclidean distance as a similarity measure metric to evaluate the clustering performance of the DBSCAN algorithm, with each author using different MinPts and ϵ values. The Euclidean distance is the most widely used similarity measure metric within the context of fingerprint database clustering [24]. It measures the shortest straight-line

distance between fingerprint vectors, and fingerprint vector pairs with the smallest Euclidean distance value are considered to be similar. Another metric based on the distance between fingerprint vectors is the Manhattan distance, which has been used by the authors in [12], [20], and [21] to determine the similarity between fingerprint vectors for clustering with the DBSCAN algorithm. The Manhattan distance is the second most commonly used distance-based metric and is determined by finding the sum of the absolute differences in RSS measurements across each fingerprint vector [24]. In [22], hamming distance was used as a similarity measure to determine the similarity of fingerprint vector pairs with the DBSCAN algorithm as the clustering algorithm. Hamming distance measures the minimum number of substitutions required to change one binary representation of a fingerprint vector into another. The Spearman distance is another form of distance-based metric that has been used as a similarity measure with the DBSCAN algorithm [23]. It is determined by first converting the raw values of each fingerprint vector into ranks. Then, the square root of the sum of the squared differences between the ranks of the RSS measurements for the fingerprint vector pair is computed. A summary of the different works on DBSCAN clustering using different similarity measure metrics is shown in Table 1.

TABLE 1. Summary of comparisons of related works on the DBSCAN algorithm with different similarity measure metrics.

Reference work	Clustering parameter	
	Similarity measure metric	MinPts and ϵ values
[16]	Euclidean distance	
[17]	Euclidean distance	MinPts = 1; $\epsilon = 2$
[18]	Euclidean distance	MinPts = 5; $\epsilon = 2$
[19]	Euclidean distance	MinPts = 2, $\epsilon = 40.4$
[20]	Euclidean and Manhattan distances	
[12]	Manhattan distance	MinPts = 2 to 4, $\epsilon =$ knee point algorithm
[21]	Manhattan distance	MinPts = 1, $\epsilon = 300$
[22]	Hamming distance	MinPts = 5, $\epsilon = 20$
[23]	Spearman distance	MinPts = 1, $\epsilon = 0.22$

From Table 1, it can be seen that most research work focuses on using distance-based metrics as similarity measure metrics for the DBSCAN algorithm. The Euclidean and Manhattan distances are the most commonly used. In [12] and [20], it was shown that the performance of the DBSCAN algorithm varies with the similarity measure metric used and also varies with fingerprint database structural characteristics, respectively. A promising metric that has not been fully explored for use as a fingerprint vector similarity measure is the pattern-based metric, especially the CSC metric. The authors in [3] and [10] have attempted to improve the performance of the APC and k-means algorithms, respectively, using the CSC metric as a similarity measure metric. Both authors concluded that the CSC metric has the potential to improve clustering performance by generating distinct and well-separated clusters. Thus, this paper attempts to improve

the clustering performance of the DBSCAN algorithm using the CSC metric as a similarity measure. This implies that the value of ϵ used in step 2 of the DBSCAN clustering methodology outlined in Section II-A will be obtained using the CSC metric when determining the core fingerprint vector ϵ -neighborhoods. In the next section of the paper, an overview of the pattern-based metric as well as the mathematical description for the determination of the CSC metric are presented.

III. MATHEMATICAL DESCRIPTIONS OF THE PROPOSED PATTERN-BASED SIMILARITY METRIC

As previously stated, distance-based metrics such as the Euclidean and Manhattan distances are the commonly used similarity measure metrics with the DBSCAN algorithm. These metrics only take into account the actual or sum of the absolute differences between the two fingerprint vectors, i.e., how close to each other the two fingerprint vectors are. However, the behavior of each RSS measurement in the fingerprint vector as well as the linear and non-linear relationships between fingerprint vectors need to be taken into consideration during similarity determination. The pattern-based metric takes both factors into account. The pattern-based metric evaluates the similarity of two fingerprint vectors by examining how closely their RSS measurement patterns align qualitatively. To illustrate, imagine four fingerprint vectors, each with its own RSS pattern distribution, as shown in Figure 2 below.

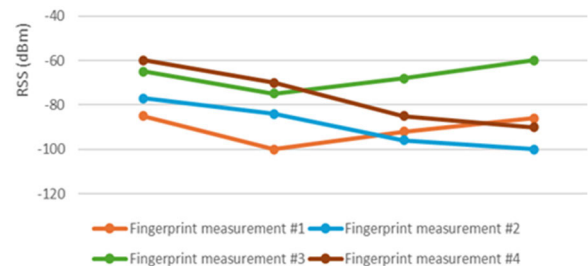


FIGURE 2. RSS measurement pattern of Fingerprint vectors.

Fingerprint vectors #1 and #3 exhibit an identical RSS measurement pattern, indicating a high degree of similarity. As such, during clustering, both fingerprint vectors #1 and #3 will be grouped within the same cluster. Similarly, fingerprint vectors #2 and #4 share an identical RSS measurement pattern, signifying a high degree of similarity, and thus will also be clustered together.

The CSC metric is one of the methods used to quantitatively assess the similarity between two fingerprint vectors, focusing on their RSS pattern. The mathematical process for calculating the CSC similarity value for a pair of fingerprint vectors is described below.

Given two fingerprint vectors, $\mathbf{f}_i(n)$ and $\mathbf{f}_j(n)$, each consisting of N RSS measurements, as shown in (1) and (2),

respectively [3] and [10]:

$$\mathbf{f}_i(n) = [rss_i(1), rss_i(2), \dots, rss_i(N)] \quad (1)$$

$$\mathbf{f}_j(n) = [rss_j(1), rss_j(2), \dots, rss_j(N)] \quad (2)$$

The CSC metric between the $\mathbf{f}_i(n)$ and $\mathbf{f}_j(n)$ pair is called as follows:

Step 1: Calculate \mathbf{t}_{ij} as shown in (3) based the vector addition of $\mathbf{f}_i(n)$ and $\mathbf{f}_j(n)$

$$\mathbf{t}_{ij}(\mathbf{n}) = [t(1), t(2), t(3), \dots, t(N-1), t(N)] \quad (3)$$

where: $t(n) = rss_i(n) + rss_j(n)$

Step 2: Compute the probability of the outcome, $p_{\mathbf{f}_i}$, for the fingerprint vector $\mathbf{f}_i(n)$ as shown in (4)

$$p_{\mathbf{f}_i} = \frac{\sum_{n=1}^N \mathbf{f}_i(n)}{\sum_{n=1}^N \mathbf{t}_{ij}(n)} \quad (4)$$

Step 3: Calculate the expected value, denoted as $\langle rss_i(n) \rangle$, for each RSS measurement within $\mathbf{f}_i(n)$ using (5).

$$\langle rss_i(n) \rangle = p_{\mathbf{f}_i} \times \mathbf{t}_{ij}(n) \text{ for } 1 \leq n \leq N \quad (5)$$

Step 4: Using (6), calculate the error for each RSS measurement in $\mathbf{f}_i(n)$.

$$error_{\mathbf{f}_i}(n) = \frac{\langle rss_i(n) \rangle - \mathbf{f}_i(n)}{\sqrt{\mathbf{t}_{ij}(n) \times p_{\mathbf{f}_i} \times (1 - p_{\mathbf{f}_i})}} \quad (6)$$

Step 5: Compute the CSC metric value between $\mathbf{f}_i(n)$ and $\mathbf{f}_j(n)$ using (7).

$$sim_{csc}(\mathbf{f}_i, \mathbf{f}_j) = \frac{\sum_{n=1}^N \left((error_{\mathbf{f}_i}(n))^2 \times \sqrt{\mathbf{t}_{ij}(n)} \right)}{\sum_n \sqrt{\mathbf{t}_{ij}(n)}} \quad (7)$$

where $error_{\mathbf{f}_i}(n)$ is obtained using (6) and $\mathbf{t}_{ij}(n)$ in (1).

The similarity value between \mathbf{f}_i and \mathbf{f}_j , calculated using (7), quantifies the degree of resemblance through the correlation among individual RSS measurements within the fingerprint vectors. A low CSC value denotes strong similarity, whereas a high value indicates significant dissimilarity. The ε value between fingerprint vectors will be calculated based on (7) and used to determine the nearest ε -neighborhood to form a cluster.

In the following section of the paper, the clustering performance of the DBSCAN algorithm with the CSC metric as a similarity measure metric is evaluated and compared with two of the commonly used distance metrics.

IV. SIMULATION RESULTS AND DISCUSSION

As previously stated, the goal of this paper is to enhance the clustering performance of the DBSCAN algorithm by employing a pattern-based metric as a similarity measure. In this section, the paper evaluates the clustering performance

of the DBSCAN algorithm using the CSC metric as a similarity measure and compares it with two commonly used distance-based metrics. It starts by presenting the simulation parameters and setup and then proceeds to compare and analyze the clustering performance.

A. SIMULATION PARAMETER AND SETUP

The clustering performance of the DBSCAN algorithm with different similarity measure metrics is evaluated across four experimentally generated RSS-based fingerprint databases: SEUG_IndoorLoc [25], IIRC_IndoorLoc [26], PIEP_UM_IndoorLoc [27], and MSI_IndoorLoc [28]. The SEUG_IndoorLoc database, created with three Wi-Fi-based APs, includes 49 RLS. The IIRC_IndoorLoc database, developed with four Zigbee-based APs, contains 194 RLS. Both SEUG_IndoorLoc and IIRC_IndoorLoc are considered low-density databases. In contrast, PIEP_UM and MSI_IndoorLoc are high-density databases utilized in the International Conferences on Indoor Positioning and Indoor Navigation (IPIN) of 2019 and 2016, respectively. The PIEP_UM database comprises 1000 RLS and was generated using eight Wi-Fi-based APs, while the MSI_IndoorLoc database consists of 4973 RLS and was generated using eleven Wi-Fi-based APs. Table 2 provides a summary of the characteristics of these four fingerprint databases.

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

Databases	Wireless technology	Database characteristics	
		No. of AP	No. of RLS/fingerprint vectors
SEUG_IndoorLoc	Wi-Fi	3	49
IIRC_IndoorLoc	Zigbee	4	194
PIEP_UM_IndoorLoc	Wi-Fi	8	1000
MSI_IndoorLoc	Wi-Fi	11	4973

The two commonly used distance metrics to be compared with the CSC metric for clustering performance are the Euclidean and Manhattan distances [24]. The clustering performance metric used for this comparison is the silhouette score, which evaluates the quality of clusters formed by any clustering algorithm [29]. The silhouette score measures how similar a fingerprint is to its own cluster (cohesion) compared to other clusters (separation). Mathematically, the silhouette score for a single fingerprint within a cluster is calculated as:

$$s(i) = \frac{b(i) - a(i)}{\max\{b(i), a(i)\}} \quad (8)$$

Where $a(i)$ is the average intra-cluster distance, which is the average distance between the fingerprint and other fingerprints within the same cluster, and $b(i)$ is the average inter-cluster distance, which is the average distance between the fingerprint and the fingerprints in the nearest cluster.

The overall silhouette score of a clustered fingerprint database is the mean of all silhouette coefficients for all fingerprint measurements in the database. Let there be M total

number of fingerprints in the database, the overall silhouette score is obtained as:

$$S = \frac{1}{M} \sum_{i=1}^M s(i) \quad (9)$$

The silhouette score in (9) ranges from 1 to -1, with a score of 1 indicating the best clustering performance [29]. Table 3 provides an overview of how to interpret silhouette scores in the context of clustering performance [29].

TABLE 3. Silhouette score interpretations for clustering evaluation.

Silhouette Score Range	Interpretation
$0.7 \leq S \leq 1$	Very good clustering performance
$0.7 < S \leq 0.25$	Moderate clustering performance
$0.25 < S \leq -1$	Poor clustering performance

As indicated in Table 3, a clustering operation resulting in a silhouette score between 1 and 0.25 is considered well-clustered, signifying distinct and well-grouped fingerprints. This enhances the reliability of matching to specific locations, reducing ambiguity and improving the overall localization accuracy of the system. It also increases the likelihood of accurate localization, as the matching algorithm can more effectively differentiate between closely situated fingerprints, thereby enhancing the robustness of the localization process. This paper uses a silhouette score threshold of 0.25, considering any score above this threshold to indicate that the DBSCAN algorithm, when using any of the fingerprint similarity metrics, achieves good clustering performance.

For the clustering performance comparison of the DBSCAN algorithm with the different similarity metrics, the number of clusters to be generated using each similarity metric is considered to be the same while their obtained silhouette scores are compared. By maintaining the same number of clusters for the different similarity metrics, a standardized and consistent basis for comparison is created. This eliminates variability that could arise from differing numbers of clusters and focuses solely on the direct assessment of the impact of each similarity metric on the clustering performance of the DBSCAN algorithm. Also, ensuring the same number of clusters helps in result interpretation and comparison, as each configuration aims to divide the fingerprint database into the same number of clusters. The implementation strategy for clustering performance comparison, taking into account different similarity metrics, is as follows:

Step 1: Determination of the target number of clusters:

For the clustering performance comparison, considering the different similarity metrics, four different cluster numbers were considered: $k = 3$, $k = 7$, $k = 10$, and $k = 15$. These specific values for k were chosen based on a review of existing literature, which indicates that these cluster sizes are commonly used in previous research works to cluster the considered fingerprint databases.

Step 2: Parameter tuning for each similarity metric:

For each similarity metric, the ϵ value is tuned to enable the DBSCAN algorithm to generate the target number of clusters. However, the MinPts value is set to 1 to eliminate the classification of fingerprint vectors as outliers; that is, a cluster with only one fingerprint vector as a member is allowed.

Step 3: Compute the silhouette score:

Once the targeted number of clusters has been obtained for each similarity metric, the silhouette score for the clustering results is determined.

Step 4: Silhouette score comparison:

The silhouette scores are compared to determine which similarity metric yields the best clustering performance.

In the next subsection of the paper, the clustering performance of the DBSCAN algorithm using the proposed CSC metric, the Euclidean distance and the Manhattan distance as similarity measure metrics is presented using the implementation strategy highlighted in steps 1 to 4.

B. DBSCAN CLUSTERING PERFORMANCE COMPARISON USING VARIOUS SIMILARITY METRICS

Using various similarity measure metrics, including the CSC metric, Euclidean distance, and Manhattan distance, the clustering performance of the DBSCAN algorithm is evaluated and compared for each of the four fingerprint databases described in Table 2. The comparison of silhouette scores for each similarity measure metric, considering different numbers of clusters, is presented in Tables 4, 5, 6, and 7 for the SEUG_IndoorLoc, IIRC_IndoorLoc, PIEP_UM_IndoorLoc, and MSI_IndoorLoc databases, respectively. As earlier mentioned, a clustering algorithm is considered to have moderate clustering performance when the clusters generated by the algorithm have an overall silhouette score within the range of 0.7 to 0.25, while poor clustering performance occurs when the silhouette score is below 0.25.

TABLE 4. Silhouette score comparison for varying cluster numbers across SEUG_IndoorLoc database.

Fingerprint database	Cluster number (k)	Similarity measure metric		
		Euclidean	Manhattan	CSC
SEUG_IndoorLoc	3	0.36	0.36	0.45
	7	0.29	0.29	0.36
	11	0.33	0.19	0.10
	15	0.29	0.27	0.11

Based on the silhouette scores obtained by each similarity measure metric for the different cluster numbers as shown in Table 4, the performance of the DBSCAN algorithm across the SEUG_IndoorLoc database ranges from poor to moderate clustering performance. The silhouette score comparison for the SEUG_IndoorLoc database shows that the CSC metric has the highest silhouette score of 0.45 and 0.36, which indicates the best clustering performance for the smaller

cluster numbers $k = 3$ and $k = 7$, respectively. However, the Euclidean distance has consistent performance across different cluster numbers, with better clustering performance for larger cluster numbers ($k = 11$ and $k = 15$). As a result, selecting the best similarity measure metric for use with the DBSCAN algorithm is determined by the number of clusters to be generated. For the SEUG_IndoorLoc database, if the number of clusters to be generated is $k = 7$ or less, the proposed CSC metric is preferable, while for a larger cluster number, the Euclidean distance metric is preferred. Considering $k = 7$ or less, the structural characteristic of the SEUG_IndoorLoc database favors the use of RSS pattern of each fingerprint to generate well structured clusters than using actual or absolute distance differences.

TABLE 5. Silhouette score comparison for varying cluster numbers across IIRC_IndoorLoc database.

Fingerprint database	Cluster number (k)	Similarity measure metric		
		Euclidean	Manhattan	CSC
IIRC_IndoorLoc	3	0.51	0.39	0.21
	7	-0.12	-0.04	-0.17
	11	-0.25	-0.26	-0.23
	15	-0.32	-0.21	-0.36

From the silhouette scores presented in Table 5, the overall performance of the DBSCAN algorithm, irrespective of the similarity measure metric and number of clusters generated, is poor. For the IIRC_IndoorLoc database, as the number of clusters increases, the performance of the DBSCAN algorithm using any of the similarity measure metrics degrades. The comparison of silhouette scores for the different similarity measure metrics and considering the different number of clusters shows that the Euclidean distance has the best clustering performance when $k = 3$, with the highest silhouette score of 0.51. This is followed by the Manhattan distance, with a silhouette score of 0.39 when $k = 3$. The proposed CSC metric has the lowest silhouette score of 0.21 when $k = 3$, placing it within the poor clustering performance range. For the remainder of the cluster numbers ($k = 7$, $k = 11$, and $k = 15$), all three similarity measure metrics performed poorly, as their silhouette scores are all below 0.25.

Overall, for the IIRC_IndoorLoc database, using the Euclidean distance as a similarity measure metric for the DBSCAN algorithm yields the best clustering results. However, this is only true for cluster numbers of $k = 3$ or fewer. The proposed CSC metric performed poorly and was not an appropriate choice for the IIRC_IndoorLoc database. This means that the IIRC_IndoorLoc database has a structure that allows for the generation of distinct clusters based on actual distance rather than RSS patterns between fingerprint vectors.

The performance of the DBSCAN algorithm using three different similarity measure metrics on the PIEP_UM_IndoorLoc database matches its performance on the IIRC_IndoorLoc database. In both cases, at $k = 3$,

TABLE 6. Silhouette score comparison for varying cluster numbers across PIEP_UM_IndoorLoc database.

Fingerprint database	Cluster number (k)	Similarity measure metric		
		Euclidean	Manhattan	CSC
PIEP_UM_IndoorLoc	3	0.27	0.32	0.27
	7	-0.15	-0.10	-0.09
	11	-0.28	-0.24	-0.25
	15	-0.30	-0.37	-0.34

all three similarity measure metrics exhibit moderate clustering performance. As the number of clusters increases, the performance of the DBSCAN algorithm on the PIEP_UM_IndoorLoc database, irrespective of the similarity measure metric used, degrades. However, in this current database, the use of the Manhattan distance metric as the similarity measure metric results in the best clustering performance with a silhouette score of 0.32. The Euclidean distance and the proposed CSC metric both came in second with a silhouette score of 0.27. For a higher number of clusters, all three different similarity measure metrics performed poorly, as their silhouette scores were below 0.25. This implies that the structure of the PIEP_UM_IndoorLoc database is better suited for utilizing absolute distance differences instead of actual distance or RSS patterns among fingerprint vectors to form well-separated clusters. Moreover, achieving moderate clustering performance with the DBSCAN algorithm on the PIEP_UM_IndoorLoc database requires limiting the number of clusters to a smaller value, specifically $k = 3$ or fewer.

TABLE 7. Silhouette score comparison for varying cluster numbers across MSI_IndoorLoc database.

Fingerprint database	Cluster number (k)	Similarity measure metric		
		Euclidean	Manhattan	CSC
MSI_IndoorLoc	3	0.40	0.40	0.47
	7	0.06	0.01	0.01
	11	-0.07	-0.17	-0.12
	15	-0.20	-0.25	-0.32

According to the silhouette scores presented in Table 7, the DBSCAN algorithm performs similarly across all three similarity metrics, consistent with its performance on the PIEP_UM_IndoorLoc and IIRC_IndoorLoc databases. That is, only at a smaller cluster number is moderate clustering performance achieved, and in this case, at $k = 3$. Unlike the PIEP_UM_IndoorLoc and IIRC_IndoorLoc databases, the use of the proposed CSC metric with the DBSCAN algorithm resulted in the best clustering performance, with the highest silhouette score of 0.47 when $k = 3$. This is followed by both the Euclidean and Manhattan distances, both with silhouette scores of 0.40 when $k = 3$. For a higher number of clusters ($k = 7$, $k = 11$, and $k = 15$), the clustering performance of the DBSCAN algorithm degrades for all the similarity measure metrics, as the silhouette scores obtained by each metric are lower than 0.25. Overall, based on the

MSI_IndoorLoc database, the findings suggest that the CSC metric is the most suitable similarity measure for optimum clustering performance with the DBSCAN algorithm. However, optimal clustering performance for the MSI_IndoorLoc database can only be achieved using a smaller cluster number. Furthermore, the result suggests that the MSI_IndoorLoc database has a structure that favors the use of an RSS pattern to measure the similarity between fingerprint vectors rather than the actual distance or absolute distance differences. It is worth mentioning that the Manhattan and Euclidean distances are strong alternatives for use as similarity measure metrics with the DBSCAN algorithm to cluster the MSI_IndoorLoc database.

Based on the result analysis for the four different databases considered, the use of a smaller number of clusters, that is, $k = 3$ or less, results in the best clustering performance of the DBSCAN algorithm, irrespective of the similarity measure metric. Furthermore, the proposed CSC metric appears to be the overall best similarity measure metric across the four fingerprint databases. It resulted in the best clustering performance in two databases (MSI_IndoorLoc and SEUG_IndoorLoc) out of the four fingerprint databases considered. It ranked second on the PIEP_UM_IndoorLoc database and third or last on the IIRC_IndoorLoc database.

The suitability of a similarity metric in generating well-defined fingerprint clusters from a fingerprint database depends on the structural and statistical characteristics of the fingerprints within that database. These characteristics are influenced by factors such as the layout of the environment—specifically the shape of the building where the fingerprint measurements are taken—and the placement of wireless APs within the indoor space. For instance, when wireless APs are uniformly distributed in a rectangular configuration with minimal overlap, Euclidean distance is an effective fingerprint similarity metric for creating distinct clusters. Conversely, Manhattan distance is better suited for databases where wireless APs are arranged in a grid-like pattern with sparse placement. Pattern-based metrics excel when dealing with fingerprint databases derived from environments with irregular shapes and non-uniform AP placement.

The SEUG_IndoorLoc and MSI_IndoorLoc databases, for which the proposed pattern-based metric has good clustering performance, are created using an odd number of wireless APs, 3 and 11, respectively. For the SEUG_IndoorLoc database, the wireless APs are deployed in a right-angle triangle shape, while for the MSI_IndoorLoc database, the placement of the wireless APs does not follow either a uniform or grid-like distribution. The placements of the wireless APs for these two databases resulted in a fingerprint database structure in which the distance and absolute distance difference between fingerprint vectors are not sufficient enough to create distinct clusters but rather the underlying RSS pattern in each fingerprint vector. This means that it is possible to use the RSS pattern as a similarity measure for the DBSCAN algorithm and be able to achieve good clustering performance in scenarios where the fingerprint database is created using

an odd number of wireless APs, placed in a non-uniform or non-grid-like distribution.

Even though clustering operations are performed once during system deployment, it is also important to consider the computational complexity of using a similarity metric. The CSC metric typically involves more complex calculations than Euclidean or Manhattan distances, leading to increased computational load and longer processing times, especially as the size and density of the fingerprint database grow. The CSC metric has a computational complexity of $O(5N)$ based on big O notation, whereas Euclidean and Manhattan distances have a complexity of $O(N)$, with N representing the number of RSS measurements in the fingerprint vector. For large fingerprint databases, the use of the CSC metric may result in significant delays in clustering operations. Longer clustering times increase latency in the localization process as well as hinder the system's ability to instantly adapt to changes in the environment, making real-time applications challenging. As such, it is recommended that the proposed CSC metric be used as a fingerprint similarity metric with the DBSCAN algorithm to cluster fingerprint databases of smaller density and also use a smaller number of clusters for maximum clustering performance.

V. CONCLUSION AND FUTURE WORKS

The clustering performance of the DBSCAN algorithm is heavily influenced by the choice of similarity metric used in the determination of the similarity between fingerprint vectors. Most researchers focus on using any of the distance-based metric variants to determine the similarity between fingerprint vectors. However, this research proposes to use a pattern-based metric as a similarity metric for the DBSCAN algorithm. This is aimed at improving the clustering performance of the DBSCAN algorithm. The clustering performance of the DBSCAN algorithm using the proposed pattern-based metric is evaluated across four experimentally generated fingerprint databases, of which two are of lower density and the other two are of higher density. The simulation results indicate that the proposed pattern-based metric can enhance clustering performance when used as a similarity measure metric for the DBSCAN algorithm and clustering fingerprint database created using an odd number of wireless APs placed in a non-uniform or non-grid-like distribution. However, several limitations must be acknowledged regarding the use of the proposed pattern-based metric as a similarity measure metric with the DBSCAN algorithm. Firstly, the proposed pattern-based metric demonstrates optimal performance primarily when fewer clusters are generated ($k = 3$ or less). This may restrict its applicability on fingerprint databases with a naturally higher number of clusters. Additionally, the computational complexity of the proposed metric is notably high, particularly in densely populated fingerprint databases, which could lead to significant processing times and resource demands. To achieve optimal clustering performance with the proposed pattern-based metric, it is recommended to use it on small to medium-sized fingerprint

databases and to generate a smaller number of clusters. Future research will aim to enhance the computational efficiency of the proposed pattern-based metric, allowing its application to larger and denser fingerprint databases without compromising clustering accuracy.

ACKNOWLEDGMENT

The authors acknowledge funding from the FIM Excellence project run at the Faculty of Informatics and Management, University of Hradec Kralove, Czech Republic. Special thanks to Ing. Daniel Schmidt for his help with data collection and preparation.

REFERENCES

- [1] A. S. Yaro, F. Maly, and 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, Feb. 2023, doi: [10.3390/s23052545](https://doi.org/10.3390/s23052545).
- [2] P. Sadhukhan, "Performance analysis of clustering-based fingerprinting localization systems," *Wireless Netw.*, vol. 25, no. 5, pp. 2497–2510, Jul. 2019, doi: [10.1007/s11276-018-1682-7](https://doi.org/10.1007/s11276-018-1682-7).
- [3] A. S. Yaro, F. Maly, P. Prazak, and K. Malý, "Enhancing fingerprint localization accuracy with inverse weight-normalized context similarity coefficient-based fingerprint similarity metric," *IEEE Access*, vol. 12, pp. 73642–73651, 2024, doi: [10.1109/ACCESS.2024.3405350](https://doi.org/10.1109/ACCESS.2024.3405350).
- [4] S. Shang and L. Wang, "Overview of WiFi fingerprinting-based indoor positioning," *IET Commun.*, vol. 16, no. 7, pp. 725–733, Apr. 2022, doi: [10.1049/cmu2.12386](https://doi.org/10.1049/cmu2.12386).
- [5] A. S. Yaro, F. Maly, P. Prazak, and K. Malý, "Improved fingerprint-based localization based on sequential hybridization of clustering algorithms," *Emerg. Sci. J.*, vol. 8, no. 2, pp. 394–406, Apr. 2024, doi: [10.28991/ESJ-2024-08-02-02](https://doi.org/10.28991/ESJ-2024-08-02-02).
- [6] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Ann. Data Sci.*, vol. 2, no. 2, pp. 165–193, Jun. 2015, doi: [10.1007/s40745-015-0040-1](https://doi.org/10.1007/s40745-015-0040-1).
- [7] G. Minaev, A. Visa, and R. Piché, "Comprehensive survey of similarity measures for ranked based location fingerprinting algorithm," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–4, doi: [10.1109/IPIN.2017.8115922](https://doi.org/10.1109/IPIN.2017.8115922).
- [8] M. Ahmed, R. Seraj, and S. M. S. Islam, "The k-means algorithm: A comprehensive survey and performance evaluation," *Electronics*, vol. 9, no. 8, p. 1295, Aug. 2020, doi: [10.3390/electronics9081295](https://doi.org/10.3390/electronics9081295).
- [9] Y. Sun, Y. Xu, L. Ma, and Z. Deng, "KNN-FCM hybrid algorithm for indoor location in WLAN," in *Proc. 2nd Int. Conf. Power Electron. Intell. Transp. Syst. (PEITS)*, vol. 2, Dec. 2009, pp. 251–254, doi: [10.1109/PEITS.2009.5406793](https://doi.org/10.1109/PEITS.2009.5406793).
- [10] A. S. Yaro, F. Malý, and 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](https://doi.org/10.1109/ACCESS.2023.3283592).
- [11] V. Neerugatti, M. Moni, and R. M. Reddy A, "Density based spatial clustering application with noise by varying densities," *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 5881–5886, Nov. 2019, doi: [10.35940/ijrte.D8757.118419](https://doi.org/10.35940/ijrte.D8757.118419).
- [12] D. Quezada-Gaibor, L. Klus, J. Torres-Sospedra, E. S. Lohan, J. Nurmi, and J. Huerta, "Improving DBSCAN for indoor positioning using Wi-Fi radio maps in wearable and IoT devices," in *Proc. 12th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Oct. 2020, pp. 208–213, doi: [10.1109/ICUMT51630.2020.9222411](https://doi.org/10.1109/ICUMT51630.2020.9222411).
- [13] A. Kulkarni, V. Tokekar, and P. Kulkarni, "Discovering context of labeled text documents using context similarity coefficient," *Proc. Comput. Sci.*, vol. 49, pp. 118–127, Jan. 2015, doi: [10.1016/j.procs.2015.04.235](https://doi.org/10.1016/j.procs.2015.04.235).
- [14] H. K. Kanagala and V. V. Jaya Rama Krishnaiah, "A comparative study of K-means, DBSCAN and OPTICS," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2016, pp. 1–6, doi: [10.1109/ICCCI.2016.7479923](https://doi.org/10.1109/ICCCI.2016.7479923).
- [15] J. Bi, H. Cao, Y. Wang, G. Zheng, K. Liu, N. Cheng, and M. Zhao, "DBSCAN and TD integrated Wi-Fi positioning algorithm," *Remote Sens.*, vol. 14, no. 2, p. 297, Jan. 2022, doi: [10.3390/rs14020297](https://doi.org/10.3390/rs14020297).
- [16] F. Zou and H. Zhai, "Browser fingerprinting identification using incremental clustering algorithm based on autoencoder," in *Proc. IEEE 23rd Int. Conf. High Perform. Comput. Commun., 7th Int. Conf. Data Sci. Syst., 19th Int. Conf. Smart City, 7th Int. Conf. Dependability Sensor, Cloud Big Data Syst. Appl. (HPCC/DSS/SmartCity/DependSys)*, Dec. 2021, pp. 525–532, doi: [10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00093](https://doi.org/10.1109/HPCC-DSS-SmartCity-DependSys53884.2021.00093).
- [17] F. Cheng, G. Niu, Z. Zhang, and C. Hou, "Improved CNN-based indoor localization by using RGB images and DBSCAN algorithm," *Sensors*, vol. 22, no. 23, p. 9531, Dec. 2022, doi: [10.3390/s22239531](https://doi.org/10.3390/s22239531).
- [18] J. Li, I. Tobore, Y. Liu, A. Kandwal, L. Wang, and Z. Nie, "Non-invasive monitoring of three glucose ranges based on ECG by using DBSCAN-CNN," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 9, pp. 3340–3350, Sep. 2021, doi: [10.1109/JBHI.2021.3072628](https://doi.org/10.1109/JBHI.2021.3072628).
- [19] M. Zhang and H. Liao, "Privacy-preserving DBSCAN clustering algorithm based on negative database," in *Proc. 5th IEEE Int. Conf. Big Data Analytics (ICBDA)*, May 2020, pp. 209–213, doi: [10.1109/ICBDA49040.2020.9101288](https://doi.org/10.1109/ICBDA49040.2020.9101288).
- [20] P. Chen, F. Liu, S. Gao, P. Li, X. Yang, and Q. Niu, "Smartphone-based indoor fingerprinting localization using channel state information," *IEEE Access*, vol. 7, pp. 180609–180619, 2019, doi: [10.1109/ACCESS.2019.2958957](https://doi.org/10.1109/ACCESS.2019.2958957).
- [21] A. Blaise, M. Bouet, V. Conan, and S. Secci, "BotFP: FingerPrints clustering for bot detection," in *Proc. NOMS IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–7, doi: [10.1109/NOMS47738.2020.9110420](https://doi.org/10.1109/NOMS47738.2020.9110420).
- [22] Z. Xiaolin, Z. Yiman, L. Xuhui, and C. Quanbao, "Research on malicious code homology analysis method based on texture fingerprint clustering," in *Proc. 17th IEEE Int. Conf. Trust. Secur. Privacy Comput. Commun., 12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 1914–1921, doi: [10.1109/TrustCom/BigDataSE.2018.00291](https://doi.org/10.1109/TrustCom/BigDataSE.2018.00291).
- [23] G. Shtar, B. Shapira, and L. Rokach, "Clustering Wi-Fi fingerprints for indoor-outdoor detection," *Wireless Netw.*, vol. 25, no. 3, pp. 1341–1359, Apr. 2019, doi: [10.1007/s11276-018-1753-9](https://doi.org/10.1007/s11276-018-1753-9).
- [24] A. Yaro, M. Filip, K. Maly, and P. Prazak, "Clustering performance analysis of the K-Medoids algorithm for improved fingerprint-based localization," *Jordan J. Electr. Eng.*, vol. 10, no. 3, p. 1, Jan. 2024, doi: [10.5455/jjee.204-1703256698](https://doi.org/10.5455/jjee.204-1703256698).
- [25] S. Sadowski, P. Spachos, and K. N. Plataniotis, "Memoryless techniques and wireless technologies for indoor localization with the Internet of Things," *IEEE Internet Things J.*, vol. 7, no. 11, pp. 10996–11005, Nov. 2020, doi: [10.1109/JIOT.2020.2992651](https://doi.org/10.1109/JIOT.2020.2992651).
- [26] T. Alhmiedat, "Fingerprint-based localization approach for WSN using machine learning models," *Appl. Sci.*, vol. 13, no. 5, p. 3037, Feb. 2023, doi: [10.3390/app13053037](https://doi.org/10.3390/app13053037).
- [27] J. Torres-Sospedra, A. Moreira, G. M. Mendoza-Silva, M. J. Nicolau, M. Matey-Sanz, I. Silva, J. Huerta, and C. Pendão, "Exploiting different combinations of complementary sensor's data for fingerprint-based indoor positioning in industrial environments," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8, doi: [10.1109/IPIN.2019.8911758](https://doi.org/10.1109/IPIN.2019.8911758).
- [28] A. Moreira, I. Silva, F. Meneses, M. J. Nicolau, C. Pendão, and J. Torres-Sospedra, "Multiple simultaneous Wi-Fi measurements in fingerprinting indoor positioning," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–8, doi: [10.1109/IPIN.2017.8115914](https://doi.org/10.1109/IPIN.2017.8115914).
- [29] K. R. Shahapure and C. Nicholas, "Cluster quality analysis using silhouette score," in *Proc. IEEE 7th Int. Conf. Data Sci. Adv. Anal. (DSAA)*, Oct. 2020, pp. 747–748, doi: [10.1109/DSAA49011.2020.00096](https://doi.org/10.1109/DSAA49011.2020.00096).



ABDULMALIK SHEHU YARO was born in Kaduna, Nigeria, in 1990. He received the B.Eng. degree in electrical engineering from Ahmadu Bello University (ABU), Zaria, and the M.Eng. degree in telecommunications engineering and the Ph.D. degree in electrical engineering from the University of Technology Malaysia, in 2014 and 2018, respectively. Currently, he is a Senior Researcher with the Department of Informatics and Quantitative Methods, University of Hradec

Králové, Czech Republic, and an Assistant Professor with the Department of Electronics and Telecommunications Engineering, ABU. He has published over 20 papers in international peer-reviewed journals and conference proceedings. His research interests include wireless localization and sensor networks.



FILIP MALÝ was born in Czech Republic, in 1981. He received the master's degree in information management from the University of Hradec Králové, Czech Republic, in 2005, and the Ph.D. degree in information and knowledge management, in 2009. He has been an Associate Professor of systems engineering and informatics with the University of Hradec Králové, since 2014. Currently, he is a Senior Researcher and a Teacher with the Department of Informatics and Quantitative Methods, University of Hradec Králové. His research interests include indoor localization, wireless sensor networks, programming languages, and operating systems.



KAREL MALÝ was born in Czech Republic, in 1973. He received the master's degree in cybernetics and the Ph.D. degree in artificial intelligence and biocybernetics from the Faculty of Electrical Engineering, Czech Technical University (CTU), Prague, Czech Republic, in 1998 and 2005, respectively. Since 2000, he has been an Assistant Professor and a Researcher with the Department of Informatics and Quantitative Methods, University of Hradec Králové, Czech Republic.

His research interests include artificial intelligence, robotics, indoor localization, programming languages, and Unix operating systems.



PAVEL PRAZAK was born in Czech Republic, in 1968. He received the master's degree in teaching mathematics and physics and the Ph.D. degree in general problems of mathematics and informatics from the Faculty of Mathematics and Physics, Charles University, Prague, in 1996 and 2006, respectively. He has been an Associate Professor of systems engineering and informatics with the University of Hradec Králové, Czech Republic, since 2014. Currently, he is the Head of the Department of Informatics and Quantitative Methods, Faculty of Informatics and Management, University of Hradec Králové. His research interests include indoor localization, dynamical systems and optimization in economics, and applications of statistics.

...