A Fast and Precise Indoor Positioning System Based on Deep Embedded Clustering

Hui Wen Khoo[1], Yin Hoe Ng[1](✉), and Chee Keong Tan[2]

[1] Faculty of Engineering, Multimedia University, 63100 Cyberjaya, Malaysia
yhng@mmu.edu.my

[2] School of Information Technology, Monash University Malaysia, 47500 Subang Jaya, Malaysia

Abstract. In indoor positioning, the real-world scenario often involves a multi-floor indoor environment, resulting in the construction of a large Bluetooth low energy (BLE) fingerprint database. Subsequently, high computational complexity and increased computational time are usually associated with such a large scale indoor environment. To circumvent this issue, a clustering-based indoor positioning system (IPS) known as DECIPS is proposed in this paper to reduce the computational complexity and execution time required by the localization algorithm for location prediction. The proposed DECIPS leverages on deep embedded clustering (DEC) algorithm to group the dataset into several subsets before using them to train separate classifiers and regressors specifically customized to handle only data from one cluster. Subsequently, the performance of DECIPS is benchmarked against several state-of-the-art clustering-based IPSs. Numerical results demonstrate that DECIPS is capable of outperforming the existing clustering-based IPSs in terms of average positioning error and execution time.

Keywords: Indoor positioning · BLE fingerprint · deep embedded clustering

1 Introduction

In this globalized era, the demand for an accurate and real-time IPS rises to fulfill the need for indoor location-based services (LBS) which are popularly used in sectors such as hospitals, indoor parking lots, airports and shopping malls for location identification and indoor navigation [1, 2]. While current mature technology such as the global positioning system (GPS) is widely used for outdoor navigation and positioning, it is nevertheless not suitable for indoor localization purposes due to the requirement for a direct line of sight (LOS) between the satellites and the user [1–3].

In view of that, various approaches such as Bluetooth, radio frequency identification (RFID), ultra-wideband (UWB), geomagnetism, visible light and Wi-Fi are considered and researched for their potential applications in the field of indoor positioning [2, 3]. Nonetheless, among the many existing technologies, the received signal strength (RSS) based fingerprinting method has garnered the most attention since it does not require any additional hardware besides the readily available Wi-Fi access points (APs)

M. Y. bin Alias (Ed.): MECON 2022, AER 214, pp. 38–48, 2023.
https://doi.org/10.2991/978-94-6463-082-4_6

or BLE beacons and mobile devices with built-in network interface card (NIC) for RSS measurements [2, 4].

Generally, the RSS based fingerprinting IPS involves the offline and online phases. In the offline phase, a site survey must first be performed at the indoor environment to construct a radio map which contains the location labeled RSS measurements from surrounding APs or beacons at specific reference points (RPs). Meanwhile, in the online phase, the RSS values measured from APs or beacons that can be detected from the user's unknown location creates a test sample which will then be compared with the RSS stored in the constructed radio map via a localization algorithm, thus predicting the user's current location [2, 3].

Larger space in an indoor environment will correspondingly result in a larger fingerprint database being constructed. Hence, applying a localization algorithm directly over the large fingerprint database causes high computational overhead and also increased computational time. To overcome this issue, various clustering based fingerprinting method have been proposed. Instead of comparing the observed RSS values at the user's unknown location with the RSS values at all RPs, a narrowed down search is performed on only a particular cluster that contains a smaller number RPs for location prediction. Apart from reducing the overall computational complexity and computational time, a classifier or regressor trained per cluster would also be able to learn the data subset dynamics better [5].

In this paper, a clustering-based IPS called DECIPS is proposed to overcome the high computational complexity and long computational time caused by a large dataset. DECIPS adopts the DEC clustering algorithm to partition the large dataset into several subsets before performing location prediction using separate localization algorithms explicitly established for each cluster.

The rest of the paper is organized as follows. Section 2 presents the background and related work on clustering-based IPS, while Sect. 3 elaborates on the proposed DEC clustering-based IPS known as DECIPS. Thereafter, the performance evaluation and discussion of findings are presented in Sect. 4. Finally, the conclusion is made in Sect. 5.

2 Background and Related Work

In this section, the related work on clustering based IPSs and DEC clustering are introduced.

2.1 Related Work

Ezhumalai et al. proposed a strongest AP (SAP) based clustering technique that labels each RP according to the AP that yields the strongest RSS value for that particular RP. The RPs are then grouped into several clusters subsequently by using the SAP labels. This clustering technique enables the SAP information based clustering results to be in-line with the position distribution of the RPs. The weighted K-nearest neighbor (WKNN) algorithm is then utilized to estimate the user's unknown location [2].

In [6], Altintas et al. developed an improved RSS based indoor positioning algorithm via K-Means clustering. After the K-nearest neighbor (KNN) algorithm had identified a set of RPs as the K nearest neighbors, they would then be classified into k number of clusters. The cluster with the closest proximity to the user would be identified as the delegate cluster whose centroid would be used for the indoor position estimation. Nevertheless, this technique did not help to reduce the overall computational complexity and time of the IPS since clustering is only performed after localization.

However, the K-Means clustering based IPS as described in [6] belongs to a type of hard clustering which divides each observation into exactly one cluster only. Hard clustering is unable to group the Wi-Fi fingerprints of a radio map into clusters that are robust to noises such as multipath interference. Thus, fuzzy C-means (FCM), which is a type of soft clustering is proposed in [7] to compensate for the environmental effects besides reducing the online computational time. The membership grade allows the realization of soft clustering such that a Wi-Fi fingerprint can belong to more than one cluster simultaneously. KNN algorithm is then utilized to estimate the user's unknown location after performing soft clustering via FCM.

An IPS based on a deep neural network (DNN) integrated with an improved KNN algorithm was presented in [8]. Firstly, the indoor environment is separated into 4 sections, thus splitting the Wi-Fi fingerprint database into 4 clusters. The trained DNN algorithm is then used to predict the cluster of the user's unknown location. Next, weights are given to the K nearest neighbors according to their number of matching APs. The final user position is predicted among all the K nearest neighbors in the same cluster. This technique is a supervised machine learning technique that requires manual separation of the indoor environment into k number of clusters.

Akram et al. proposed a Wi-Fi fingerprinting based IPS named HybLoc which integrated the Gaussian mixture model (GMM) based soft clustering and random decision forest (RDF) ensembles for indoor positioning [5]. GMM based soft clustering splits the Wi-Fi fingerprint database of each building into overlapping or non-overlapping data subsets, thus achieving soft clustering. The RDF ensembles combine the strength of many weak learners to improve the overall indoor positioning accuracy and the generalization capability. The clustered data subsets are then assigned to different RDF ensembles established to handle the respective data subsets so that they could learn the underlying data dynamics better.

2.2 Deep Embedded Clustering (DEC)

The clustering algorithm to be applied in the proposed DECIPS is known as DEC. In recent years, DEC has been applied in various fields such as coral reef bioacoustic detection and the characterization of the patient clusters at high risk of mortality and kidney injury apart from exploration of the striatal functional connectivity alterations associated with Parkinson's Disease [9–11]. As of thus far, there is no existing work that uses DEC for the clustering of RSS data of the fingerprint database in the area of indoor positioning yet.

The first phase of DEC focuses on parameters and centroids initialization with a deep autoencoder. In this phase, a stacked autoencoder (SAE) network is initialized layer-wise with each layer functioning as a denoising autoencoder having a 2-layer neural network defined in (1) to (4) below. The denoising autoencoder attempts to reconstruct the output of previous layer after suffering random corruption [12].

$$\widetilde{x} \sim Dropout(x) \tag{1}$$

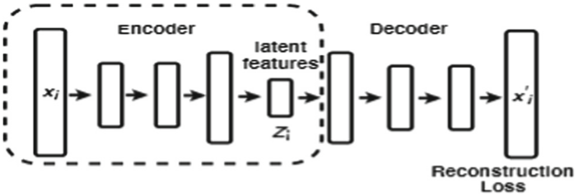


Fig. 1. DEC autoencoder network [12]

$$h = g_1(W_1\widetilde{x} + b_1) \tag{2}$$

$$\widetilde{h} \sim Dropout(h) \tag{3}$$

$$y = g_2(W_2\widetilde{h} + b_2) \tag{4}$$

where $Dropout(\cdot)$ arbitrarily sets some of its input dimensions to 0, g_1 and g_2 are the activation functions for the encoding and decoding layers respectively and $\theta = \{W_1, b_1, W_2, b_2\}$ are the model parameters.

The training is carried out by minimizing the least-squares loss. After training a layer, its output h is used as the input for the training of the next layer. At the end of the greedy layer-wise training, all encoder layers are concatenated followed by the decoder layers to form a multilayer deep autoencoder and fine-tuned to minimize the reconstruction loss, as shown in Fig. 1. The decoder layers are removed and the encoder layers are used for initial non-linear mapping between the data space and feature space to obtain the embedded data points. The initial centroids are obtained via K-means clustering in the feature space [12].

In the second phase, parameter optimization is performed in 2 steps. First, a soft assignment between the embedded data points and the cluster centroids is computed. Subsequently, with the aid of the auxiliary target distribution, the model learns from current high confidence soft assignments in order to update the deep mapping and refine the cluster centroids. In this self-training strategy, the dataset is labeled with a classifier to train on its high confidence predictions [12].

The encoder is fine-tuned by optimizing the objective which is defined as a Kullback-Leibler (KL) divergence loss between the soft assignments and the auxiliary target distribution as shown in (5):

$$L = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{5}$$

where q_{ij} is the similarity between the embedded data point z_i and cluster center μ_j measured by Student's t-distribution as defined in (6):

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2)^{-1}}{\sum_j (1 + ||z_i - \mu_j||^2)^{-1}} \tag{6}$$

while p_{ij} is the target distribution defined in (7):

$$p_{ij} = \frac{q_{ij}^2 / \sum_i q_{ij}}{\sum_j (q_{ij}^2 / \sum_i q_{ij})} \tag{7}$$

The joint optimization of the cluster centers and the encoder mapping are carried out using stochastic gradient descent (SGD) with momentum. The gradients of L with respect to the feature space embedding of each data point $\frac{\partial L}{\partial z_i}$ and cluster center $\frac{\partial L}{\partial \mu_j}$ could be determined during backpropagation. The gradient $\frac{\partial L}{\partial z_i}$ is then passed down to update the encoder mapping while $\frac{\partial L}{\partial \mu_j}$ is used to update the cluster center μ_j as shown in (8):

$$\mu_j = \mu_j - \lambda \frac{\partial L}{\partial \mu_j} \tag{8}$$

For discovering cluster assignments, the procedure is repeated until the convergence criterion is met, whereby less than *tol%* of points change their cluster assignments between 2 consecutive iterations [12].

3 Proposed Clustering-Based IPS

The block diagram for the proposed DECIPS is depicted in Fig. 2. After the site survey, the FOE BLE fingerprint dataset is constructed. Nevertheless, this dataset requires some data preprocessing since the coordinates are initially recorded in the form of a mixture of categorical and numerical variables for the x-coordinates and y-coordinates respectively. Thus, the x-coordinates must first be represented in the form of a numerical variable. Apart from that, there are also several rows of BLE fingerprints that contain outlier RSS values that happened due to human errors made during RSS data collection or data entry. In view of that, those instances with outliers are removed since they occupy only a negligible portion of the entire dataset.

Subsequently, the preprocessed dataset is split into 2 subsets, namely the training and the testing datasets. During the training phase, the RSS vectors of the RPs recorded in the training dataset are provided as the input to the DEC clustering algorithm in

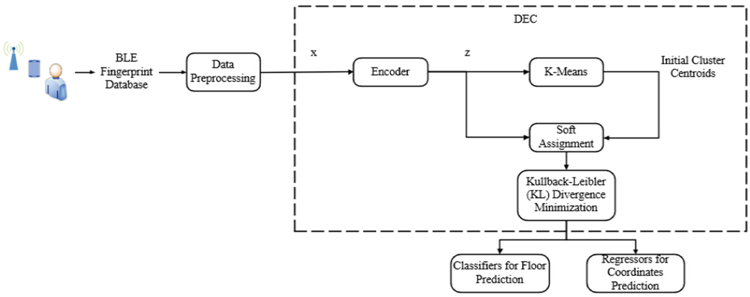


Fig. 2. Block diagram for DECIPS

its original data space x. The encoder layers in the autoencoder performs nonlinear mapping of the input RSS vectors between the data space x and the feature space z, thus producing embedded RSS data points. The process is then followed by K-Means clustering carried out in the feature space z to obtain k initial cluster centroids $\{\mu_j\}_{j=1}^{k}$. With the embedded RSS data points and initial cluster centroids obtained in the first phase of DEC, the second phase continues to alternate between two major steps in order to further improve the clustering. In the first step, soft assignment is performed between the embedded RSS data points and the initial cluster centroids. Subsequently, the nonlinear mapping is updated and the cluster centroids are refined iteratively by minimizing the KL divergence loss between the soft assignments and the auxiliary target distribution until the convergence criterion is met. The output of the DEC clustering algorithm is the cluster labels for the RSS vectors of each RP.

The cluster labeled training dataset will be split into several data subsets and then used to train separate WKNN classifiers for floor prediction and WKNN regressors for coordinates prediction that are established to handle only data subsets from a particular cluster. Meanwhile, during the testing phase, cluster matching is performed for each instance in the testing dataset using the trained DEC clustering algorithm in order to select one cluster as its delegate cluster. Soon after, the location of each clustered testing instance is predicted using the corresponding WKNN classifier and WKNN regressor trained specifically for that cluster.

The WKNN classifiers and regressors first determine the K nearest neighbors by calculating the distance between the RSS of the test sample X and the RSS of each training sample Y as shown in (9):

$$d(X, Y) = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}} \tag{9}$$

where x_i refers to the RSS measured from the ith beacon for the test sample while y_i refers to the RSS measured from the ith beacon for the training sample, n refers to the total number of beacons and p is the power parameter. Note that the Euclidean distance metric is selected by taking p as 2. Depending on the number of K selected, the first K nearest neighbors with the shortest RSS distance will contribute to the location prediction of the test sample. For a classifier, the concept of majority voting is applied whereby the floor with the highest vote among the K nearest neighbors will be chosen. As for a regressor, the mean coordinates of the K nearest neighbors will be predicted as the user location of the test sample. The WKNN algorithm applies a weight w_i to each K nearest neighbors as defined in (10) where N denotes the RSS of the training sample belonging to one of the K nearest neighbors. More weight would be assigned to nearer neighbors, and the larger the weight, the higher the influence of the neighboring point on the location prediction of the test sample.

$$w_i = \frac{1}{d(X, N)^2} \tag{10}$$

4 Simulations and Analysis

The performance of the proposed DECIPS is compared to those of the existing clustering-based IPSs reviewed in the Sect. 2.1 which adopted the GMM, SAP information and DNN based clustering techniques in terms of their floor prediction accuracy (the percentage of correctly predicted floors over the total number of floors predicted), average positioning error, maximum positioning error, 75th percentile for the positioning error, clustering testing time, localization testing time and also overall testing time. Moreover, the effects of the number of clusters are also investigated. WKNN is used for the indoor location prediction after DEC, SAP and DNN clustering is performed, while the random forest is used for the GMM based clustering technique instead.

Extensive computer simulations are performed using the FOE dataset [13]. The region of interest covers the 2nd and 3rd floor of the Faculty of Engineering (FOE) Wing B building of the Multimedia University Cyberjaya campus. The FOE dataset consists of 12637 instances in which 6300 instances are recorded for the 2nd floor while the remaining 6337 instances are recorded for the 3rd floor. The number of RPs defined for the 2nd floor and 3rd floor are 126 and 125, respectively. The 20 attributes of this dataset include the BLE fingerprint for 16 beacons, x-coordinate, y-coordinate, floor number, and the relative height of the reference point from the 2nd floor. The RSS intensity values are represented as negative values ranging from -100 dBm to -54 dBm.

Table 1 shows the results for the performance of different clustering-based techniques with a varying number of clusters. The number of clusters investigated are 2 and 4 clusters for all the clustering techniques except for SAP information-based clustering, which has 16 clusters since 16 beacons are deduced as the strongest beacons in the indoor environment. Note that the localization testing time represents the time needed for the trained localization algorithms to predict the indoor location while the clustering testing time represents the time required for the trained clustering algorithms to group the testing dataset into several clusters. On the other hand, the overall time represents the total time needed for clustering and localization to be carried out in the testing phase. For all techniques which involved WKNN, the results are shown for $K = 1$.

For the SAP-based clustering technique, the cluster formation, which should theoretically be consistent with the position distribution of the RPs lacks consistency since

Table 1. Performances of various clustering based indoor positioning techniques

Technique	Number of Clusters	Floor Accuracy (%)	Average Positioning Error (m)	Maximum Positioning Error (m)	75th Percentile of Positioning Error (m)	Localization Testing Time (s)	Clustering Testing Time (s)	Overall Testing Time (s)
SAP-WKNN	16	100	0.8890	26	1	0.0394	0.4543	0.4937
GMM-RDF	2	100	1.0208	12.8320	1.3350	422.0839	0.0130	422.0969
GMM-RDF	4	100	1.0077	13.4240	1.2776	455.4051	0.0186	455.4237
DNN-WKNN	2	100	0.7066	16	0	0.0504	0.2236	0.274
DNN-WKNN	4	100	0.7876	41.4367	1	0.0490	0.2811	0.3301
DECIPS	2	100	0.7068	16	0	0.0416	0.1232	0.1648
DECIPS	4	100	0.7175	15	1	0.0366	0.1356	0.1722

RPs in the same region (located adjacent to each other) could belong to different clusters. Apart from that, the clusters for quite a number of test points (TPs) are inaccurately predicted since they are assigned to a cluster which is different from that of the RPs sharing the same coordinates as them. In view of that, this clustering technique might not be feasible for the FOE testbed with many beacons detected since this results in a large number of clusters formed which complicates the clustering process besides defeating the purpose of reducing the computational complexity and time.

Moving on to the GMM based clustering technique with 2 clusters, similar situation whereby the TPs are assigned to a cluster which is different from that of the RPs sharing the same coordinates as them has occurred. Consequently, the location prediction of TPs grouped into a wrong cluster would now be based on the locations of RPs in that wrong cluster which is located further away from the ground truth locations of the TPs.

As the number of clusters increases to 4, the average positioning error generally increases. Note that the number of TPs with wrongly predicted clusters increases apparently with an increase in the number of clusters, thus producing less accurate results during the indoor location prediction.

In GMM based soft clustering technique, the number of RDF ensembles invoked per clustered observation is dependent on the number of clusters that the observation belongs to. This indirectly results in both higher computational complexity and longer time.

Also, from Table 1, it is observed that a larger number of clusters would result in a longer clustering time due to an increased computational complexity for all the clustering based techniques applied.

Figure 3 illustrates the percentage of performance gain of various techniques in terms of average positioning error benchmarked over GMM-RDF which acts as the baseline. Meanwhile, the bar charts for the percentage of GMM-RDF's localization and overall testing time are shown in Fig. 4. Note that for SAP-WKNN, both the blue and orange bars represent 16 clusters instead of 2 clusters and 4 clusters respectively due to the 16 strongest beacons detected for the indoor environment considered in the FOE dataset.

From Fig. 3, it is observed that the percentage of performance gain in terms of average positioning error over GMM-RDF obtained for all techniques regardless of 2 clusters or 4 clusters are positive. This implies that all techniques with both number of

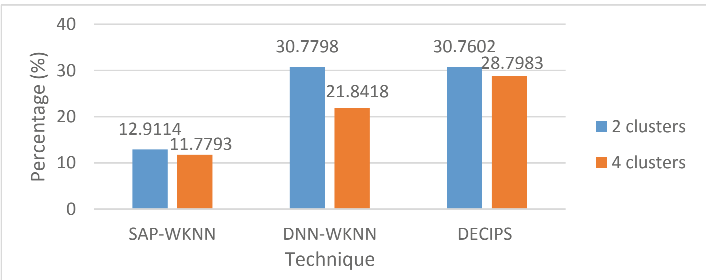
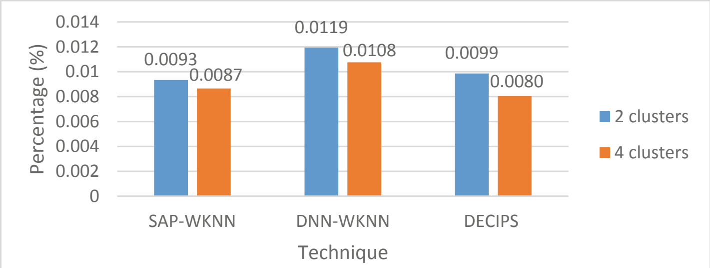
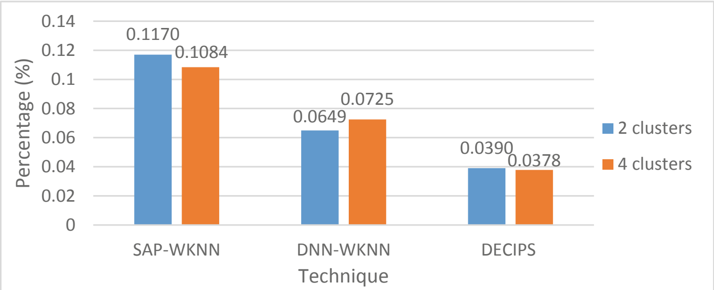


Fig. 3. Percentage of performance improvement in terms of average positioning error over GMM-RDF



(a)



(b)

Fig. 4. Percentage of GMM-RDF's time for (a) localization testing time and (b) overall testing time

clusters showed an improvement in average positioning error as compared to GMM-RDF. When the number of clusters is 2, the highest performance gain is achieved by the DNN-WKNN technique, while the proposed DECIPS has a slightly lower performance gain followed by the SAP-WKNN technique with the lowest performance gain. As the number of clusters increases to 4, the proposed DECIPS exhibits the highest performance gain, followed by the DNN-WKNN and SAP-WKNN techniques. It is also noteworthy that the percentage of performance gain decreases as the number of clusters increases from 2 to 4. This is because the average positioning error for the investigated techniques degrades with an increasing number of clusters.

From Fig. 4, it could be observed that the localization and overall testing time of the investigated techniques for both number of clusters are much shorter than those of the GMM-RDF technique. For both number of clusters, the localization and overall testing time of the proposed DECIPS are the shortest compared to SAP-WKNN and DNN-WKNN. As the number of cluster increases, the percentage of GMM-RDF's localization and overall testing time decreases for the investigated techniques due to the increase in localization and overall testing time of the GMM-RDF technique with the number of clusters.

To summarize, the proposed DECIPS yields the best performance among other clustering based IPS. When the number of clusters is 2, DECIPS produces an average positioning error that is 0.0283% higher than that of DNN-WKNN which is capable of producing the lowest average positioning error among all the techniques, but its overall testing time is 39.8540% lower than DNN-WKNN counterpart. Besides, DECIPS also attains a performance gain of 20.4949% and 30.7602% in terms of average positioning error along with 66.62% and 99.96% improvement in terms of overall testing time compared to SAP-WKNN and GMM-RDF, respectively. When the number of clusters is increased to 4, DECIPS produces the lowest average positioning error and also the shortest overall testing time among all the other clustering-based IPSs.

The problem of the "curse of dimensionality" often associated with large RSS BLE fingerprint datasets could be reduced with the DEC algorithm. The K-Means clustering performed in the feature space with a lower dimensionality enables only the essential features to be extracted without losing much information. This helps the K-Means clustering algorithm to better identify the hidden patterns in the embedded RSS data and can maximize the inter-cluster distance as far as possible, thus resulting in better cluster formation. Besides, clustering performed in lower dimensionality will also have a lower computational complexity, which is why the clustering testing time for DEC is shorter than that of the DNN which is another neural network approach.

5 Conclusion

A clustering-based IPS known as DECIPS is proposed to overcome the high computational complexity and long computational time caused by a large dataset. It is found that DECIPS is capable of outperforming the other clustering-based IPSs in terms of average positioning error and execution time. Although its average positioning error is 0.0283% slightly higher than the lowest average positioning error achievable by DNN-WKNN, nevertheless DEC still outperforms DNN in terms of the overall time by a significant 39.8540%. Besides, DECIPS also produced a performance gain of 20.4949% and 30.7602% in terms of average positioning error along with 66.62% and 99.96% improvement in terms of overall testing time compared to SAP-WKNN and GMM-RDF respectively, which makes it completely outperforms the SAP and GMM based clustering techniques in all aspects presented.

Acknowledgments. This research was funded by the Ministry of Higher Education Malaysia, grant number FRGS/1/2019/ICT02/MMU/03/13.

References

1. K. G. Tan, Z. A. Khaing, S. A. Moe, T. S. Min, A. Abdaziz, P. L. Chia, F. Hossain, P. T. Chih, H. Y. Wong, Review of Indoor Positioning: Radio Wave Technology, in: Applied Sciences, vol. 11, no. 1, 2020, pp. 1-44. DOI: https://doi.org/10.3390/app11010279
2. B. Ezhumalai, M. Song, K. Park, An Efficient Indoor Positioning Method Based on Wi-Fi RSS Fingerprint and Classification Algorithm, in: Sensors, vol. 21, no. 10, 2021, pp. 1-19. DOI: https://doi.org/10.3390/s21103418
3. J. Wang, J. Park, An Enhanced Indoor Positioning Algorithm Based on Fingerprint Using Fine-Grained CSI and RSSI Measurements of IEEE 802.11n WLAN, in: Sensors, vol. 21, no. 8, 2021, pp. 1-25. DOI: https://doi.org/10.3390/s21082769
4. A. Khalajmehrabadi, N. Gatsis, D. Akopian, Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges, in: IEEE Communications Surveys & Tutorials, vol. 19, 2017, pp. 1974–2002. DOI: https://doi.org/10.1109/COMST.2017.2671454

5. B. A. Akram, A. H. Akbar, O. Shafiq, HybLoc: Hybrid Indoor Wi-Fi Localization using Soft Clustering-based Random Decision Forest Ensembles, in: IEEE Access, vol. 6, 2018, pp. 38251–38272. DOI: https://doi.org/10.1109/ACCESS.2018.2852658
6. B. Altintas, T. Serif, Improving RSS-Based Indoor Positioning Algorithm via K-Means Clustering, in: 17th European Wireless 2011 - Sustainable Wireless Technologies, 2011, pp. 1-5.
7. H. Zhou, N. N. Van, Indoor Fingerprint Localization Based on Fuzzy C-Means Clustering, in: 2014 Sixth International Conference on Measuring Technology and Mechatronics Automation, 2014, pp. 337-340. DOI: https://doi.org/10.1109/ICMTMA.2014.83
8. P. Dai, Y. Yang, M. Wang, R. Yan, Combination of DNN and Improved KNN for Indoor Location Fingerprinting, in: Wireless Communications and Mobile Computing, vol. 2019, 2019, pp. 1-9. DOI: https://doi.org/10.1155/2019/4283857
9. E. Ozanich, A. Thode, P. Gerstoft, L. A. Freeman, S. Freeman, Deep Embedded Clustering of Coral Reef Bioacoustics, in: The Journal of the Acoustical Society of America, vol. 149, no. 4, 2021, pp. 2587–2601. DOI: https://doi.org/10.1121/10.0004221
10. J. Castela Forte, G. Yeshmagambetova, M. L. van der Grinten, B. Hiemstra, T. Kaufmann, R. J. Eck, F. Keus, A. H. Epema, M. A. Wiering, I. C. van der Horst, Identifying and Characterizing High-Risk Clusters in A Heterogeneous ICU Population With Deep Embedded Clustering, in: Scientific Reports, vol. 11, 2021, pp. 1-12. DOI: https://doi.org/10.1038/s41598-021-91297-x
11. Y. Li, A. Liu, T. Mi, et al., Striatal Subdivisions Estimated via Deep Embedded Clustering With Application to Parkinson's Disease, in: IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 9, 2021, pp. 3564–3575. DOI: https://doi.org/10.1109/JBHI.2021.3083879
12. J. Xie, R. Girshick, A. Farhadi, Unsupervised Deep Embedding for Clustering Analysis, in: Proceedings of the 33rd International Conference on Machine Learning, vol. 48, 2016, pp. 478–487.
13. I. S. Idi Amir, M. Z. I. Sapian, Y. H. Ng, C. K. Tan, Indoor Location Fingerprint Dataset for Multi-Floor Environments, in: IEEE Dataport, 2022. DOI: https://doi.org/10.21227/mabd-w688