International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. #, No.# (Mon-20..)

Evaluating Navigation Efficiency: A Comparative Study of Search Performance in Indoor Positioning Systems

Jevon Sebastian[1], Justin Orlean[2], Gede Putra Kusuma [3]

[1,2,3] *Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480*

E-mail address: jevon.sebastian@binus.ac.id justin.orlean@binus.ac.id, inegara@binus.edu

Received ## Mon. 20##, Revised ## Mon. 20##, Accepted ## Mon. 20##, Published ## Mon. 20##

Abstract: Indoor positioning systems are becoming more popular due to the limitations of Global Positioning System (GPS) in locating a person's location in a closed environment. One of the most frequently used method for indoor positioning systems is fingerprinting. Fingerprinting relies on obtaining a set of Received Signal Strength Indicator (RSSI) values from multiple access points and then comparing it to the database to predict a device location using methods such as k-Nearest Neighbors (KNN). However, this process takes a long time and does not scale well with large and high dimensional dataset. In this study, we evaluate and compare three different methods for search optimization which are Hierarchical Navigable Small World (HNSW), Locality-Sensitive Hashing (LSH) and Inverted File Index (IVF). The evaluation will be conducted based on the search speed and the number of correct predictions of each searching method. Our results show that HNSW outperforms the other methods by a slight margin in terms of accuracy, giving an accuracy of 74.1% compared to LSH accuracy of 72.5% and IVF accuracy of 73.2%. In terms of search speed, HNSW and IVF are significantly faster than LSH, with average time per query of 0.014 and 0.016 seconds respectively, compared to LSH time of 4.813 seconds per query.

Keywords: ***Indoor Positioning Systems***, ***Similarity Search***, ***Hierarchical Navigable Small World, Inverted File Index***, ***Locality-Sensitive Hashing, Nearest neighbor search***

1. INTRODUCTION

Recently, indoor positioning systems has been used very frequently due to the limitations of GPS for indoor locations. GPS cannot accurately locate a user's position indoors since there is No Line-of-Sight (NLoS), the signal transmitted by a GPS satellite to the user's device is blocked by various obstacles (e.g. walls, roofs, doors, etc.) [1]. On the other hand, indoor positioning systems can better locate a user's position indoors by using various tools which do not require external dependencies, unlike GPS which depends on a satellite signal transmission. Tools that can be used to implement indoor positioning systems include but are not limited to Wi-Fi, BLE beacons, RFID anchors, ultra-wideband anchors, and other various signal emitting devices [2].

The use of indoor positioning systems has played a crucial role in various scenarios. One of its applications is to enhance navigation in multi-floor buildings, such as shopping malls [3], hospitals [4], or museums [5]. Indoor positioning systems facilitates efficient and effective navigation, particularly assisting first-time visitors, the elderly, or emergency responders in unfamiliar buildings. Moreover, indoor positioning systems have also proven to be very useful in the industrial sector. It contributes significantly to enhancing manufacturing processes by enabling robots and automated guided vehicles to achieve precise and real-time positioning [6].

In indoor positioning system, one of the most popular method to estimate a user location is fingerprinting [7], [8]. Fingerprinting consists of offline phase and online phase. The offline phase is done by collecting data from reference points (RP) using a mobile device to represent a location, which is called a fingerprint. RPs are usually chosen by

E-mail:author's email

dividing an area into multiple smaller sub-areas so that the whole area is covered when taking fingerprint samples. A fingerprint data usually consists of a set of RSSI values obtained from multiple signal emitting devices called access points and the real world coordinate of the location such as latitude, longitude, and altitude. The fingerprints are then stored in the database and are later used to predict a device position. This process visualization can be seen in figure 1.

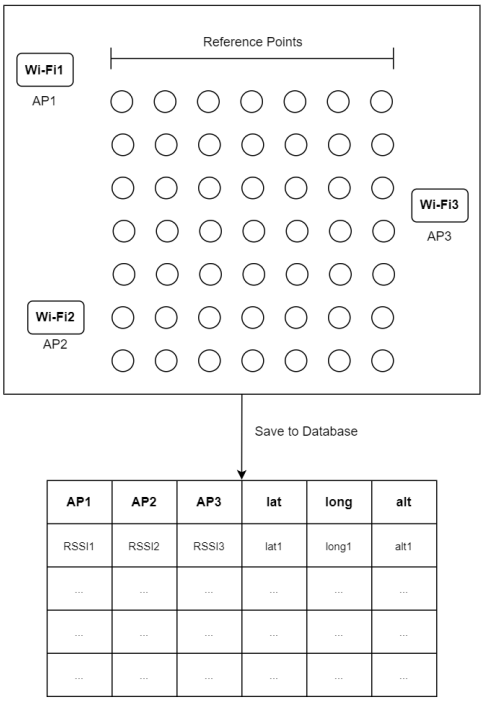

Figure 1. Offline phase of fingerprinting

The online phase of fingerprinting aims to predict a device real-time position by comparing the RSSI values received by the device to the fingerprints that are stored in the database. The device location is then predicted using an algorithm that searches for the most similar fingerprints in the database, such as KNN. The most similar fingerprints selection is chosen based on a certain similarity metric, such as the Euclidean distance of two RSSI vectors. From the obtained most similar fingerprints, another algorithm is used to predict the exact location of the device or to classify the RP where the device is.

However, one of the biggest problems in fingerprinting method is the searching speed for the online phase. The time complexity for comparing an RSSI vector to all fingerprints in the database increases substantially as the number of fingerprints and the number of access points increases [9]. Furthermore, since indoor positioning system aims to locate a person's real-time location indoors like GPS, a fast and accurate indoor positioning system method is required to ensure the position prediction is as accurate as possible and received as fast as possible. Therefore, many previous works have explored algorithms that can be used for indoor positioning systems to improve searching speed without severely compromising the indoor localization accuracy.

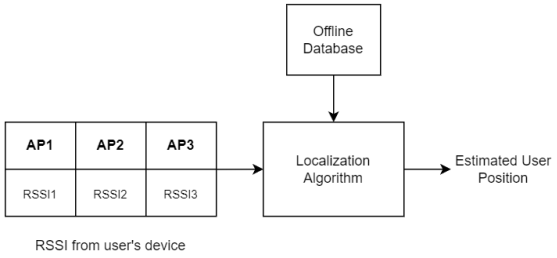

Figure 2. Online phase of fingerprinting

Research in indoor positioning system localization is largely divided into two categories. The first category aims to identify the exact location of a person according to a certain coordinate system, such as longitude and latitude or a coordinate relative to a certain point in an area. The other category focuses on classifying a person's position based on known coordinate frames. A coordinate frame is defined as an area inside a range of coordinate. Each coordinate frame is identified using a unique identifier and usually encompasses a certain room or area, such as an office, classroom, hallway, etc. This study focuses on the latter type of research.

In this study, the objective is to identify the most effective algorithm for efficiently locating coordinate frames to expedite indoor positioning systems, without compromising accuracy. This study aims to compare algorithms used to improve KNN search based on their search accuracy and search speed. Three main algorithms that are going to be compared are: Locality-Sensitive Hashing (LSH), Hierarchical Navigable Small World (HNSW) and Inverted File Index (IVF). LSH is chosen as one of the algorithms due to its effectiveness in performing an approximate nearest neighbor search in datasets containing high-dimensional vectors [10]. In comparison, HNSW, represents each dataset instance as a node in a multilayer graph so it enables the evaluation of only a small subset of connections for each element during a search [11]. IVF is also one of the more efficient similarity search algorithms by clustering the data points before doing a search [12]. These algorithms will be put to the test using the UJI Indoor Dataset, aiming to provide a clear understanding of how well LSH, HNSW and IVF perform similarity search, with a particular emphasis on search speed and accuracy.

2. RELATED WORKS

Several studies have sought to address the challenges related to the high computational cost of searching and processing algorithms in indoor positioning systems. In

the journal [13], researchers proposed a binary fingerprinting technique, which is considered highly suitable for conditions where the number of beacons is not a limiting factor in a design, such as BLE devices, RFID tags, or microsensors. The main concept of this technique is to reduce the quantization levels of the Received Signal Strength Indicator (RSSI) to a binary state. By using one bit to represent RSSI, a radio map with a binary vector can be obtained, and measurements during the online phase also become simple binary vectors. The results can be interpreted based on the concept of Hamming distance between vectors on the radio map, which can directly improve localization performance.

In the journal [14], the research focuses on solving computational cost issues, especially those caused by high-dimensional RSS vectors. Several proposed computation reduction techniques are applied both during the offline and online phases of fingerprinting. The first applied technique is dimension reduction using a modified version of the Fast Orthogonal Search (mFOS) algorithm, claimed to be faster and more accurate than other dimension reduction algorithms such as Principal Component Analysis (PCA). This dimension reduction technique running in the offline phase will later be combined with fast search strategies (Three Step Search, Orthogonal Search, and Diamond Search) and clustering using the KNN algorithm. The results of using mFOS and the hybrid solution (clustering and fast search) show that this technique is the most cost-efficient solution.

Research [15] combines the Weighted Centroid Localization (WCL) method with traditional fingerprinting to reduce the number of reference points (RP) in fingerprinting localization. The accuracy of predictions from this combined method is not significantly different from the accuracy of the Weighted k-Nearest Neighbors (WKNN) method and practical fingerprinting that uses more RPs. This method is much faster in the online fingerprinting phase and less time consuming in the offline fingerprinting phase since it uses fewer RPs.

A new approach to reduce searching complexity is introduced in [9]. The Hierarchical Navigable Small World (HNSW) is designed to improve the efficiency of indoor positioning system search compared to traditional KNN and tree-based methods. The HNSW technique uses a graph-based data structure with small-world characteristics, allowing efficient nearest neighbor searches. The formed graph will consist of several layers, where the first/top layer depicts the relationship between distant nodes, and the bottom layers depict the relationship between closer nodes, i.e., the relationships between nodes and their neighbors (nearest neighbors). The search process starts from the highest layer, where each node in this layer is visited greedily until finding the local optimum. After the local optimum is found, the search continues in the layer below. This process is iteratively repeated, starting from the minimum local found in the previous layer, until reaching the base layer. This solution outperforms classic KNN speed by 98% and at least 80% compared to other tree-based methods while maintaining accuracy. The HNSW technique shows promising results for large-scale indoor positioning system applications.

In [6], a low-complexity and memory efficient LSH function was proposed to reduce the computational cost of nearest neighbor search (NNS). The algorithm was built based on a randomized sum-to-one (STONE) transform [16]. Approximate hash matches were also used to further reduce the complexity since it reduces the size of the hash tables, thus improving the speed of the NNS. The proposed LSH-based positioning achieved the same accuracy as an exhaustive search over the channel state information (CSI) fingerprinting database, while being much more effective complexity-wise and storage-wise.

Study [17] proposed a system called infrastructureless Pedestrian Dead Reckoning (iPDR) for indoor pedestrian tracking. The term "infrastructureless" means that the system does not require any external infrastructure such as Wi-Fi Access Points, Bluetooth, and even previous knowledge such as map constraints, pre-trained vocabulary, or image database. The iPDR system utilizes several smartphone sensors as accelerator, magnetometer, gyroscope and camera to provide accurate position. The system integrates Hybrid Orientation Filter (HOF) that fuse information from the gyroscope, camera, and magnetometer to achieve accurate step orientation and Rapid Loop Detection (RLD) to summarize input camera frames. In the RLD process, Locality Sensitive Hashing (LSH) is used for rapid loop closure detection due to its efficiency in high dimensional similarity search. Due to this ability, LSH can maintain accurate tracking with a small amount of compute time.

Indoor positioning system localization using K-means Clustering and Bayesian Estimation is explored by Pinto in [18]. During the offline phase of fingerprinting, the proposed algorithm KLIP (K-means and Log-distance model-based Indoor Positioning) first uses K-means Clustering to cluster RSSI fingerprint samples into K log-distance models. Similar RSSI vectors are grouped into the same cluster based on Euclidean distance as the similarity function. Linear regression is then performed for each cluster to get the log-distance parameters for the cluster's log-distance path loss model. In the online phase, the nearest cluster to the queried RSSI vector is chosen based on the distance between the cluster's centroid and the RSSI vector. Afterwards, the position is estimated using the associated cluster's log-distance model and Bayesian estimation. KLIP generally outperforms traditional KNN

algorithm for data with fewer training points and outperforms traditional Bayes algorithm for any number of training points. Unlike KNN which produces better results with increased number of training points, KLIP performance does not increase alongside dataset size since the model regressor stabilizes with few training points. KLIP also scales much better than KNN in terms of computational cost.

Research [19] presents an approach using Wi-Fi fingerprinting and a random statistical method to improve accuracy and reliability. The random statistical method refers to statistical techniques used to process and standardize RSSI values collected from multiple Wi-Fi access points. Its goal is to provide a more accurate representation of the actual signal strength at each reference point. The localization process consists of two phases: offline and online. In the offline phase, a large number of Wi-Fi signals are collected to create a fingerprinting database. This database is processed and standardized before it is ready to be used. In the online phase, the Wi-Fi signals collected from the user are standardized and compared with the existing fingerprinting database using Mahalanobis distance to find the reference point with the minimum distance. The proposed method achieves a maximum positioning error of less than 0.75 meters, while the WKNN algorithm, which is used for comparison, achieves an average positioning error of 1.5 meters.

[20] utilizes BLE for indoor positioning system communication between signal transmitter and receivers. In this study, the authors conducted two types of evaluation to determine the localization accuracy, which are static analysis and dynamic analysis. In static analysis, RSSI measurements are taken on predetermined spots 40 times each. When collecting the data, RSSI values below a certain threshold, in this case set as -78 dBm, are discarded. The raw signals are then smoothed using simple moving average and weighted moving average approaches. The smoothed data is then used to estimate the user's location by using trilateration. The outputs of the last 5 trilateration positions are used to enhance prediction accuracy by calculating the centroid of those outputs. The results of this static analysis evaluation shows that trilateration using minimal signal threshold, data smoothing with simple moving average and centroid calculation produces the least positioning error compared to only data smoothing or only minimal signal threshold. The authors then conducted dynamic analysis, which is done by estimating the location of a moving user. It was found that it is impossible to determine a moving user's location in real time, hence only the user's predicted path can be saved for further analysis. From the results obtained, higher inaccuracy was found when the user was in areas with low signal strength, so the authors recommend increasing the number of beacons in such areas to improve positioning accuracy.

Research [7] also uses BLE based indoor positioning system for person localization in home. Kalman filter and Particle filter-based noise reduction methods are used to smoothen RSSI data. The authors use trilateration to track and predict a person's position within a coordinate frame. The experiment yielded an average error of 0.6 meters within 3 meters. Another experiment was done by the authors to classify a person's location. There were two types of classification that were done. The first was done by dividing a room into 1m x 1m grids, while the second used location-of-interest based classification, where only certain locations are labeled. Classification models, such as BayesNet and Random Forest, were then used to classify the person's location. Both methods showed good results with average accuracy of 95.94% and 99.4% respectively using the Random Forest classifier.

Most researches focuses on improving accuracy of indoor positioning systems [7, 13, 14, 15, 19, 20], but several algorithms have been proposed to also improve indoor positioning system similarity search speed such as using HNSW [5, 9], LSH [6, 17], K-means clustering [18] and dimensionality reduction [14], while some similarity search algorithms have not been explored in the context of indoor positioning systems such as IVF [8]. This research aims to evaluate IVF searching speed and accuracy compared to algorithms that have been tested in indoor positioning systems similarity search, namely HNSW and LSH.

3. METHODOLOGY

This section addresses the dataset utilized, the data preprocessing, as well as the methodologies applied in this research. The flow of the research is shown in Figure 3.

A. Dataset

In this study, we utilize the publicly available UJIIndoorLoc dataset obtained from Kaggle. The UJIIndoorLoc dataset was made by Torres-Sospedra et al. [21] with the aim of providing a comprehensive dataset to facilitate the comparison of various indoor positioning system methods. This dataset is expected to address limitations in literature, where exclusive databases are often utilized, promoting more consistent research practices. UJIIndoorLoc consists of 21049 data points containing RSSI (Received Signal Strength Indicator) from 520 different routers, real-world coordinates (latitude, longitude, and floor), BuildingID, SpaceID, relative position with respect to SpaceID, UserID, PhoneID, and Timestamp. The dataset was collected across three different buildings, each consisting of 4 to 5 floors. The description of each attribute in the dataset is shown in table 1.

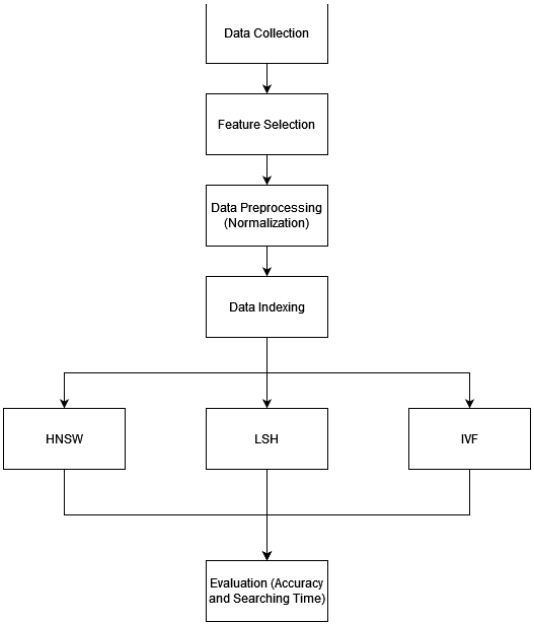

Figure 3. Experiment Flowchart

TABLE I. DESCRIPTION OF DATASET

Features	Description
WAP001 - WAP520	The RSSI value ranges from -104 to 0 for detected signals, and +100 is used when the Wireless Access Point (WAP) is not detected.
Latitude	Captured latitude coordinate of the user.
Longitude	Captured longitude coordinate of the user.
Floor	The floor level of the user at the time of capture.
BuildingID	ID used to identify the building when the data is captured.
SpaceID	ID used to identify the space such as office, corridor, classroom, etc.
RelativePosition	Values representing relative position to the space (1 for inside the room, 2 for outside the room/in front of the door).
UserID	Unique identifier utilized to distinguish each user who took the samples.
PhoneID	Unique identifier of an Android device that captures RSSI signals.
Timestamp	The timestamp indicates when the capture was taken, represented in UNIX time.

B. *Preprocessing*

Before doing the tests, the dataset undergoes an initial data cleaning process, which involves checking for null values or data anomalies. In our case, as there are no null values in the dataset, we can skip this process, streamlining our workflow. Following this, feature extraction is applied to the data to reduce dimensions and retain features that are relevant or significant for this analysis. Given our focus on searching the coordinate frame (SpaceID in this dataset), features that can be represented by SpaceID, such as Floor and BuildingID, are redundant and, therefore, we excluded from our dataset. Additionally, features unrelated to our coordinate frame search such as RelativePosition, UserID, PhoneID, Timestamp, longitude, and latitude are also excluded to enhance the specificity of our analysis. Following that, the RSSI is normalized using min-max. The equation of min-max scaling is described in equation (1).

$$X' = \frac{X - Xmin}{Xmax - Xmin} \quad (1)$$

This normalization process ensures that RSSI values are on a consistent scale, preventing undue influence from variables with high values that could impact the analysis process. After the data is fully preprocessed, the data is split to training and testing sets using an 80 to 20 split ratio. Following this, we proceed to apply the selected algorithms namely HNSW, LSH, and IVF to process the dataset

C. *Algorithm*

In this section, we explore the algorithms that we used to analyze the accuracy and search speed of similarity search in indoor positioning systems.

1) *Hierarchical Navigable Small World (HNSW)*

The Hierarchical Navigable Small World (HNSW) is a similarity search algorithm developed by combining two algorithms: Navigable Small World (NSW) and skip lists. In the NSW algorithm, data is represented as a proximity graph, where the searching process is conducted using a greedy search approach. In the search process, the algorithm starts from an entry point as the current node and traverses by calculating the distance from the query to nodes in the friend-list of the current node. The node with the closest distance to the query is chosen as the next

current node if its distance is closer compared to the current node's distance to the query. If the new node's distance is not closer to the query, the current node is returned as the answer, referred to as a local minimum [22]. This algorithm visualization can be seen in Figure 4.

Skip lists are a data structure that utilizes a probabilistic balancing approach. They consist of layers, each of which is composed of a linked list where the number of nodes may vary for each layer. The bottom layer represents the original linked list, and as you move upward, the number of skipped nodes increases. The search process begins from the top layer, traversing from the initial node. If the found node is larger than the target node, the search continues to the layer below. This process repeats until the search reaches the bottom layer, where the sought-after node is found. [23]. Skip lists algorithm visualization can be seen in Figure 5.

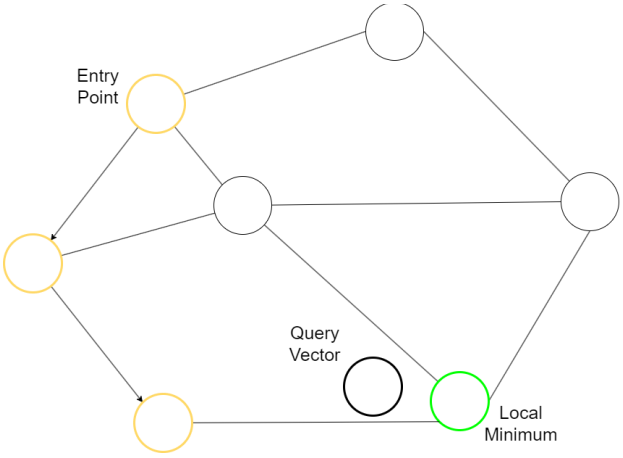

Figure 4. NSW Algorithm Visualization

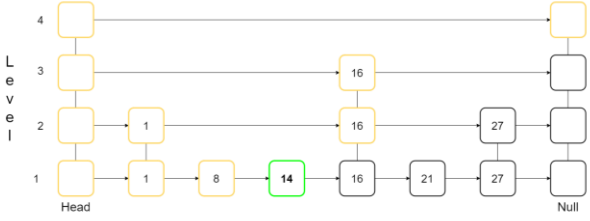

Figure 5. Skip List Algorithm Visualization

By combining the strengths of existing algorithms, the HNSW (Hierarchical Navigable Small World) algorithm emerges as an effective solution for performing robust nearest neighbor searches in high-dimensional spaces. HNSW is structured with multiple proximity graphs distributed across layers, where the bottom layer comprises complete nodes. Similar to skip lists, the search initiates at the first layer. The search process at each layer closely mirrors the NSW algorithm, wherein the search is halted upon reaching a local minimum. Upon identifying a local minimum, the search seamlessly proceeds to the lower layer and continues iteratively until it reaches the bottommost layer. The illustration of the idea of how HNSW works can be seen in Figure 6.

The number of maximum number of neighbors of each node need to be considered in HNSW to balance search speed and identifying nearest neighbor correctly. Having a proximity graph that is too dense (too many connections) in an upper layer will significantly slow the searching process, while sparser proximity graphs will lead to inaccurate results. Therefore, the size of dataset and the number of dimensions must be considered to choose the optimal parameters in HNSW. In this study, the maximum number of neighbors a node has is set to 60.

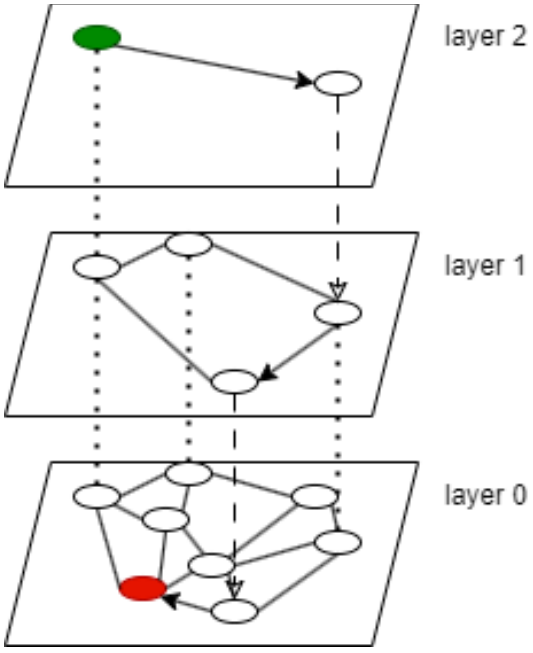

Figure 6. Basic Idea of HNSW Algorithm

2) Locality Sensitive Hashing (LSH)

Locality Sensitive Hashing (LSH) was introduced by Indyk and Motwani [24] as an approximate nearest neighbor (ANN) similarity search method in a large dataset with high dimensionality. The purpose of LSH is to reduce the very high time complexity for similarity search by reducing the number of pairs to compare in the dataset using hash tables. In other words, LSH can be viewed as a dimensionality reduction method. LSH works by creating multiple hash tables with different hash functions, where each data point p_i is stored in bucket $g_j(p_i)$ on hash table T_i after computing it using hash function g_i. Every data point in a bucket in a hash table is considered similar since it produces the same hashing for a certain hash function. For

the ANN query of a data point q, q is only compared to data points in bucket $g_j(q)$ which are considered similar. Afterwards, we can find the nearest neighbors based on some similarity search function using the original high-dimensional data (e.g., Euclidean distance or Cosine similarity).

Unlike other hashing algorithms, which aims to minimize collisions between different data points, LSH tries to maximize the chance of collisions of similar items. By maximizing the number of collisions, similar items can be grouped in the same bucket and the distribution of items will be more even. Choosing the hash function is crucial to correctly identify similar items. If a hash function does not classify a query to a bucket where its nearest neighbors are, the algorithm will produce a highly inaccurate approximation of the nearest neighbor. That is why multiple hash tables are used, in exchange for higher time and space complexity. Figure 7 below illustrates the fundamental concept of how LSH operates.

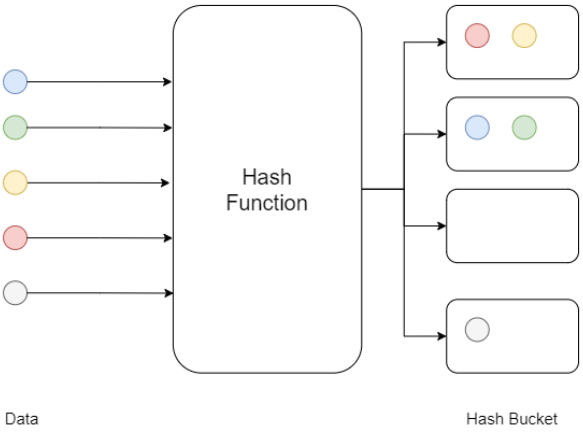

Figure 7. Basic Idea of LSH Algorithm

In LSH, the length of the resulting hash affects the performance and accuracy of the similarity search. Increasing the hash length will result in faster but less accurate search because the probability of two vectors having the same hash is reduced, hence reducing the number of pairs to compare. On the other hand, decreasing the hash length will significantly increase the search time because there will be more pairs to compare. Having multiple hash tables also allows for more robust search although the computational time will be slower. In this study, we set the length of the hash size for LSH to 6-bits and the number of hash tables used is 3 to match the other algorithms accuracy.

3) Inverted File Index (IVF)

Inverted File Index (IVF) was proposed by Amato and Savino [12] as an alternative for approximate similarity search in metric spaces. This method is because two distinct objects are considered "similar" if they are close according to a certain distance metric. In the context of similarity search, the vectors are divided into several clusters in the training phase in a similar approach to k-means clustering [25]. When performing a query, the algorithm searches for a centroid with the least distance to the current vector. Then the query vector is compared to other vectors in that cluster to find the most similar vector in the indexed database. Visualization of the concepts of the IVF algorithm can be seen in Figure 8.

In indoor positioning systems, building an index using IVF can be seen as doing clustering on vectors of RSSI. After each fingerprint is assigned to a cluster, searching can be done by searching for the nearest centroid to the query fingerprint based on Euclidean distance or other distance metric. Then, each data in the cluster will be compared to the query using the same distance metric to find the nearest neighbor. Searching in multiple clusters can increase nearest neighbor search accuracy, although it will make the process slower.

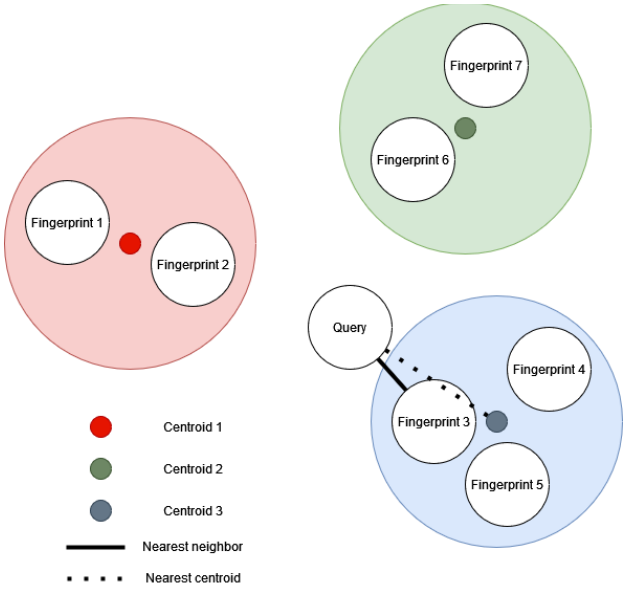

Figure 8. Visualization of IVF Algorithm

Determining the number of clusters is very important in IVF. If too many clusters are made, there will be less elements to compare in each cluster and the accuracy will be worse. Conversely, if too few clusters are made, the searching speed will be slower because for each query, more comparisons must be made. In this study, the number of clusters is set to 50 since it produces the best accuracy-to-time ratio.

4. EXPERIMENTS

In our experiment, each algorithm was run 10 times during the similarity search process to obtain the average search speed performance for each algorithm. This experiment was conducted using a computer with AMD

Ryzen 5 4600H CPU @ 3.00GHz with 16.0 GB of memory.

A. Experimental Design

To obtain optimal performance measurements, we utilized two evaluation metrics: speed and accuracy. These metrics were selected to align with the requirements of the indoor positioning system, where position determination needs to happen swiftly (ideally in real-time) while maintaining high accuracy.

The first metric is the speed in conducting similarity search. In this experiment, speed involves the search for the K most similar points and determining the correct SpaceID. The prediction is established based on the SpaceID that appears most frequently in the list containing the K points. If there is more than one SpaceID that appears most frequently, the SpaceID with the smallest average distance from the query vector will be selected.

The next crucial metric is accuracy, holding equal importance in our evaluation. In this experiment, accuracy assessment involves comparing correct predictions to the overall predicted data volume. A prediction is correct if the predicted SpaceID matches the actual SpaceID in the original dataset. The precise calculation for accuracy is delineated by equation (2).

$$Accuracy = \frac{number\ of\ correct\ prediction}{number\ of\ test\ data} \quad (2)$$

B. Experimental Results

In this section, we will compare the outcomes of the three selected algorithms, namely IVF, LSH, and HNSW. The evaluation variables include the speed and accuracy of each algorithm in performing the similarity search process. A detailed breakdown of the accuracy and speed comparisons, along with the values of K representing the number of nearest neighbor points, is described in Table II and Table III.

TABLE II. ACCURACY OF EACH ALGORITHM

Algo.	**K Value**					
	1	**3**	**5**	**7**	**9**	**11**
HNSW	0.741	0.727	0.709	0.689	0.675	0.659
LSH	0.725	0.707	0.678	0.657	0.634	0.613
IVF	0.732	0.715	0.694	0.673	0.658	0.640

TABLE III. SEARCH SPEED OF EACH ALGORITHM

Algo.	**K Value**					
	1	**3**	**5**	**7**	**9**	**11**
HNSW	0.014	0.020	0.024	0.029	0.034	0.041
LSH	4.813	4.802	4.744	4.746	4.760	4.729
IVF	0.016	0.020	0.024	0.030	0.035	0.038

HNSW has proven to be the most optimal choice for implementation as a similarity search technique in indoor positioning systems compared to the other two algorithms. The superiority of HNSW is evident through its commendable performance, showcasing the highest accuracy and the most efficient search time. Meanwhile, the IVF and LSH algorithms closely rival HNSW in terms of accuracy. However, for LSH, the time required to execute similarity searches is considered quite high compared to other algorithms, needing an average of 4.813 seconds per query compared to HNSW and LSH time of 0.014 and 0.016 seconds, respectively. This aspect can potentially hinder indoor positioning system ability to present data in real-time when using LSH as the searching method.

From the results of the conducted experiments, it is apparent that all three algorithms accuracy does not increase as the value of K increases. In fact, each algorithms' accuracy is inversely correlated to the value of K, with the highest accuracy achieved when the value of K is lowest, i.e., only one nearest neighbor. An increase in searching time is also noticeable when the value of K is elevated in the HNSW and IVF algorithms. However, in the LSH algorithm, the increase in the value of K does not proportionally correlate with an increase in searching time. This indicates a distinct characteristic response difference between the LSH algorithm compared to HNSW and IVF in handling variations in the value of K.

5. CONCLUSIONS AND FUTURE WORKS

This study focuses on comparing the search speed of different similarity search algorithms in indoor positioning systems during online phase, namely HNSW, LSH and IVF. The algorithms were tested on UJIIndoorLoc dataset which contains 21049 data points and aims to predict the SpaceID of a data point based on the given RSSI values from 520 different access points. Our results show that HNSW has the best average computational speed for various K values, as well as the best accuracy among the other algorithms. LSH is the worst performing model in both aspects, being slower than HNSW and IVF by more than 34000%. Each model's accuracy decreased as we increased the K value, meaning the best accuracy was achieved with K = 1.

We only utilized the training dataset in this research and since the validation dataset does not contain SpaceID. There are also some overlapping SpaceID for data points that are on different floors. These data points that have the same SpaceID have unsimilar RSSI. Due to these reasons, the prediction accuracy for SpaceID may not be optimal. This also explains why higher K values for nearest neighbor search give worse results than when K = 1. For future works, a more suitable dataset may be used to predict the coordinate frame in indoor positioning systems.

Since we used personal computer to run this experiment, the time needed for doing the searching and prediction will vary depending on the computer used. Faster computer may have better results for the localization speed, and vice versa. Experiments conducted using a computer with better specification may be used in the future to better simulate the search speed of indoor positioning system for real world scenario.

ACKNOWLEDGMENT

The authors acknowledge Torres-Sospedra et al. for providing the publicly available UJIIndoorLoc dataset.

REFERENCES

[1] N. S. C. Jailani, N. H. A. Wahab, N. Sunar and S. H. S. Ariffin, "Indoor Positioning System: A Review," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 6, pp. 477-490, 2022.

[2] M. E. Hossen, K. Kamardin, S. N. Maidin and T. D. N. W. Hlaing, "Wi-Fi Fingerprinting for Indoor Positioning," *International Journal of Integrated Engineering*, vol. 14, no. 6, pp. 223-238, 2022.

[3] N. Al-Sabbagh and A. Al-Omary, "A Centralized Multi-Floor Indoor Navigation," *International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, 2019.

[4] G. Shipkovenski, T. Kalushkov, E. Petkov and V. Angelov, "A Beacon-Based Indoor Positioning System for Location Tracking of Patients in a Hospital," *2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA)*, pp. 1-6, 2020.

[5] R. Giuliano, G. C. Cardarilli, C. Cesarini, L. D. Nunzio, F. Fallucchi, R. Fazzolari, F. Mazzenga, M. Re and A. Vizzarri, "Indoor Localization System Based on Bluetooth Low Energy for Museum Applications," *Electronics*, vol. 9, no. 6, 2020.

[6] G. Schroeer, "A Real-Time UWB Multi-Channel Indoor," *IEEE*, 2018.

[7] B. Lu, F. Ciravegna, R. Bond and M. Mulvenna, "A Low Cost Indoor Positioning System Using Bluetooth Low Energy," *IEEE Access*, vol. 8, pp. 136858-136871, 2020.

[8] M. M. K. Lie and G. P. Kusuma, "A fingerprint-based coarse-to-fine algorithm for indoor positioning system using Bluetooth Low Energy," *Neural Computing and Applications*, vol. 33, pp. 2735-2751, 2021.

[9] M. Lima, L. Guimarães, E. Santos and E. Moura, "A Small World Graph Approach for an Efficient Indoor," *Sensors*, vol. 21, no. 15, pp. 1-19, 2021.

[10] L. Tang, R. Ghods and C. Studer, "Reducing the Complexity of Fingerprinting-Based Positioning using Locality-Sensitive Hashing," pp. 1086-1090, 2020.

[11] M. W. S. Lima, H. A. B. F. d. Oliveira, E. Santos and E. S. d. Moura, "Efficient and Robust WiFi Indoor Positioning using Hierarchical Navigable Small World Graphs," *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*, pp. 1-5, 2018.

[12] G. Amato and P. Savino, "Approximate similarity search in metric spaces using inverted files.," *InfoScale '08: Proceedings of the 3rd international conference on scalable information systems, ICST.*, pp. 1-10, 2008.

[13] M. Mizmizi and L. Reggiani, "Binary Fingerprinting-Based," *INTERNATIONAL CONFERENCE ON INDOOR POSITIONING AND INDOOR NAVIGATION (IPIN)*, pp. 1-6, 2017.

[14] A. Abusara, M. S. Hassan and M. H. Ismail, "Reduced-complexity fingerprinting in WLAN-based indoor," *Springer*, 2016.

[15] S. Subedi and J.-Y. Pyun, "Practical Fingerprinting Localization for Indoor Positioning," *Journal of Sensors*, pp. 1-16, 2017.

[16] T. Goldstein, L. Xu, K. F. Kelly and R. Baraniuk, "The STONE Transform: Multi-Resolution Image Enhancement and Real-Time Compressive Video," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5581-5593, 2013.

[17] Z. Yang, Y. Pan, Q. Tian and R. Huan, "Real-time Infrastructureless Indoor Tracking for Pedestrian Using a Smartphone," *IEEE Sensors Journal*, vol. 19, no. 22, pp. 10782-10795, 2019.

[18] B. H. O. U. V. Pinto, "Robust RSSI-based indoor positioning system using K-means clustering and Bayesian estimation," *IEEE Sensors Journal*, vol. 21, no. 21, pp. 24462-24470, 2021.

[19] D. B. Ninh, J. He, V. T. Trung and D. P. Huy, "An effective random statistical method for Indoor Positioning System using WiFi fingerprinting," *Future Generation Computer Systems*, vol. 109, pp. 238-248, 2020.

[20] R. Bembenik and K. Falcman, "BLE Indoor Positioning System Using RSSI-based Trilateration," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications (JoWUA)*, vol. 11, no. 3, pp. 50-69, 2020.

[21] J. Torres-Sospedra, R. Montoliu, A. Mart´ınez-Uso, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau and J. Huerta, "UJIIndoorLoc: A New Multi-building and Multi-floor Database for WLAN Fingerprint-based Indoor Localization Problems," *International Conference on Indoor Positioning and Indoor Navigation*, pp. 261-270, 2014.

[22] Y. Malkov, A. Ponomarenko, A. Logvinov and V. Krylov, "Approximate Nearest Neighbor Algorithm based on Navigable Small World Graphs," *Information Systems.*

[23] W. Pugh, "Skip Lists: A Probabilistic Alternative to Balanced Trees," *Algorithms and Data Structures* , vol. 33, no. 6, pp. 668-676, 1990.

[24] P. Indyk and R. Motwani, "Approximate Nearest Neighbors: towards Removing the curse of Dimensionality," *STOC*, pp. 604-613, 1998.

[25] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 100-108, 1979.

[26] N. Swangmuang and P. Krishnamurthy, "Location Fingerprint Analyses Toward Efficient," *Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pp. 100-109, 2008.

Jevon Sebastian is a graduate student of Master of Computer Science Department in Bina Nusantara University. and a short biography. His research interests include machine learning and deep learning.

Justin Orlean is a graduate student of Master of Computer Science Department in Bina Nusantara University. and a short biography. His research interests include machine learning, deep learning and heuristics.

Gede Putra Kusuma received PhD degree in Electrical and Electronic Engineering from Nanyang Technological University (NTU), Singapore, in 2013. He is currently working as a Lecturer and Head of Department of Master of Computer Science, Bina Nusantara University, Indonesia. Before joining Bina Nusantara University, he was working as a Research Scientist in I2R – A*STAR, Singapore. His research interests include computer vision, deep learning, face recognition, appearance-based object recognition, gamification of learning, and indoor positioning system.