ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

Fusion Localization Algorithm for Non-Line-of-Sight Environment

Yujie Yang

Hunan University of Science and Technology, Xiangtan, China
yang15073879822@163.com

Abstract: *In indoor non-line-of-sight (NLOS) environments, wireless signals are often disrupted by obstacles, multipath effects, and other factors, which severely impact the performance and accuracy of positioning systems. To address this issue, an indoor positioning algorithm is proposed that integrates the K nearest neighbor (KNN) algorithm with the least squares (LS) method. First, the received signal strength (RSS) fingerprint database is corrected to reduce error fluctuations caused by NLOS conditions, providing a more stable foundation for positioning. Then, the KNN algorithm is applied to the corrected fingerprint library to compute an initial position estimate for the target, leveraging proximity-based signal characteristics in the target's vicinity. Finally, the LS method is used to refine the initial position estimate from KNN, minimizing residual errors and enhancing positioning accuracy. Experimental results demonstrate that, in NLOS environments, the proposed algorithm significantly outperforms standalone KNN and LS methods, achieving superior accuracy and positioning performance. This integrated approach shows promise for improving accuracy in indoor positioning systems under challenging NLOS conditions.*

Keywords: *Non-Line-of-Sight; Fingerprint Library; Localization; Received Signal Strength*

1. Introduction

In recent years, the advent of mobile smart phones has made mobile communication increasingly popular, and most users can use the Global Positioning System (GPS) to locate their position, and GPS has a satisfactory positioning accuracy in outdoor areas for most people.[1].However, GPS is not suitable for indoor environments because of the large number of Non-Line-of-Sight (NLOS) environments that exist indoors, and it cannot provide sufficiently accurate positioning. A lot of research has been done on indoor positioning, and among all the proposed technologies, WiFi is one of the most promising technological approaches, and most of the indoor scenarios where we are in our daily life are usually equipped with some kind of WiFi devices, and WiFi is even the most basic and necessary feature of smartphones.[2]

Received Signal Strength (RSS) is the strength or power of the wireless signal received by the receiving device from the transmitting source, which can be obtained from various WiFi devices due to its ease of implementation and widespread use in various indoor positioning systems.[3-4] In reality, variations in RSS tend to affect the positioning results[5]and hence the accuracy, especially when large obstacles are present, the impact on RSS is huge. [6]

Machine learning based indoor localization algorithms are very popular nowadays, such as KNN (K Nearest Neighbor),[7] Random Forest, Support Vector Machines. For example, KNN, a fingerprint library-based localization method, requires the creation of a fingerprint library offline before localization, and then the algorithm is used for localization at that stage. This non-line-of-sight localization method[8] is highly resistant to various obstacles present in the room itself. As for the line-of-sight localization algorithms, including triangulation, least squares localization and maximum likelihood estimation algorithms, these traditional algorithms have high localization accuracy in the absence of obstacle interference, but the localization accuracy decreases significantly under the interference of other obstacles or large fluctuations in RSS.

To address the above problems, the fingerprint library was calibrated and then the different algorithms were fused together to obtain an algorithm with higher localisation accuracy. Based on the constructed RSS fingerprint database, the KNN algorithm is used to match the reference position as the initial estimated target position, and then the least squares method is used to accurately adjust the target position.

ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

Experiments verify that the proposed algorithm can effectively deal with target positioning in non-line-of-sight environments with better robustness and stability.

2. Fingerprint Positioning Methods

2.1. KNN localization algorithm

KNN (K Nearest Neighbour) algorithm is a classical machine learning algorithm, before carrying out the calculation, it is first necessary to set the value of k to facilitate the subsequent calculations, through the formula to calculate the results of multiple preliminary estimates, and then arrange these results from small to large, take the first k results, and then take the average of these k results, then you can get the final results of the KNN algorithm. The application in indoor positioning is as follows.

Firstly, the value of k is set first and its Euclidean distance D from each set of fingerprints of the fingerprint library is calculated using the RSS fingerprint library built in the offline phase and the RSS_fp={R(1),R(2),...,R(K)} received at the mobile node (the node that needs to be localized), where R(1) is the RSS value of the first anchor received by that node. as follows.

$$D = \begin{bmatrix} \left(RSSf(x_1, y_1, 1) - R(1)\right)^2 + \ldots + \left(RSSf(x_1, y_1, K) - R(K)\right)^2 \\ \left(RSSf(x_2, y_2, 1) - R(1)\right)^2 + \ldots + \left(RSSf(x_2, y_2, K) - R(K)\right)^2 \\ \ldots \\ \left(RSSf(x_N, y_N, 1) - R(1)\right)^2 + \ldots + \left(RSSf(x_N, y_N, K) - R(K)\right)^2 \end{bmatrix} \tag{1}$$

where $RSSf(x_1, y_1, 1)$ is the RSS value of the first anchor point received at the first grid (x_1, y_1) point, and $\left(RSSf(x_1, y_1, 1) - R(1)\right)^2 + \ldots + \left(RSSf(x_1, y_1, K) - R(K)\right)^2$ is the Euclidean distance between the point(x_1, y_1) and the predicted point at (x_1, y_1),after getting all the values in D, the Euclidean distances are sorted, and then the first k values with the smallest Euclidean distances, e.g., $(x_i, y_i), \ldots, (x_k, y_k)$ are averaged, and that is the final result of the localization:

$$P = (x, y) = \left(\frac{x_1 + x_2 + .. + x_k}{k}, \frac{y_1 + y_2 + .. + y_k}{k}\right) \tag{2}$$

2.2. Least-squares localization

The least squares method does not require a fingerprint library to aid in localisation, it simply receives the RSS values sent from the K anchor points and then uses these K RSS values to be able to achieve the localisation goal, the distance between the predicted point and each anchor point can be calculated using these RSS values, by using the following equation.

$$dist = d0 * 10^{(Pt - Pl\_d0)/(10 * eta)} \tag{3}$$

Assuming that (x, y) is the position we require, the equations can be listed using its distance from each anchor point, as follows to obtain K equations that

$$\begin{cases} (X_1 - x)^2 + (Y_1 - y)^2 = dist_1^2 \\ (X_2 - x)^2 + (Y_2 - y)^2 = dist_2^2 \\ \ldots \\ (X_K - x)^2 + (Y_K - y)^2 = dist_K^2 \end{cases} \tag{4}$$

Where $dist_1$ is the distance from (x, y) to the anchor point (X_1, Y_1), subtracting each equation from the second line of the above equation from the first line of the above equation, expanding them and then simplifying them, we can get

$$\begin{cases} 2(X_2 - X_1)x + 2(Y_2 - Y_1)y = X_2^2 - X_1^2 + Y_2^2 - Y_1^2 - dist_2^2 + dist_1^2 \\ 2(X_3 - X_1)x + 2(Y_3 - Y_1)y = X_3^2 - X_1^2 + Y_3^2 - Y_1^2 - dist_3^2 + dist_1^2 \\ \ldots \\ 2(X_K - X_1)x + 2(Y_K - Y_1)y = X_K^2 - X_1^2 + Y_K^2 - Y_1^2 - dist_K^2 + dist_1^2 \end{cases} \tag{5}$$

Where $dist_i$ is the distance from the predicted position to the ith anchor node.

ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

The above equation can be reduced to the matrix$AX = B$.

$$A = \begin{bmatrix} 2(X_2 - X_1) & 2(Y_2 - Y_1) \\ 2(X_3 - X_1) & 2(Y_3 - Y_1) \\ & \cdots \\ 2(X_K - X_1) & 2(Y_K - Y_1) \end{bmatrix} \tag{6}$$

$$B = \begin{bmatrix} X_2^2 - X_1^2 + Y_2^2 - Y_1^2 - dist_2^2 + dist_1^2 \\ X_3^2 - X_1^2 + Y_3^2 - Y_1^2 - dist_3^2 + dist_1^2 \\ \cdots \\ X_K^2 - X_1^2 + Y_K^2 - Y_1^2 - dist_K^2 + dist_1^2 \end{bmatrix} \tag{7}$$

The value of $P = (x, y)$ is calculated by matrix multiplication and division, and is then the result of localization.

3. Positional fingerprint library established

3.1. Establishment of a fingerprint database

The fingerprint library is the set of RSS data received by all grid points from all anchor points,[10] we partition the indoor environment into N grid points, where each grid is numbered with grid_n (grid_n = 1, 2, ... , N) for numbering. Then the position of the center point of each grid is recorded in its entirety and written into the matrix grid_xy as follows:

$$grid_{xy} = \{(x_1, y_2), (x_2, y_2), \dots, (x_N, y_N)\} \tag{8}$$

Where $grid_{xy}(grid\_n) = (x_n, y_n)$

We have K localized anchor nodes (APs) as follows:

$$anchor = [(X_1, Y_1), (X_2, Y_2), \dots, (X_K, Y_K)] \tag{9}$$

We write the RSS received at the center of each grid from a localized anchor node into the matrix, such as the first localized anchor node, the RSS received at the center of each grid from anchor 1 is written one by one according to the grid number into the matrix $RSSfinger(1)$, and the RSS received at the centre of grid n from anchor 1 is $RSSfinger(x_n, x_n)$.

$$RSSfinger = \begin{bmatrix} RSSf(x_1, y_1, 1), RSSf(x_2, y_2, 1), \dots, RSSf(x_N, y_N, 1) \\ RSSf(x_1, y_1, 2), RSSf(x_2, y_2, 2), \dots, RSSf(x_N, y_N, 2) \\ \cdots \\ RSSf(x_1, y_1, K), RSSf(x_2, y_2, K), \dots, RSSf(x_N, y_N, K) \end{bmatrix} \tag{10}$$

Where $RSSf(x_1, y_1, 1)$ denotes the RSS received at (x_1, y_1),i.e. at the centre of the first grid, from the location anchor node 1.

After that test data can be created to test whether this localization method is accurate. We randomly lay out 50 test data points, where the ith data is $p\_true = (x_i, y_i)$, and then derive a test fingerprint for p_true(i)

$$RSS_{fp}(i) = \{R(1), R(2), \dots, R(K)\} \tag{11}$$

3.2. Fingerprint library calibration

Set up an obstacle, the obstacle is a straight line, the location of the obstacle is as follows,

$$line = \begin{bmatrix} x\_start & x\_end \\ y\_start & y\_end \end{bmatrix} \tag{12}$$

When building a fingerprint library, for example when building a fingerprint of (x_i, y_i), a judgement can be made as to whether or not an obstacle passes between (x_i, y_i) and each anchor point. This can be seen as a judgement of whether two directs are comparable, for example to judge whether there is an obstacle between (x_i, y_i) and anchor point 1, the following two straight lines can be used to determine

ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

that the

$$line1 = ([x_{start}, x_{end}], [y_{start}, y_{end}]), line2 = ([x_i, X_i], [y_i, Y_i]) \tag{13}$$

Firstly, whether the two line segments intersect or not is judged, if they intersect then it is considered that there is an obstacle, then a random noise $er(x_i, y_i, 1)$ is set (between the anchor node 1 and (x_i, y_i)), and vice versa, $er(x_i, y_i, 1) = 0$. The new fingerprint after fingerprint library correction is

$$RSSf(x_1, y_1, 1) = RSSf(x_1, y_1, 1) + er(x_1, y_1, 1) \tag{14}$$

4. Localization algorithms incorporating KNN and least squares

In indoor localization, the KNN algorithm and the least squares method have their own advantages and disadvantages. The KNN algorithm is suitable for dealing with nonlinear and complex environments for initial estimation, while the two least squares method excels in exact optimisation for more linear errors. The combination of both can give a better result. The fusion algorithm is as follows:

Firstly, we set the value of K in the KNN algorithm and the number of anchor nodes N, and also use the previously derived fingerprint library RSSfinger to calculate the Euclidean distance D from each set of fingerprints of the fingerprint library D after obtaining the $RSS_{fp} = \{R(1), R(2), \dots, R(K)\}$,received by the mobile node (the one that needs to be localised), as follows shown.

$$D = \begin{bmatrix} d_1 = \left(RSSf(x_1, y_1, 1) - R(1)\right)^2 + \dots + \left(RSSf(x_1, y_1, K) - R(K)\right)^2 \\ d_2 = \left(RSSf(x_2, y_2, 1) - R(1)\right)^2 + \dots + \left(RSSf(x_2, y_2, K) - R(K)\right)^2 \\ \dots \\ d_N = \left(RSSf(x_N, y_N, 1) - R(1)\right)^2 + \dots + \left(RSSf(x_N, y_N, K) - R(K)\right)^2 \end{bmatrix} \tag{15}$$

After sorting the Euclidean distances and renumbering the first k points with the smallest Euclidean distances, the first k estimated coordinates were obtained as follows.

$$knnp = [(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)] \tag{16}$$

$knnp$ is the first k coordinates with the smallest Euclidean distance, which can then be optimised by least squares to achieve the fusion of these two algorithms, we can use these k coordinates with their Euclidean distances to perform least squares, which can be listed in the following equation, the

$$\begin{cases} (x_1 - x)^2 + (y_1 - y)^2 = d_1^2 \\ (x_2 - x)^2 + (y_2 - y)^2 = d_2^2 \\ \dots \\ (x_k - x)^2 + (y_k - y)^2 = d_k^2 \end{cases} \tag{17}$$

Similarly the above equation can be obtained by subtracting each equation from the second row onwards from the equation in the first row, expanding them and then simplifying them to get

$$\begin{cases} 2(x_2 - x_1)x + 2(y_2 - y_1)y = x_2^2 - x_1^2 + y_2^2 - y_1^2 - d_2^2 + d_1^2 \\ 2(x_3 - x_1)x + 2(y_3 - y_1)y = x_3^2 - x_1^2 + y_3^2 - y_1^2 - d_3^2 + d_1^2 \\ \dots \\ 2(x_k - x_1)x + 2(y_k - y_1)y = x_K^2 - x_1^2 + y_K^2 - y_1^2 - d_K^2 + d_1^2 \end{cases} \tag{18}$$

The above equation can be reduced to the matrix $AX = B$。

$$A = \begin{bmatrix} 2(x_2 - x_1) & 2(y_2 - y_1) \\ 2(x_3 - x_1) & 2(y_3 - y_1) \\ \dots & \\ 2(x_K - x_1) & 2(y_K - y_1) \end{bmatrix} \tag{19}$$

ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

$$B = \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 - d_2^2 + d_1^2 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 - d_3^2 + d_1^2 \\ \cdots \\ x_K^2 - x_1^2 + y_k^2 - y_1^2 - d_k^2 + d_1^2 \end{bmatrix} \tag{20}$$

The value of the final positioning result $P = (x, y)$ is calculated by matrix multiplication and division.

Then set a threshold ε to determine whether the overall KNN preliminary localization result is offset too much, that is to say, there is an overall distortion, then in order to reduce the error and improve the localization accuracy, take the first value d_1 of the reordering of the Euclidean distance calculated by the previous KNN, if there is a $d_1 > \varepsilon$, then it is determined that there is a distortion in the KNN algorithm, and then only take the first n (n < k) numbers to do the average, to get the final localization result is:

$$P = \left(\frac{x_1+x_2+\cdots+x_n}{n}, \frac{y_1+y_2+\cdots+y_n}{n}\right) \tag{21}$$

5. Performance Analysis

Firstly, 5 positioning anchor nodes and 1 obstacle are set up, then 50 real coordinate points are placed to test the positioning accuracy of the three algorithms, and the actual coordinate distribution is shown in Figure 1.

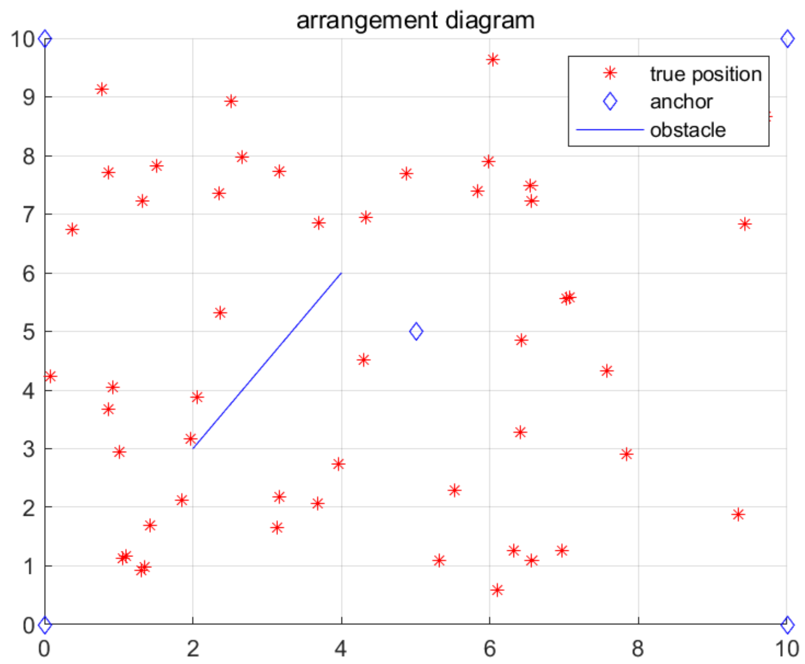

Figure 1: Real coordinate distribution map

After establishing the fingerprint library and performing the fingerprint library correction, set the K-value, in this case we set it to 7, and view the three algorithms localization results respectively, as shown in the following Figure 2.

ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

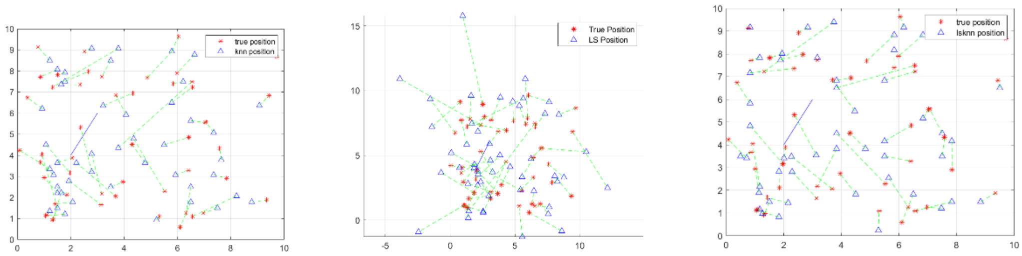

Figure 2: The positioning results of the three algorithms

The KNN localization results, the least squares localization results and the LSKNN localization results are shown in Figure5.2. The KNN localization error is 1.0976,the LS localization error is 2.4170,and the LSKNN localization error is 1.0584.Under these conditions it can be seen that the LSKNN has the best overall localization performance.

After changing the value of K in the KNN as well as LSKNN algorithms, other things being equal, after calculating their localization errors, the following results can be obtained. as shown in the following Figure 3.

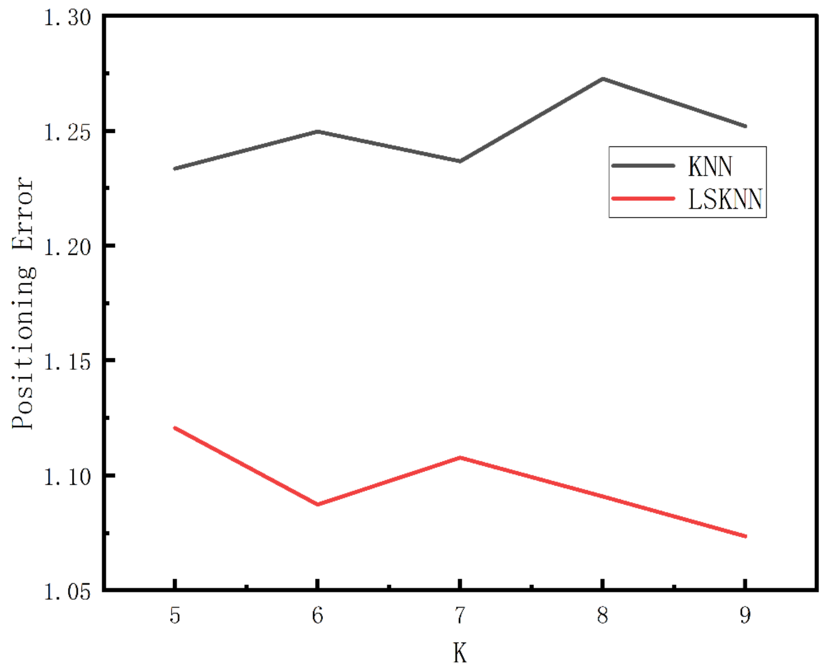

Figure 3: The performance of the two algorithms under different K values

From the above figure, it can be seen that there is no obvious change in the positioning accuracy of KNN after the K value increases and changes, while after the fusion algorithm LSKNN after the K value increases and changes, the positioning accuracy is a little bit deceptive and ups and downs, but the overall trend of upward movement, which indicates that after fusion of the algorithm, the K value in a certain range of the increase will improve the positioning accuracy.

After setting the value of K constant at 7 and increasing the number of anchor points, the change in accuracy of the three localization algorithms is observed as shown in the following Figure 4.

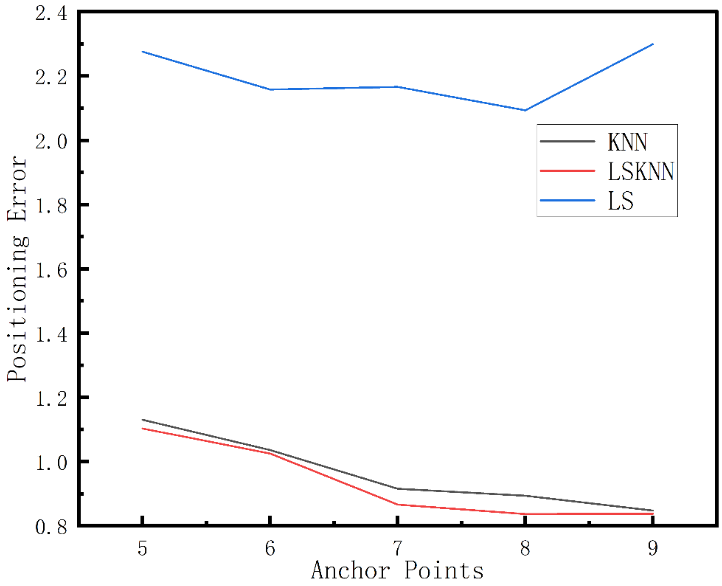

Figure 4: The performance of the three algorithms under different number of anchor points

From the above figure, it can be seen that the LS least squares method does not show any significant increase in localization accuracy even with an increase in the number of anchor points, whereas the KNN as well as the LSKNN show a steady increase in localization accuracy and the fusion algorithm LSKNN basically has the best localization accuracy.

The results of fusion localization under a variety of simulations all result in the smallest error for the fusion localization method, indicating good adaptability.

6. Conclusion

In today's indoor positioning demand is getting higher and higher, the requirement of positioning accuracy is also getting higher and higher. This paper firstly proposes a fingerprint database error correction method, which can effectively correspond to the error in reality and increase the positioning accuracy. Secondly, this paper also fuses the two algorithms, KNN and least squares, which have their own advantages and disadvantages, to test how the two algorithms and the fused algorithm locate under different K values and different anchor points, and the test results show that the fused localization algorithm has a better localization accuracy compared to the two non-fused algorithms.

References

[1] Xiansheng Guo, Shilin Zhu, Lin Li, Fangzi Hu & Nirwan Ansari.(2019). Accurate WiFi Localization by Unsupervised Fusion of Extended Candidate Location Set. IEEE Internet of Things Journal(2),2476-2485.

[2] Guo Xiansheng, Li Lin, Ansari Nirwan & Liao Bin. (2018). Knowledge Aided Adaptive Localization via Global Fusion Profile. IEEE Internet of Things Journal(2),1081-1089.

[3] Mehrabian Hamidreza & Ravanmehr Reza. (2023). Sensor fusion for indoor positioning system through improved RSSI and PDR methods. Future Generation Computer Systems, 254-269.

[4] Xiansheng Guo, Nirwan Ansari, Fangzi Hu, Yuan Shao, Nkrow Raphael Elikplim & Lin Li. (2020). A Survey on Fusion-Based Indoor Positioning. IEEE Communications Surveys & Tutorials(1), 566-594.

[5] Naveed Salman, Mounir Ghogho & Andrew H. Kemp. (2014).Optimized Low Complexity Sensor Node Positioning in Wireless Sensor Networks. IEEE Sensors Journal(1), 39-46.

[6] Ritabroto Ganguly, Manjarini Mallik & Chandreyee Chowdhury.(2024). Design of a knowledge distillation network for wifi-based indoor localization. Multimedia Tools and Applications (prepublish), 1-17.

[7] Bohang Chen, Jun Ma, Lingfei Zhang, Zhuang Xiong, Jinyu Fan & Haiming Lan. (2023). An improved weighted KNN fingerprint positioning algorithm. Wireless Networks(6),6011-6022.

[8] Wang Zhiyuan, Huang Huiling, Li Haoran, Chen Ziyang, Han Jun & Pu Jixiong. (2023).Non-line-

ISSN 2616-5775 Vol. 7, Issue 11: 142-149, DOI: 10.25236/AJCIS.2024.071119

of-sight imaging and location determination using deep learning. Optics and Lasers in Engineering
[9] Jingxue Bi, Yunjia Wang, Xin Li, Hongxia Qi, Hongji Cao & Shenglei Xu.(2018).An Adaptive Weighted KNN Positioning Method Based on Omnidirectional Fingerprint Database and Twice Affinity Propagation Clustering. Sensors(8),2502-2502.
[10] A. Anastasijević.(2013).An Indoor Positioning Realisation for GSM using Fingerprinting and kNN. Telfor Journal(1),82-86.