Enhancing Indoor Localization Estimation Using RSS Similarity-Based *k*-Nearest Neighbors

Benyamain Yacoob
yacoobby@udmercy.edu
University of Detroit Mercy
Detroit, Michigan, USA

Daniel Marku
markuda@udmercy.edu
University of Detroit Mercy
Detroit, Michigan, USA

Mina Maleki
malekimi@udmercy.edu
University of Detroit Mercy
Detroit, Michigan, USA

Shadi Banitaan
banitash@udmercy.edu
University of Detroit Mercy
Detroit, Michigan, USA

ABSTRACT

The rapid growth of indoor positioning is revolutionizing our understanding of entity locations within indoor spaces. The fingerprint-based indoor localization method using Wi-Fi access points (APs) stands out for its minimal hardware requirements, making it one of the promising techniques in this domain. The k-nearest neighbors (k-NN) algorithm, a common machine learning (ML) approach, provides location estimations by pinpointing the k neighbors with the most similar representation values. However, conventional distance functions utilized in k-NN, including *Euclidean distance* and *cosine similarity*, prove insufficient in accurately identifying nearest neighbors based on the meaningful interpretation of received signals from APs. Thus, in this research, we propose a new distance function based on received signal strength (RSS) similarity that can be employed in tandem with k-NN to find the optimal nearest neighbors for real-time localization on a more consistent basis when compared to other distance functions. The experimental results of the collected dataset demonstrated a 1 to 3% improvement in the coefficient of determination (R^2) score and a reduction in distance error by 6.5 to 10 inches, as determined from the mean absolute error (MAE).

CCS CONCEPTS

• **Computing methodologies → Machine learning approaches**.

KEYWORDS

indoor localization, machine learning, received signal strength, rss similarity, fingerprinting

ACM Reference Format:
Benyamain Yacoob, Daniel Marku, Mina Maleki, and Shadi Banitaan. 2024. Enhancing Indoor Localization Estimation Using RSS Similarity-Based *k*-Nearest Neighbors. In *2024 6th Asia Conference on Machine Learning and Computing (ACMLC 2024), July 26–28, 2024, Bangkok, Thailand*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3690771.3690792

ACMLC 2024, July 26–28, 2024, Bangkok, Thailand

ACM ISBN 979-8-4007-1001-8/24/07
https://doi.org/10.1145/3690771.3690792

1 INTRODUCTION

Indoor positioning has emerged as a critical technology in recent years, playing a pivotal role in enabling a wide range of applications that depend on accurate location information within buildings and other indoor environments. From enhancing user navigation and wayfinding in complex structures like airports and shopping malls to optimizing logistics and asset management in industrial settings such as warehouses and factories, the capability to pinpoint locations indoors has revolutionized operational efficiency and user experience across various domains. Unlike outdoor positioning, which utilizes Global Positioning System (GPS) satellites, indoor positioning faces limitations because GPS signals are often weak or unavailable indoors due to obstruction by walls, ceilings, and other physical structures. As a result, various indoor positioning methods have been developed, leveraging different technologies and techniques to determine the precise location of users or devices within indoor spaces [10].

One of the most promising approaches to indoor positioning is fingerprint-based methods, which rely on the unique characteristics of wireless signals, such as Wi-Fi, Bluetooth, or radio-frequency identification (RFID), to create a "fingerprint" of a specific location. These methods typically involve a two-phase process: an offline training phase, where signal measurements are collected at known locations to create a fingerprint database, and an "online phase", where real-time signal measurements are compared against the database to estimate the user's location [1, 13, 17].

Wi-Fi-based fingerprinting has become a widely adopted and effective approach for indoor positioning, leveraging the ubiquitous presence of Wi-Fi access points (APs) in indoor environments. By capturing the Received Signal Strength (RSS) of Wi-Fi signals from multiple APs, a unique fingerprint can be generated for each location within a building. This fingerprint serves as a reference dataset, enabling the estimation of a user's device location based on the RSS measurements received from nearby APs. Leveraging the connectivity of networked devices like smartphones and computers, RSS levels are collected from nearby APs to create these fingerprints. RSS fingerprinting employs machine learning (ML) algorithms trained on these RSS measurements during an offline phase, where a database is populated with recorded RSS levels. This training data enables accurate positioning of network devices within a 2-D Cartesian coordinate plane, effectively utilizing the spatial distribution of Wi-Fi signals for indoor localization.

In recent years, ML approaches have been increasingly applied to improve the accuracy and robustness of indoor positioning models. Various ML algorithms, such as k-nearest neighbors (k-NN)[4, 5, 19], random forest (RF)[9, 15], support vector machine (SVM)[8], deep learning (DL)[3, 6, 11, 21, 24], autoencoder[6, 11, 20, 24], and gradient boost (GB)[6, 16, 23], have been explored to enhance the performance of indoor positioning models.

The k-NN algorithm is one of the primary and powerful methods for capturing indoor positioning at a high level of accuracy. The algorithm calculates the distances between the user whose location needs to be estimated and previously collected data points representing user locations. Subsequently, it identifies the k points with the most similar representation values (the nearest neighbors) and computes the average of their corresponding locations to arrive at an estimated location. However, the most commonly used distance functions in k-NN like Euclidean distance or cosine similarity may not fully capture the complexities of indoor environments, where signal strengths can vary significantly due to factors such as walls, obstacles, and interference[10].

In this paper, we propose a customized distance function based on RSS similarity that can be used in conjunction with the k-NN algorithm. This distance function considers the complexities of the signals in the indoor environment, making it a more adaptable method to find the closest neighbors. RSS similarity finds the optimal training instances by dynamically adapting to the user's latent representations.

The discussion in this paper will follow in the order listed as follows. Section II reviews the ML approach using fingerprinting methods in related work. In Section III, we cover the fingerprinting process and the implementation of the RSS localization estimation algorithms and our proposed distance function. Section IV compares and analyzes results from each distance/similarity algorithm. Section V concludes with remarks on our findings.

2 RELATED WORK

Many of the problems for indoor localization using Wi-Fi APs stem from how random AP signals can be. Therefore, much of the literature on indoor localization uses fingerprinting-based methods to address this issue. Existing research contends with GPS software since 68% of smartphone users have location-based services enabled on their apps[25]. However, location-based services can receive weak GPS signals in indoor environments, which produces ambiguous tracking and navigation. Different approaches to tracking indoor occupants become preferable to GPS when localizing at the building scale[10]. Salamah et al.[1], Khullar et al.[13], and Roy et al.[17] operate on an approach that can be carried out offline. Xu et al.[7] consider a signal ratio to create the Apollonius circles, eliminating the transmission power distance parameter, as well as the unknown attenuation factor. The technique is synchronous to creating virtual APs and providing respectable estimations, differing from previous methods as they disregard the absolute physical distance to locate each AP. Shu et al.[23] also address uneven fingerprint density and device heterogeneity and propose gradient fingerprinting (GIFT) that extracts binary RSS gradients from their fingerprinting map and establishes a gradient database to minimize any influence of how Wi-Fi signals are transmitted when looking at a certain device. Gufran et al.[6] propose stacked autoencoder neural networks with gradient boosting for indoor localization (SANGRIA) that also account for device heterogeneity. However, Njima et al.[21] provide a top-level alternative, advocating for the results of indoor localization using weighted semi-supervised methods with a supervised model that trains on a small amount of labeled data as a way to predict emergent, pseudo-labels. Yoo et al.[22] also adopt the idea of pseudo-labeling to compensate for the lack of labeled data, but add a temporal relation to the unlabeled training data that are collected as time series. Jia et al.[3] also utilized the neural network framework approach to address the topic of indoor localization and found that a deep neural network (DNN) can be fed the fingerprint database to determine the structure of the DNN and the weight of each of its neurons. Yan et al.[14] account for the complexity of localizing estimations within indoor environments, so their method requires location matching from a recurrent neural network (RNN) that converts the continued signals into the continued path to form the mapping relation from signal to location. Liu et al.[11] also consider the unstable accuracy of RSS-based localization and suggest denoising the noisy RSS to estimate the correct RSS for each AP. Similar to our experimental conditions, Akram et al.[2] target indoor localization at the building level by splitting their dataset using soft clustering, made possible with the Gaussian mixture model (GMM) by taking into consideration overlapping and non-overlapping data subsets. Arthi et al.[18] share their struggle to find an evaluation metric that would be ideal for representing their results. That is primarily the reason the metric they used was the minimum mean squared error (MMSE). From their analysis, they found that using the k-NN algorithm coupled with this metric gave them more accurate data and their statistics were far more interpretable than what was used before. Bi et al.[12] experiment with evaluation metrics as well, using MAE, root mean square error (RMSE), 50$^{\text{th}}$ percentile error, 75$^{\text{th}}$ percentile error, 95$^{\text{th}}$ percentile error. Others, like Khatab et al.[24], suggest increasing the number of fingerprints to output an improved localized performance. They saw this by gradually comparing the metrics, with each iteration increasing the number of training fingerprints. Singh et al.[16] deviate from other frameworks and introduce relational labeling (RL) combined with an XGBoost-based ML method. Finding the soundest method that delivers good results requires trials of experimentation, such as when Kia et al.[8] experimented with many different ML algorithms like lasso regression (LR), ridge regression (RR), support vector machine regression (SVR), etc. to extract the best method. Hou and Wang[9] introduced the RF-KELM indoor positioning algorithm, which combines random forest (RF) for feature importance evaluation and kernel extreme learning machine (KELM) for positioning. The RF is used for AP selection, enhancing the algorithm's robustness to signal changes and improving computational efficiency, while KELM is employed for fast and accurate position prediction. Narasimman and Alphones[15] addressed the issue of feature extraction before classification or regression, introducing a framework called DumbLoc. This framework utilizes RSS values from the strongest AP signals and normalized output labels to achieve high positioning accuracy without optimization. Some studies explored k-NN because it is a straightforward yet powerful ML algorithm that can be used to estimate a user's location based on the characteristics of nearby fingerprints. They have also explored custom distance functions in conjunction with the k-NN algorithm for indoor positioning. Zhou et al.[19] proposed the Q weighted k-NN algorithm (Q-WKNN), which utilizes base Q to transform RSS to Q-based RSS, enhancing positioning accuracy and real-time performance. Wang et al.[5] introduced a novel WKNN based on signal similarity and spatial position, analyzing the relationship between RSS similarity and position distance to improve the positioning accuracy of the WKNN algorithm. Another study by Wang et al.[4] proposed an improved Wi-Fi positioning method based on fingerprint clustering and signal-weighted Euclidean distance (SWED). Their experiments, conducted in two experimental fields, indicated that the proposed position label-assisted (PL-assisted) clustering result can reflect the position distribution of reference points (RPs) and the proposed SWED-based WKNN (SWED-WKNN) algorithm can significantly improve the positioning accuracy compared to traditional methods. These studies highlight the potential of custom distance functions in combination with the k-NN algorithm to enhance indoor positioning performance. Because of this, we propose a new customized distance function that works with k-NN to select the optimal neighbors considering the RSS similarity.

3 INDOOR LOCALIZATION WITH RSS-BASED SIMILARITY METHODOLOGY

There are two phases across the indoor localization framework as shown in Figure 1. The initial step of the framework is termed the "offline phase" which constitutes the fingerprinting method. Fingerprinting accrues data such as AP positions, identities of said APs, and RSS levels to these APs from various fingerprint locations into databases. This data is then preprocessed to train an ML model. During the "online phase", live Wi-Fi scans are taken, collecting RSS-based fingerprints as test data for our k-NN model to output an estimated user location.

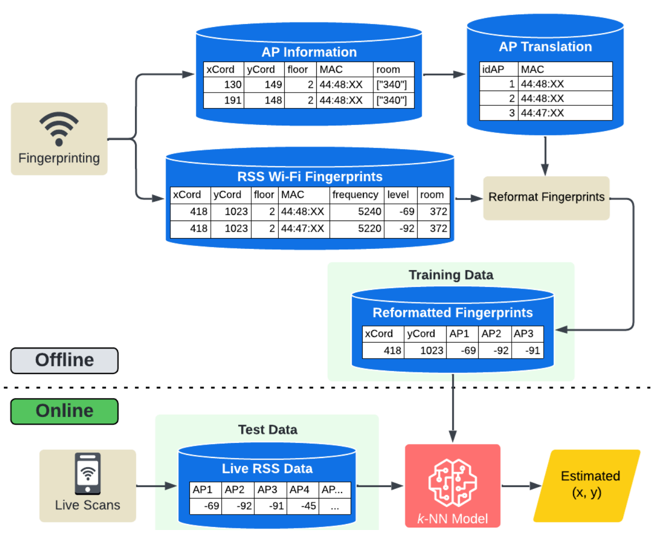

Figure 1: Indoor localization framework

3.1 Offline Phase

The target of the "offline phase" is to collect fingerprints and AP data to initialize the model. This is conducted via an internal application on a smartphone device. The provided data involves all available APs, their MAC addresses, and locations of these APs within our area of interest; all of which are then filtered and stored in a database as shown in Table 1.

Fingerprints collect RSS values emitted by all reachable AP Wi-Fi signals from the tagged locations. Once an adequate number of fingerprints are amassed, detected APs are localized using their RSS signals in conjunction with the (x, y) coordinates of the fingerprints placed by the user. The data retrieved is then stored for later use in bringing the model online and outputting an estimation on the map representative of the user's current location.

Table 1: AP information

xCoord	yCoord	MAC	roomNum
2806	1494	XX:XX:XX:XX:XX:XX	441
2352	1455	XX:XX:XX:XX:XX:XX	441
1959	632	XX:XX:XX:XX:XX:XX	437
2813	873	XX:XX:XX:XX:XX:XX	437

The fingerprints in this dataset were taken in areas with high foot traffic and within the radius of the APs. This guarantees accurnate localization in key areas via strong RSS values. In this research, RSS data is measured between -100 (weakest) and -30 (strongest). To combat RSS noise, three consecutive RSS scans were taken at each fingerprint. The strongest RSS values obtained from each AP across all scans were stored. Some APs, however, are not reached from certain fingerprint locations. Wang et al.[20] account for missing RSS values in their data collection process by filling in those values to normalize the raw RSS data. We compensate for this by declaring every signal to an AP to be -100 unless some RSS is detected. The fingerprints are then reformatted, which requires that each AP found is designated a unique identifier digit (UID) to reference an RSS value for consistency across all fingerprint records. The table of *Reformatted Fingerprint* records can be seen in Table 2, which indicates all RSS levels for each fingerprint to every APs assigned digit. This dataset of fingerprints is then used as potential neighbors to localize users within the k-NN algorithm.

Table 2: Fingerprint RSS's mapped to identified APs (Reformatted Fingerprints)

xCoord	yCoord	floor	AP1	AP2	AP3	AP#
413	94	2	-81	-80	-39	...
531	93	2	-100	-83	-41	...
642	92	2	-85	-86	-46	...

3.2 Online Phase

The user location is estimated in this phase where live Wi-Fi scans are taken to collect the RSS-based fingerprints as test data. These scans extract the relevant data and are fed into the ML model, to output a coordinate pair (x, y) prediction. Using the fingerprints collected in the "offline phase" and the live RSS signals received in the "online phase", a k-NN regressor algorithm was utilized to estimate the user location within a designated indoor environment.

The k-NN algorithm uses a distance function to find the distance of each fingerprint to the test data. These distances are sorted and best k neighbors are selected. The estimated (x, y) coordinate pair of the user's location would be calculated as the weighted average of the coordinates of these k RPs, where the weights are proportional to the distance. Distance functions used in this study are described below:

3.2.1 Euclidean Distance Function. This is a widely used metric for measuring the L2-norm of the difference between two representations in a multi-dimensional space. In indoor localization using the k-NN algorithm, Euclidean distance is utilized to determine the proximity of neighboring points based on their RSS values. The Euclidean distance between two representations is calculated by taking the square root of the sum of the squared differences between corresponding RSS values from each Wi-Fi AP. Given the user's RSS vector: $u = (u_1, u_2, \dots, u_n)$, where u_i represents the RSS value from the i-th AP at the user's current location, and the RSS vector of a reference point j: $r_j = (r_{j1}, r_{j2}, \dots, r_{jn})$, where r_{ji} represents the RSS value from the i-th AP at the reference point j, the Euclidean distance between the user's RSS vector and the RSS vector of a reference point j can be calculated as:

$$\begin{aligned} d(u, r_j) &= \sqrt{\sum_{i=1}^{n} (u_i - r_{ji})^2} \\ &= \sqrt{(u_1 - r_{j1})^2 + (u_2 - r_{j2})^2 + \cdots + (u_n - r_{jn})^2} \end{aligned} \tag{1}$$

3.2.2 Cosine Similarity Distance Function. Cosine similarity is another popular metric used to measure the similarity between two representations, particularly in high-dimensional spaces. Using the k-NN algorithm, cosine similarity is applied to assess the proximity of neighboring points based on the angle between their RSS vectors. The cosine similarity between two representations is calculated by taking the dot product of their RSS vectors and dividing it by the product of the magnitudes of the vectors, as shown in Equation 2.

$$cos(\theta_j) = \frac{u \cdot r_j}{||u|| \cdot ||r_j||} = \frac{\sum_{i=1}^{n} u_i \cdot r_{ji}}{\sqrt{\sum_{i=1}^{n} u_i^2} \cdot \sqrt{\sum_{i=1}^{n} r_{ji}^2}} \tag{2}$$

3.2.3 Proposed RSS Similarity Distance Function. Commonly used distance functions in k-NN, such as Euclidean distance, become less effective in capturing the intricate relationships between Access Points (APs). This limitation is particularly pronounced in indoor environments where AP signal strengths can vary significantly based on the user's location. For instance, the smallest Euclidean distance between a test sample and a fingerprint in the training set does not always indicate the most accurate match. In certain scenarios, a larger Euclidean distance for a specific fingerprint can actually better correlate with the test sample's RSS values, underscoring the nuanced variability in signal strengths and emphasizing the necessity for a more refined similarity metric like RSS similarity.

RSS similarity is crucial for determining the proximity of neighboring points and plays an important role in the k-NN algorithm. As illustrated in Algorithm 1, RSS similarity distance function is designed to calculate a weighted similarity between two input vectors. The weights are dynamically assigned based on the percentiles of representation values in the input arrays. The weighted similarity calculation involves taking the absolute difference between corresponding representation values and summing the product of these differences and the assigned weights. Each element (or column) in the input vector represents the RSS value from a specific Wi-Fi AP. This approach guarantees that higher weights are assigned to values in the arrays with higher percentiles, emphasizing the impact of taking the highest values on the similarity calculation. We only consider RSS signals above the 95^{th} percentile. RSS similarity considers all APs and their relationship to each other, making it a more adaptable method. It also considers the training data for each test input of the user moving, finding the best training fingerprints that are closest.

The pseudocode of our approach can be seen in Algorithm 1.

4 RESULTS AND DISCUSSION

In this section, we discuss our findings regarding distance function performance determined by various evaluation metrics.

4.1 Dataset

The dataset comprises Wi-Fi scans (collected fingerprints) from the engineering building at the University of Detroit Mercy. To ensure the representativeness of our findings, we collected fingerprint data across three respective floors within this building. There were 77 fingerprints conducted on the first floor, 105 on the second, and 72 on the third. Each floor exhibited varying classroom sizes, which allowed us to capture a diverse range of environmental conditions. Additionally, the fingerprints were strategically distributed throughout the entire floor area, ensuring comprehensive coverage and minimizing potential biases or blindspots. By having a dataset that encompasses such diverse settings, from small to large classrooms and spanning multiple floors, we aimed to create a thorough representation of the real-world scenarios to generalize our proposed approach.

4.2 Evaluation Metrics

To evaluate the performance of the applied model for the estimation of the location of the user, we calculate both the mean absolute error (MAE) and the coefficient of determination (R^2), comparing the actual location of the user with the estimated location.

The MAE is calculated as the average of the absolute differences between the predicted and actual coordinates:

$$MAE = \frac{1}{n} \sum_{i=1}^{n} \sqrt{\left(x_{i_{\text{est.}}} - x_{i_{\text{act.}}}\right)^2 + \left(y_{i_{\text{est.}}} - y_{i_{\text{act.}}}\right)^2} \tag{3}$$

where $(x_{i_{actual}}, y_{i_{actual}})$ and $(x_{i_{estimated}}, y_{i_{estimated}})$ indicate the coordinates of the actual and the corresponding estimated points for the i-th sample, and n is the total number of samples.

Algorithm 1 RSS Similarity

Require: Two arrays of RSS values from Wi-Fi APs: u and r_j
Ensure: Weighted similarity score between u and r_j

Calculate percentile: $per = \text{percentiles}((u \cup r_j), [96, 98, 99.5, 100])$
Initialize empty array: $rss\_weights = \text{zeros}(dimensions(u))$
for $i \in 0 \dots dimensions(u) - 1$ **do**
  if $per_2 \leq u_i \leq per_3$ or $per_2 \leq r_{ji} \leq per_3$ **then**
    $rss\_weights_i = 1$
  else if $per_1 \leq u_i < per_2$ or $per_1 \leq r_{ji} < per_2$ **then**
    $rss\_weights_i = 0.75$
  else if $per_0 \leq u_i < per_1$ or $per_0 \leq r_{ji} < per_1$ **then**
    $rss\_weights_i = 0.30$
  else
    $rss\_weights_i = 0.15$
  end if
end for
Calculate absolute difference: $rss\_similarity = |u - r_j|$
Calculate weighted similarity:
$weighted\_similarity = \sum_{i=0}^{dim(u)} (rss\_similarity \cdot rss\_weights)$
return $weighted\_similarity$

The R^2 score ranges from 0 to 1, with 1 indicating a perfect fit and 0 indicating a model that performs no better than predicting the mean value of the target variable(s). It is calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} \left[(x_{i_{\text{est.}}} - x_{i_{\text{act.}}})^2 + (y_{i_{\text{est.}}} - y_{i_{\text{act.}}})^2 \right]}{\sum_{i=1}^{n} \left[(x_{i_{\text{act.}}} - \bar{x})^2 + (y_{i_{\text{act.}}} - \bar{y})^2 \right]} \tag{4}$$

where $(\bar{x}, \bar{y})$ is the mean of the actual coordinate pairs. These error metrics help assess model accuracy in estimating the user's location based on the RSS vectors. A lower MAE indicates that the predicted coordinates are closer to the actual coordinates on average and a higher R^2 score suggests that the model explains a larger proportion of the variance in the target variable (i.e., the user's location).

4.3 Distance Function Assessment

The performance of each k-NN distance function was examined across multiple floors to produce definitive results. We chose a fixed number of neighbors ($k = 5$) and employed weighted distances to emphasize the importance of the most similar data points. Cosine and Euclidean distance functions were implemented and compared with the proposed RSS similarity distance function to show the dynamic nature of our function. The results are shown in Table 3.

For all three floors, RSS similarity consistently outperformed cosine similarity and Euclidean distance in terms of both R^2 values and MAE. This indicates that the custom distance metric, designed specifically for assessing similarity between data points using the k-NN algorithm, is better suited for the given dataset and regression task.

Table 3: Comparison of k-NN regressor using different distance functions for different floors

Floor	*F1*		*F2*		*F3*	
Distance Function	$\mathbf{R^2}$	**MAE**	$\mathbf{R^2}$	**MAE**	$\mathbf{R^2}$	**MAE**
RSS Similarity	0.94	64	0.98	55	0.86	53
Cosine	0.93	73	0.97	61	0.85	54
Euclidean	0.93	75	0.97	61	0.84	57

4.4 Nearest Neighbors Visual Inspection

Additional investigation was conducted to provide insight into what neighbors were being selected by distance function. Neighbors chosen are an important element of the model since they are weighted in numerical order. The neighbors chosen by RSS similarity and Euclidean distance were scrutinized closely as those two distance functions generated the most consistent results across evaluation metrics. The visual sample of the custom RSS similarity approach against Euclidean distance for k-NN with $k = 5$ is shown in Figure 2. The figure shows that the nearest neighbors found by the custom RSS similarity function are marginally more accurate than the Euclidean distance function; $NN - 4$ in RSS similarity lies much closer than $NN - 4$ in the Euclidean distance to the actual user location. This tends to have a more accurate user location estimation by comparing the distance of the estimated user location to the actual user location.

5 CONCLUSION

In this study, we introduced a novel RSS similarity distance function designed to enhance indoor localization accuracy by finding more meaningful nearest neighbors in the k-NN algorithm. The RSS similarity algorithm's dynamic nature enhances its ability to capture meaningful patterns by using dynamic weighted similarity based on representation value percentiles. We compared our proposed method with two commonly used and powerful distance functions, Euclidean distance and cosine similarity.

Using Wi-Fi scans collected from the engineering building at the University of Detroit Mercy, we validated our model's reliability and performance improvements, achieving a 1 to 3% enhancement in R2 score and reducing distance errors by 6.5 to 10 inches as measured by MAE score using the RSS similarity distance function to find the nearest neighbors in the k-Nearest Neighbors (k-NN) algorithm. This experimental result illustrates the effectiveness of our approach compared to traditional methods such as cosine similarity and Euclidean distance, highlighting its potential for enhancing indoor localization accuracy in practical applications

Future work includes the implementation of strategies to mitigate RSS noise impact, exploration of alternative customized distance function algorithms, and integration of diverse datasets to improve overall generalizability and applicability in practical indoor positioning applications.

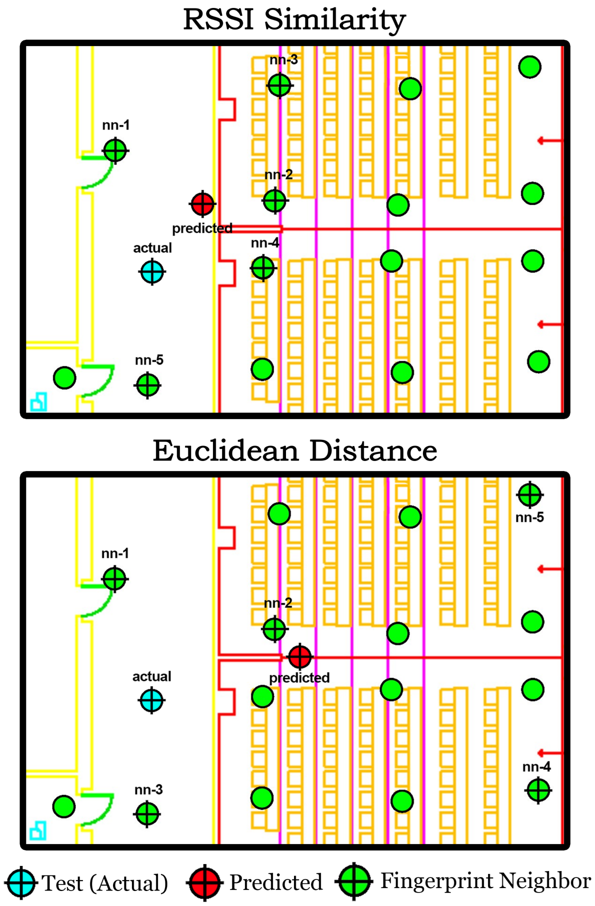

Figure 2: Sample user location prediction by $k = 5$ neighbors found by RSS similarity v. Euclidean distance (Floor #2)

ACKNOWLEDGMENTS

This work was supported by the DTE Energy/Engineering Society of Detroit E-Challenge grant. We would like to thank Nassif Rayess, Alan Hoback, Scott Vangilder, and Alexa Rihana-Abdallah for their valuable comments.

REFERENCES

[1] Maha A. Sharkas Ahmed H. Salamah, Mohamed Tamazin and Mohamed Khedr. 2016. An enhanced WiFi indoor localization system based on machine learning. In *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. 1–6. https://doi.org/10.1109/IPIN.2016.7743586

[2] Ali H. Akbar Beenish A. Akram and Omair Shafiq. 2018. HybLoc: Hybrid indoor wi-fi localization using soft clustering-based random decision forest ensembles. *IEEE Access* 6 (2018), 28526–28538. https://doi.org/10.1109/ACCESS.2018.2852658

[3] Baoqi Huang Bing Jia, Zhaopeng Zong and Thar Baker. 2020. A DNN-based WiFi-RSS indoor localization method in IoT. In *Lecture notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering*, Vol. 323. 157–166. https://doi.org/10.1007/978-3-030-67720-6_14

[4] Baoguo Yu Ruicai Jia Boyuan Wang, Xuelin Liu and Xingli Gan. 2019. An Improved WiFi Positioning Method Based on Fingerprint Clustering and Signal Weighted Euclidean Distance. *Sensors* 19, 10 (2019), 2300. https://doi.org/10.3390/s19102300

[5] Xuelin Liu Baoguo Yu Ruicai Jia Lu Huang Boyuan Wang, Xingli Gan and Haonan Jia. 2020. A Novel Weighted KNN Algorithm Based on RSS Similarity and Position Distance for WiFi Fingerprint Positioning. *IEEE Access* 8 (2020), 30591–30602. https://doi.org/10.1109/ACCESS.2020.2973212

[6] Saideep Tiku Danish Gufran and Sudeep Pasricha. 2023. SANGRIA: Stacked Autoencoder Neural Networks with Gradient Boosting for Indoor Localization. *IEEE Embedded Systems Letters* 4, 1 (2023), 27–30. https://doi.org/10.1109/LES.2023.3279017

[7] Shuaiwei Luo Fan Xu, Xuke Hu and Jianga Shang. 2020. An efficient indoor wi-fi positioning method using virtual location of AP. *ISPRS international journal of geo-information* 9, 4 (2020), 261. https://doi.org/10.3390/IJGI9040261

[8] Laura Ruotsalainen Ghazaleh Kia and Jukka Talvitie. 2022. Toward Accurate Indoor Positioning: An RSS-Based Fusion of UWB and Machine-Learning-Enhanced WiFi. *Sensors* 22, 9 (2022), 3204. https://doi.org/10.3390/s22093204

[9] Bingnan Hou1 and Yanchun Wang. 2024. RF-KELM indoor positioning algorithm based on WiFi RSS fingerprint. *Measurement Science and Technology* 35, 4 (2024). https://doi.org/10.1088/1361-6501/ad1873

[10] Somaya Al-Maadeed Jayakanth Kunhoth, AbdelGhani Karkar and Abdulla Al-Ali. 2020. Indoor positioning and wayfinding systems: a survey. *Hum. Cent. Comput. Inf. Sci.* 10, 1 (2020), 18. https://doi.org/10.1186/s13673-020-00222-0

[11] Zhiwen Pan Jing Liu, Nan Liu and Xiaohu You. 2018. AutLoc: Deep autoencoder for indoor localization with RSS fingerprinting. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*. 1–6. https://doi.org/10.1109/WCSP.2018.8555665

[12] Baoguo Yu-Hongji Cao Tongguang Shi Jingxue Bi, Yunjia Wang and Lu Huang. 2022. Supplementary open dataset for WiFi indoor localization based on received signal strength. *Satell Navig* 3 (2022), 25. https://doi.org/10.1186/s43020-022-00086-y

[13] Rajan Khullar and Ziqian Dong. 2017. Indoor localization framework with WiFi fingerprinting. In *2017 26th Wireless and Optical Communication Conference (WOCC)*. 1–5. https://doi.org/10.1109/WOCC.2017.7928970

[14] Siqi Bai Mingjiang Yan, Fengu Xu and Qun Wan. 2018. A noise reduction fingerprint feature for indoor localization. In *2018 15th International Conference on Wireless Communications and Signal Processing (WCSP)*. 1–6. https://doi.org/10.1109/WCSP.2018.8555638

[15] Srivathsan Chakaravarthi Narasimman and Arokiaswami Alphones. 2024. DumbLoc: Dumb indoor localization framework using wi-fi fingerprinting. *IEEE Sensors Journal* 24, 9 (2024). https://doi.org/10.1109/JSEN.2024.3374415

[16] Rajiv Punmiya Navneet Singh, Sangho Choe and Navneesh Kaur. 2022. XGBLoc: XGBoost-Based indoor localization in multi-building multi-floor environments. (2022). https://doi.org/10.3390/S22176629

[17] Dip Ghosh Priya Roy, Chandreyee Chowdhury and Sanghamitra Bandyopadhyay. 2019. JUIndoorLoc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity. *Wireless Personal Communications* 104, 1 (2019), 335–355. https://doi.org/10.1007/S11277-019-06188-2

[18] Abhiviraj Pillai Yash Nair R. Arthi, Digvijay Singh Rawat and S. S. Kausik. 2020. Analysis of indoor localization algorithm for WiFi using received signal strength. In *Lecture notes in electrical engineering*, Vol. 556. 313–324. https://doi.org/10.1007/978-981-15-7504-4_40

[19] Yexi Yang Rong Zhou and Puchun Chen. 2021. An RSS Transform—Based WKNN for Indoor Positioning. *Sensors* 21, 17 (2021), 5685. https://doi.org/10.3390/s21175685

[20] Haiyong Luo Fang Zhao Wenhua Shao Rongrong Wang, Zhaohui Li and Qu Wang. 2019. A robust wi-fi fingerprint positioning algorithm using stacked denoising autoencoder and multi-layer perceptron. *Remote Sensing* 11, 11 (2019), 1293. https://doi.org/10.3390/RS11111293

[21] Ahmad Bazzi Wafa Njima and Marwa Chafii. 2022. DNN-based indoor localization under limited dataset using GANs and semi-supervised learning. *IEEE Access* 10 (2022), 69896–69909. https://doi.org/10.1109/ACCESS.2022.3187837

[22] Jaehyun Yoo and Jongho Park. 2019. Indoor localization based on wi-fi received signal strength indicators: Feature extraction, mobile fingerprinting, and trajectory learning. *Applied Sciences* 9, 18 (2019), 3930. https://doi.org/10.3390/APP9183930

[23] Jiaqi Zhang Philippe Coué Peng Cheng Jiming Chen Kang G. Shin Yuanchao Shu, Yinghua Huang. 2016. Gradient-based fingerprinting for indoor localization and tracking. *IEEE Transactions on Industrial Electronics* 63, 9 (2016), 5880–5890. https://doi.org/10.1109/TIE.2015.2509917

[24] Amirhosein Hajihoseini Zahra Ezzati Khatab and Seyed Ali Ghorashi. 2018. A fingerprint method for indoor localization using autoencoder based deep extreme learning machine. *IEEE Sensors Letters* 2 (2018), 1–4. https://doi.org/10.1109/LSENS.2017.2787651

[25] ZipDo. [n. d.]. Essential smartphone usage statistics in 2024. https://zipdo.co/statistics/smartphone-usage/.