Deep Neural Network for Indoor Positioning Based on Channel Impulse Response

Conference Paper · August 2022

CITATIONS
0

READS
7

2 authors, including:

Van-Lan Dao
Malardalen University
21 PUBLICATIONS **115** CITATIONS

SEE PROFILE

Deep Neural Network for Indoor Positioning Based on Channel Impulse Response

Van-Lan Dao * and Shaik Mohammed Salman †
*Malardalen University,Västerås, Sweden, van.lan.dao@mdu.se
†ABB AB,Västerås, Sweden, shaik.salman@se.abb.com

***Abstract*—Fingerprinting positioning aided by wireless technologies plays an important role in a variety of industrial applications, such as factory automation, warehouse automation, and underground mining, where guaranteeing a position prediction error smaller than a threshold value is necessary to meet certain functional requirements. In this paper, we firstly design a deep convolutional neural network that uses the channel impulse response measurement as an input parameter to predict the position of a mobile robot. Second, we propose a simulated annealing algorithm that finds a minimum number of access points with their respective optimal positions that satisfies an expected average distance error in terms of a mobile robot's predicted position. The obtained results show that the average distance error is significantly reduced, e.g., by half compared to the case without optimal positions of access points.**

***Index Terms*—fingerprinting positioning, convolutional neural network, channel impulse response, simulated annealing**

I. Introduction

Autonomous mobile robots automate several logistics tasks in domains such as warehouses, factories and mines. In warehouses and order fulfillment centers, for example, they can transport a rack of items to a picking station and back to the point of origin [1]. In factories, they can be used to move items between different machines and conveyors [2]. In mines, they can be used for inspection along tunnel routes [3]. In these domains, the problem of assigning robots to different tasks has been investigated to optimize task completion times, shorten distances traveled, reduce waiting times, and generally improve resource utilization [4]. To achieve many of these goals, continuous information about the position of the robots is essential. For example, to minimize the distance traveled in a factory, the task of moving a finished product from a machine to a storage location can be assigned to a robot that is closer to the machine than to one that is away from it. This can only be achieved if the position of the robots is known in real time with reasonable accuracy. Although satellite-based global positioning systems can be used to locate robots outdoors, they are unreliable for indoor environments such as factories, mines, and warehouses. To address this problem, several indoor positioning systems have been proposed that rely on wireless communication technologies such as ultra-wideband and WiFi [5]–[9]. These approaches rely on methods such as lateration, angulation, and fingerprinting to locate an object of interest, such as a robot. In lateration, the distance of the receiver is measured by three different transmitters whose positions are fixed, and the receiver must be within the intersection of the area covered by each transmitter, while angulation measures the angle of arrival. In the fingerprinting approach, information in the form of received signal strength indication (RSSI), or channel impulse response (CIR) is determined and stored in a database. This is compared with the information received at runtime to estimate the position. A variety of techniques/algorithms have been proposed to improve the accuracy of position determination, and machine learning techniques have recently emerged as a possible solution for dealing with unpredictable radio propagation characteristics [10], [11]. However, in the literature, most machine learning techniques based on both supervised and unsupervised learning categories are used to improve the accuracy of the positioning system with a fixed number of wireless reference nodes/access points (APs) (also called anchors) and their fixed positions. Moreover, the positions of APs/anchors play an important role in improving the quality of service of wireless systems in general, as well as in increasing the accuracy of positioning systems [12]–[14]. Therefore, an intensive study of the effects of the number of APs/anchors and their positions on the accuracy of the positioning system is required.

Signatures in terms of RSSI, channel state information (CSI), and CIR have been adopted widely in the literature. However, CIR may provide better system performance in terms of security aspect compared to RSSI [15]. Therefore, in this work, the dataset based on CIR features is selected because it contains the environment information and is unique for each pair of wireless nodes. As a result, the CIR dataset can also be used for physical layer security [16] in future work.

In this paper, we propose to use a deep neural network to predict the position of the robot based on CIR to minimize average distance error (ADE) (See Eq. (2)). We approach this problem by determining the locations for access point placement and the minimum number of access points required to guarantee a threshold ADE. Specifically, we formulate the following research questions (RQs):

- RQ1: How to position APs to minimize ADE?
- RQ2: How to minimize the number of APs deployed while guaranteeing the threshold ADE?

To address the first RQ, we design a deep convolutional neural network (DCNN) using CIRs as input data for both training and test phases. Then, we propose a Simulated Annealing (SA) algorithm to find the optimal positions for a specific number of APs. In order to solve the second RQ, we consider an

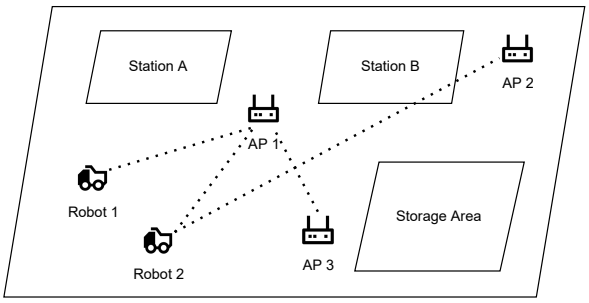

Fig. 1. Simplified layout of a factory.

increase of number of the APs and then repeat the proposed SA algorithm until the ADE requirement is satisfied, i.e. the obtained ADE is smaller or equal to the threshold ADE.

The rest of the paper is organized as follows. Typical use cases are introduced in Section II, followed by related works in Section III. Section IV presents the system model and CIR measurement. Section V describes the designed DCNN for predicting the mobile robot position. Next, Section VI provides the proposed algorithm finding the optimal number of APs and their optimal positions as well. After that, the evaluation of our solution is presented in Section VII. Finally, Section VIII concludes the paper.

II. Positioning Use Cases

We describe scenarios from three different domains, (i) factory automation, (ii) mining, and (iii) warehouse automation to show how positioning information can be used for assigning tasks to robots.

A. Factory Automation

A factory usually has several stations, and each station may consist of one or more machines that perform specific jobs, such as polishing or palletizing objects of interest. Once a job is completed at one machine, the objects must be moved to the next stage. This can be either another machine within the same station, a machine in another station, or, when all jobs are completed, a storage area. In any of these scenarios, mobile robots can be used to move the objects in question between different locations (see Fig.1). An indoor positioning system can be used to actively monitor the position of these robots and assign the task of moving objects between different stages to different robots so that a specific goal, such as a shorter traveled distance or higher throughput, can be achieved. If the ADE is sufficiently low, the position data can be used not only for task assignment but also for motion planning and control of the mobile robots.

B. Mining

Autonomous mobile robots can be used in mining for various purposes such as inspection [3] and material handling [17]. In all the scenarios, the robots operate in underground tunnels requiring indoor positioning systems. Moreover, tracking all the miners are to be a legal requirement by the Mine Improvement and New Emergency Response (MINER) [10] and consequently an indoor positioning system with increased accuracy is essential.

C. Warehouse Automation

In warehouses and order fulfillment centers, mobile robots can be used in many different ways [1]. A mobile robot can move an object from a shelf to a picking station or move an empty shelf to be replenished and then return the replenished shelf to its position. Alternatively, it can move a shelf of objects of interest to a picking station. Since these warehouses are typically large indoor spaces, multiple mobile robots may be required, and the goal may be to improve throughput with a minimal number of robots. Similar to the factory environment, indoor positioning systems can be used to monitor the position of these robots and assign tasks to the robots to achieve the goal of throughput.

III. Related Work

In [18], a ray tracing assisted fingerprinting method is proposed to estimate a position by fusing the predicted and measured signatures in terms of CIRs. By combining the measured and predicted CIR, the position prediction error decreased. In [19], a deep learning approach for WiFi positioning method is presented, where the authors show that a deep learning based model can provide better performance than others such as KNearest- Neighbor (KNN) or Support Vector Machines (SVM). To improve the positioning accuracy of the algorithm, an adaptive K-value WKNN algorithm based on WiFi signals is proposed in [20], [21]. A deep learning method for 5G positioning is presented in [22], which uses both the complex impulse response and channel geometric parameters to improve the localization error. Another approach, namely a superresolution aided fingerprinting algorithm is introduced to decrease the positioning error compared to the conventional method using CSI [23], [24]. Different neural network approaches are proposed for underground mining applications using CIR as the signature dataset in [25]–[27]. To improve the positioning accuracy, the authors in [28] build different datasets based on CIR, RSSI, time of arrival, time difference of arrival, and the ratio of the first path amplitude to the peak amplitude. Another method is also proposed for the ultrawideband systems in [29]. The Kullback-Leibler distance kernel regression is used for position prediction in [30], which significantly improves the average localization accuracy. In [31], an unsupervised learning technique is used for WiFi positioning application based on the collected CSI dataset. In [32], a transfer learning approach is proposed to deal with a small number of samples of the dataset and reduce the positioning error. Another fingerprint feature transfer approach is proposed in [33] to improve localization accuracy and adaption efficiency. In contrast to indoor positioning, outdoor positioning using a multipath fingerprinting dataset is proposed in [34]. However, all the aforementioned works only aim to improve the distance error performance using a predefined number of APs and their positions.

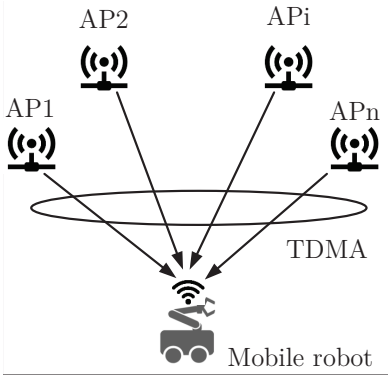

Fig. 2. System model.

IV. System Model and Channel Impulse Response Measurement

We consider n APs communicating with a mobile robot in a room, as in Fig. 2. While the mobile robot moves only on the floor with position (x_m, y_m), all APs are mounted at position (x_i, y_i, z_i). To determine the position of the mobile robot, it communicates with all APs in the downlink or uplink. In the downlink, each AP sends its signal to the mobile robot in different time slots following Time Division Multiple Access (TDMA) to avoid unexpected collisions between APs. In this case, the mobile robot can estimate its own position. In contrast, the mobile robot communicates with all APs in the uplink by broadcasting its signal in one slot. It is assumed that all APs are connected to each other to exchange their data and determine the mobile robot's position for other purposes, such as optimal path planning for the mobile robot. In a scenario with multiple robots, a scheduling algorithm is needed for the wireless transmission for all robots, which is beyond the scope of this work.

To determine the mobile robot position, we measure CIR between each AP and mobile robot and then use DCNN for predicting its position. The CIR model can be expressed as follows [35]:

$$\text{CIR}\,(t) = \sum_{i=1}^{N} h_i \text{sinc}\left[W\left(t - \tau - t_i\right)\right], \tag{1}$$

where W, N, h_i, τ, and t_i are the system bandwidth, the total number of paths, the complex channel gain in accordance to the path-i, the transmission delay, and the delay deviation between the first path and path-i, respectively.

In the literature, to investigate the performance of indoor wireless solutions, channel modeling and measurement methods have been proposed for different types of applications. Ray tracing channel modeling is a suitable tool for both line of sight (LoS) and non line of sight (NLoS) [36]–[38]. Moreover, the ray tracing channel model and CIR measurement are supported by software, e.g., the Matlab library [39]. Therefore, in this work, the ray tracing simulation of the CIR is used to calculate the CIRs between the transmitter and receiver, as shown in Fig. 3. At the transmitter, a wide range configuration

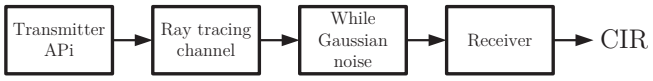

Fig. 3. The processing chain calculating CIR.

parameters are set to generate the waveform for the respective packet according to a particular standard, e.g., IEEE. 802.11az [40]. The 802.11az packet is then transmitted over the ray tracing channel before white Gaussian noise is added. At the receiver, a number of tasks are performed before extracting CIR, such as packet capture, coarse frequency correction, time synchronization, fine frequency correction, demodulation, and channel estimation. In this work, the measured CIR is sampled at a certain frequency and then the magnitude of each multi-path component in the CIR is calculated. Finally, the real CIR values are obtained.

V. Deep Convolutional Neural Network

A DCNN is used for a large number of applications such as object detection, image classification and so on. Due to the fact that the real CIR values are sequences in time, a DCNN is used in this work [41]. Various DCNN configurations have been studied in the literature, such as Alexnet [42], VGG-16 [43], ResNet-50 [44], and Efficient-B0 [45]. We adapt these approaches to construct the designed DCNN. In particular, the designed DCNN is based on K blocks, including convolutional layer (Conv), batch normalization layer (BatchNorm), rectified linear unit (ReLU), pooling layer (Pool), followed by a dropout layer, fully connected layer (FC), and a softmax layer, as shown in Fig. 4. Here, one Conv is responsible for extracting features from the input data with a kernel, while multiple Conv layers can extract high-level features. To speed up the training process, BatchNorm and ReLU are used. Also, a dropout layer is used to solve the overfitting problem. As the data passes through Conv, the size of the features is reduced before redundant information is removed using Pool to reduce computations.

A. Dataset Generation

We first create an indoor environment with a detailed dimensions, as shown in Fig. 5 using the Sketchup software [46]. We assume that three machines M1, M2, and M3 located in the room cooperate with each other following a specific chain of operations. In addition, several mobile robots are responsible for various tasks, such as feeding material to each machine. Each mobile robot moves with the velocity (v_x, v_y). While several APs are mounted at (x_i, y_i, z_i), a wireless device is also mounted on the mobile robot at height h. Here, all the mobile robots are allowed to move in the gray colored area as shown in Fig. 5. The direct link between each AP and the mobile robot exists following a LoS link. We also consider a metallic surface for the propagation ray tracing model.

To generate the DCNN dataset, we collect the measured CIRs for all pairs of APi-mobile robot with different signal-to-noise ratios (SNRs) considering all possible positions of

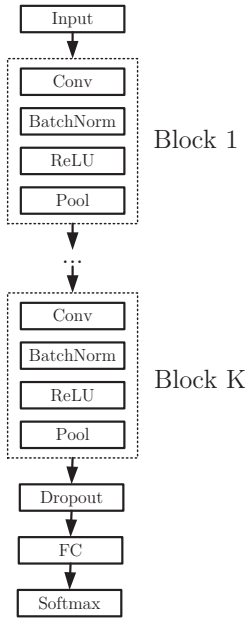

Fig. 4. Deep convolutional neural network architecture.

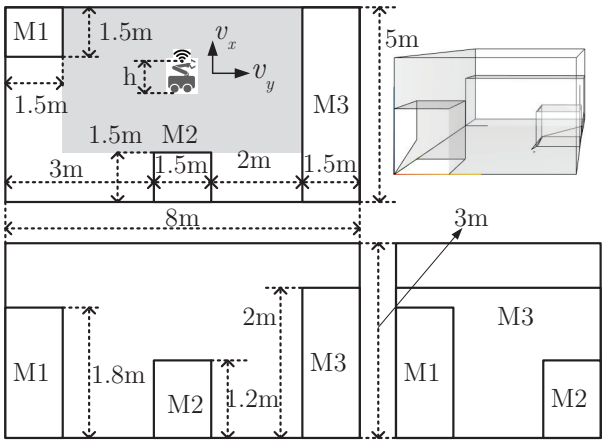

Fig. 5. Indoor environment room.

the mobile robot, as shown in Fig. 5. The distance between two seamless positions is α. The obtained dataset is divided into three subsets: (i) training set with 80%, (ii) validation set with 10%, and (iii) test set with 10%. While the training set and the validation set are used during the training phase to tune the hyperparameters of the model, the test set is used for performance evaluation.

B. Performance Evaluation

To evaluate the performance of the designed DCNN, this work uses ADE between the real mobile robot positions and the predicted positions. Also, the performance metric is evaluated based on the test set. The ADE can be expressed as follows:

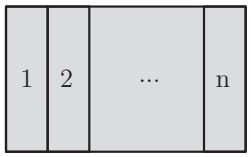

Fig. 6. Partition the gray area into n equal sub-areas.

$$\text{ADE} = \frac{\sum_{j=1}^{M_{test}} \sqrt{(x_{m.j} - \hat{x}_{m.j})^2 + (y_{m.j} - \hat{y}_{m.j})^2}}{M_{test}}, \quad (2)$$

where M_{test}, $(x_{m.j}, y_{m.j})$, and $(\hat{x}_{m.j}, \hat{y}_{m.j})$ are the number of test set samples, real position, and predicted output position of the designed DCNN of the mobile robot, respectively. Accordingly, a decrease of the ADE leads to a better model's performance.

VI. Optimal Access points' Positions

In this section, we propose an algorithm to find a minimum number of APs along with their optimal mounting positions under the given constraint of the performance metric, i.e., ADE $\leq$ ADE $_{threshold}$ using SA. In practice, the height of APs should be fixed and should be accessible for mounting the APs in the considered indoor environment. Therefore, we only try to find the optimal positions for the APs with respect to $(x_{n.i.opt}, y_{n.i.opt})$.

The proposed algorithm is provided in the**Algorithm 1**. The area of interest (the gray area in Fig. 5) is divided into n equal sub-areas in line 6 as shown in Fig. 6 where n is the number of APs. Each AP is located in each sub-area to ensure that the whole area is covered by all APs. Basically, number of APs is increased until the ADE threshold condition is satisfied, i.e. $\text{ADE}_{min} \leq \text{ADE}_{threshold}$ as indicated in lines 25-27. The ADE is evaluated using the output of the designed DCNN which provides the predicted position. Moreover, a new dataset is generated for the DCNN for each n APs and their positions. For a specific number of APs, a SA process is deployed to find ADE_{min} and optimal APs' positions. Here, the SA process includes different parameters such as maximum temperature T, factor ϵ, number of iterations L_n. The SA process is terminated by a number of iterations L_n, which is determined by experiment. To escape the local optimal points, the probability of accepting a non-improving neighbor δ is used in line 17. The convergence of the proposed algorithm and the quality of the achievable solution depend on the number of iterations. If number of iterations is large, the obtained solution is close to the global optimal solution. This leads to increase in the execution time as well. However, the optimal solution is attained by running the proposed algorithm once during offline phase and the optimal number of APs and their optimal positions are determined during this offline phase.

Algorithm 1 Finding minimum number of APs and their optimal positions

Input: $\text{ADE}_{threshold}$
Output: ADE_{min}, number of APs (n), $(x_{n.i.opt}, y_{n.i.opt})$
function main
  Number of APs: $n = n_0$;
  while True **do**
    Partition the gray area into n equal sub-areas;
    Generate initial solution $S_0 = (x_{n.i0}, y_{n.i0})$;
    Calculate the cost function at S_0: ADE_0;
    for i = 1:L_n **do**
      Generate a random neighbor $S^{'} = (x_{n.i}, y_{n.i})$;
      Calculate the cost function at $S^{'}$: ADE;
      Calculate $\Delta = \text{ADE} - \text{ADE}_0$;
      if $\Delta \leq 0$ **then**
        Update $S = S^{'}$;
        Update $\text{ADE}_{min} = \text{ADE}$;
      else
        Calculate $\delta = e^{-\frac{\Delta}{T}}$;
        if $\delta >$ random[0,1) **then**
          Update $S = S^{'}$;
          Update $\text{ADE}_{min} = \text{ADE}$;
        end if
      end if
      Update temperature: $T = \epsilon T$;
    end for
    if $\text{ADE}_{min} \leq \text{ADE}_{threshold}$ **then**
      Update $(x_{n.i.opt}, y_{n.i.opt}) = (x_{n.i}, y_{n.i})$;
      Break;
    else
      $n = n + 1$;
    end if
  end while
  return ADE_{min}, n, $(x_{n.i.opt}, y_{n.i.opt})$;
end function

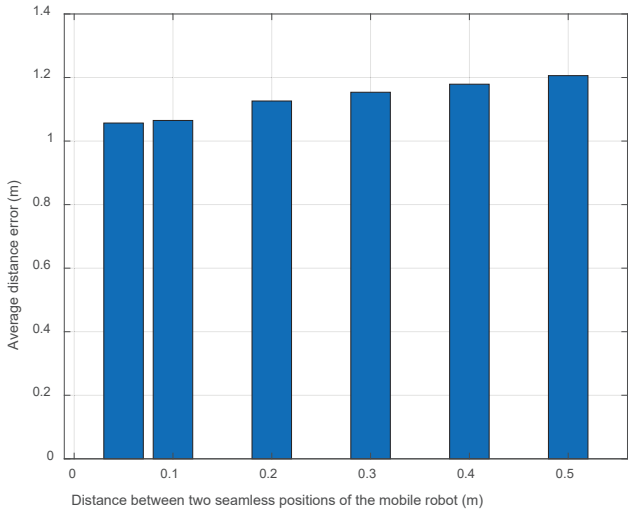

Fig. 7. Average distance error versus distance between two seamless positions of the mobile robot.

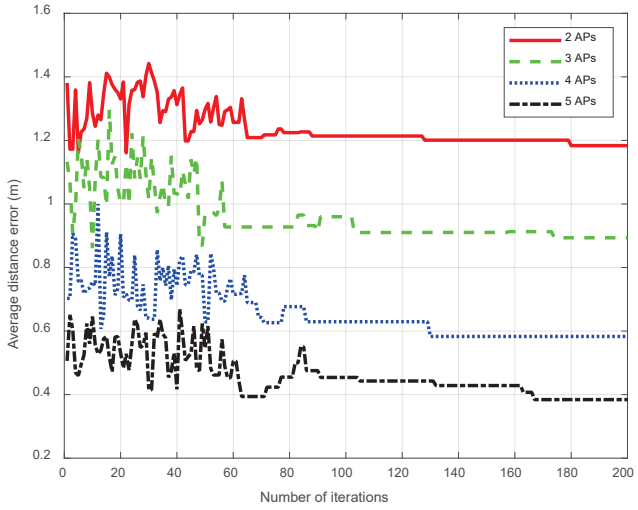

Fig. 8. Average distance error versus number of iterations using the proposed SA algorithm.

VII. Numerical Result

The *MATLAB*, *WLAN Toolbox*, *Communications Toolbox* and *Deep Learning Toolbox* are used to generate the dataset of CIRs with IEEE 802.11az standard, implement the designed DCNN and evaluate the performance of the considered system [39]. The dataset is generated with different values of SNRs, e.g. SNR = 20, 25, 30 dB. All simulations and including the evaluation of the proposed algorithm have been carried out on a 64-bit Windows 10 Pro machine with Intel Core I7-10700KF CPU @ 3.8 GHz, 16 GB memory and GPU NVIDIA GeForce RTX 3080. For the evaluation, we consider the following system parameters: $h = 1$ m, four transmit antennas at the transmitter, four receive antennas at the receiver, system bandwidth $W = 160$ MHz, mobile robot speed $(v_x, v_y) = (1, 1)$ m/s. For the designed DCNN, seven blocks $K = 7$ are used in which each convolutional layer is configured with 256 filters of size [3 3] and a stride of [1 1] and average pooling with pool size [2 2] and stride [2 2]; dropout layer with dropout probability 0.5; two neurons predicting the position of the mobile robot in terms of (x_m, y_m). In addition, all APs are placed at a height of 2.9 m.

Resolution: Fig. 7 shows how the distance between two seamless positions of the mobile robot α affects the ADE. Here, the number of APs is four and they are located in the center of each sub-area, as discussed in section VI. From the figure, it can be seen that when α decreases from 0.5 m to 0.05 m, the ADE also decreases from 1.206 m to 1.057 m. This means that choosing an appropriate α value contributes to the prediction accuracy of the mobile robot's position. However, when α is a small value, the dataset generation and the training phase take a longer time and more computer resources such as memory are required. Considering the available computer resources, we choose the value $\alpha = 0.1$ for further evaluations.

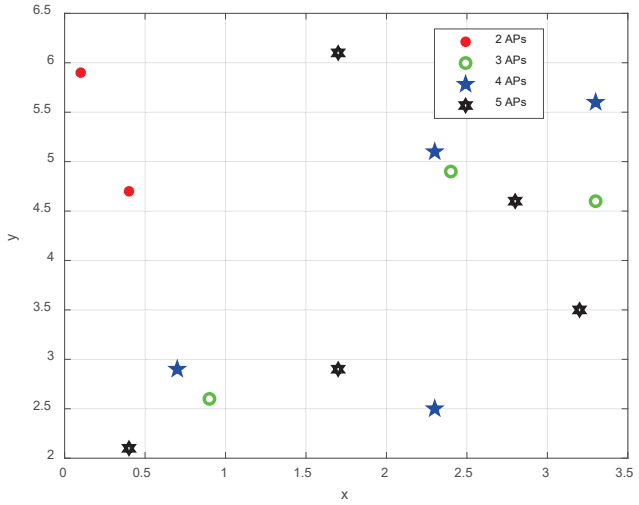

Fig. 9. Optimal positions of the APs using the proposed SA algorithm.

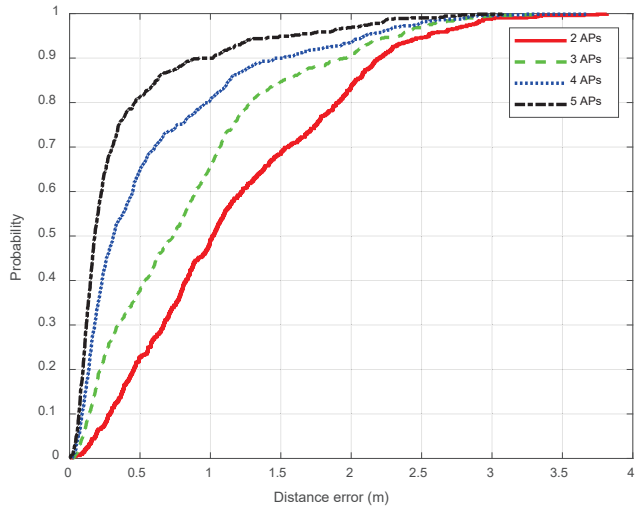

Fig. 10. The cumulative distribution function of predicted position error.

Number of Iterations: In Fig. 8, the change in the ADE by number of iterations for the proposed algorithm with different number of APs are shown. We can see that in most cases, the ADEs converge to their respective values after 180 iterations. From the figure, we can see that the ADE improves significantly with an increase in the number of APs. This is because all possible positions of the mobile robot are covered by all APs. We can also see the optimal positions of the APs in Fig. 9. Interestingly, the optimal ADE in Fig. 8 with the optimal positions of the four APs in Fig. 9 is almost half, i.e., 0.583 m compared to 1.065 m in Fig. 7 (without optimal APs' positions) with $\alpha = 0.1$.

Distance Error Probability: In Fig. 10, the cumulative distribution of predicted position error is shown with the optimal positions of APs. Again, the probability of achieving a lower distance error is increased with increase in the number of APs.

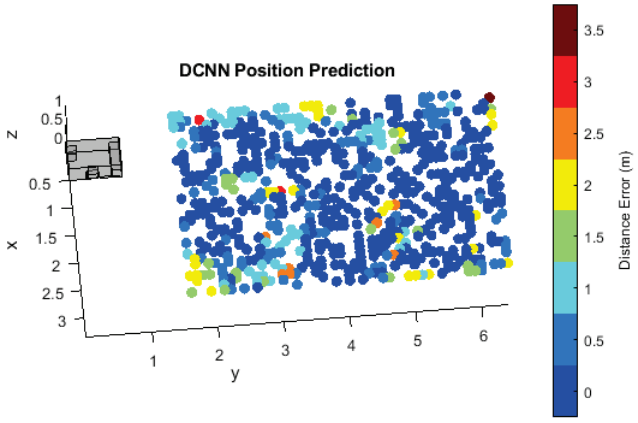

Fig. 11. The mobile robot position prediction with optimal positions of the four APs.

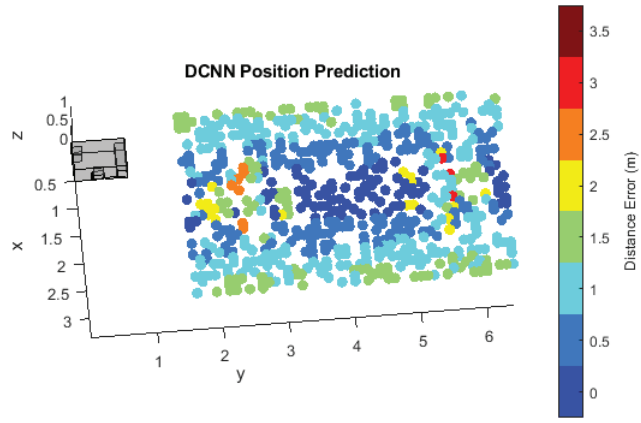

Fig. 12. The mobile robot position prediction with the four APs located at middle location of their areas.

Baseline Comparison: To illustrate what the distance error looks like in the prediction phase, we present two figures with and without optimal positions in the case of four APs, Fig.11 and Fig.12. In Fig. 11, we see a large number of points with small distance error, while the number of points with larger distance error, e.g., 1.5 m, is much larger at the edge of Fig.12 than in the Fig.11.

VIII. Conclusion

In this paper, we design a DCNN for predicting the position of a mobile robot in an indoor environment based on fingerprinting and CIR feature extraction. We also propose a SA algorithm to find a minimum number of necessary APs and their optimal positions under the condition of a threshold ADE. The obtained results show the effectiveness of the proposed algorithm and the proposed DCNN, i.e., the optimal ADE significantly decreases to half compared to the case without optimal APs' positions. In the future, we would like to evaluate the performance of the proposed algorithm and the proposed DCNN in practice considering multiple robots scenario and the interference due to the movement of them as well. Moreover, optimal hyperparameters for the designed DCNN should be

studied under different constraints, e.g. reduce the prediction time.

ACKNOWLEDGMENT

The research leading to these results has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 764785, FORA—Fog Computing for Robotics and Industrial Automation. The authors would like to thank Thang Nguyen, Manomotion AB, for his valuable discussion.

REFERENCES

[1] N. Boysen, R. de Koster, and F. Weidinger, "Warehousing in the e-commerce era: A survey," *European Journal of Operational Research*, vol. 277, no. 2, pp. 396–411, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0377221718307185

[2] A. F. Melo and L. M. Corneal, "Case study: evaluation of the automation of material handling with mobile robots," *International Journal of Quality Innovation*, vol. 6, no. 1, 2020.

[3] R. R. Murphy, J. Kravitz, S. L. Stover, and R. Shoureshi, "Mobile robots in mine rescue and recovery," *IEEE Robotics Automation Magazine*, vol. 16, no. 2, pp. 91–103, 2009.

[4] A. Bolu and O. Korcak, "Adaptive Task Planning for Multi-Robot Smart Warehouse," *IEEE Access*, vol. 9, pp. 27 346–27 358, 2021.

[5] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, *A Review of Indoor Localization Techniques and Wireless Technologies*. Springer US, 2021, vol. 119, no. 1. [Online]. Available: https://doi.org/10.1007/s11277-021-08209-5

[6] P. S. Farahsari, A. Farahzadi, J. Rezazadeh, and A. Bagheri, "A Survey on Indoor Positioning Systems for IoT-based Applications," *IEEE Internet of Things Journal*, vol. 00, no. 00, pp. 1–21, 2022.

[7] I. Silva, C. Pendo, and A. Moreira, "Real-world deployment of low-cost indoor positioning systems for industrial applications," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5386–5397, 2022.

[8] N. Chukhno, S. Trilles, J. Torres-Sospedra, A. Iera, and G. Araniti, "D2D-based cooperative positioning paradigm for future wireless systems: A survey," *IEEE Sensors Journal*, vol. 22, no. 6, pp. 5101–5112, 2022.

[9] Y. Xianjia, L. Qingqing, J. P. Queralta, J. Heikkonen, and T. Westerlund, "Applications of UWB networks and positioning to autonomous robots and industrial systems," in *Mediterranean Conference on Embedded Computing (MECO)*, Budva, Montenegro, 2021, pp. 1–6.

[10] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214 945–214 965, 2020.

[11] A. Al-Habashna, G. Wainer, and M. Aloqaily, "Machine learning-based indoor localization and occupancy estimation using 5G ultra-dense networks," *Simulation Modelling Practice and Theory*, vol. 118, p. 102543, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569190X22000430

[12] V. Gadiraju, H.-C. Wu, C. Busch, P. Neupane, S. Y. Chang, and S. C.-H. Huang, "Novel sensor/access-point coverage-area maximization for arbitrary indoor polygonal geometries," *IEEE Wireless Communications Letters*, vol. 10, no. 12, pp. 2767–2771, 2021.

[13] X. Li, B. Huang, Y. Tian, Z. Xu, and X. Li, "Optimal wifi aps deployment for localization and coverage based on virtual force," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Nanjing, China, 2021, pp. 1–6.

[14] M. A. Dastgheib, H. Beyranvand, and J. A. Salehi, "Optimal placement of access points in cellular visible light communication networks: An adaptive gradient projection method," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6813–6825, 2020.

[15] F. J. Liu, X. Wang, and S. L. Primak, "A two dimensional quantization algorithm for cir-based physical layer authentication," in *IEEE International Conference on Communications (ICC)*, Budapest, Hungary, 2013, pp. 4724–4728.

[16] J. Liu and X. Wang, "Physical layer authentication enhancement using two-dimensional channel quantization," *IEEE Transactions on Wireless Communications*, vol. 15, no. 6, pp. 4171–4182, 2016.

[17] M. Wahde, M. Bellone, and S. Torabi, "A method for real-time dynamic fleet mission planning for autonomous mining," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 5, pp. 564–590, 2019.

[18] P.-H. Tseng, Y.-C. Chan, Y.-J. Lin, D.-B. Lin, N. Wu, and T.-M. Wang, "Ray-tracing-assisted fingerprinting based on channel impulse response measurement for indoor positioning," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 5, pp. 1032–1045, 2017.

[19] S. BelMannoubi and H. Touati, "Deep neural networks for indoor localization using wifi fingerprints," in *International Conference on Mobile, Secure, and Programmable Networking*. Springer, 2019, pp. 247–258.

[20] J. Zhang and H. Mao, "Wknn indoor positioning method based on spatial feature partition and basketball motion capture," *Alexandria Engineering Journal*, vol. 61, no. 1, pp. 125–134, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1110016821003203

[21] H. Ye and J. Peng, "Robot indoor positioning and navigation based on improved wifi location fingerprint positioning algorithm," *Wireless Communications and Mobile Computing*, vol. 2022, 2022.

[22] N. Lv, F. Wen, Y. Chen, and Z. Wang, "A deep learning-based end-to-end algorithm for 5g positioning," *IEEE Sensors Letters*, 2022.

[23] Y.-J. Lin, P.-H. Tseng, Y.-C. Chan, and G.-S. Wu, "Super-resolution-aided positioning fingerprinting based on channel impulse response measurement," in *IEEE Wireless Communications and Networking Conference (WCNC)*. San Francisco, CA, USA: IEEE, 2017, pp. 1–6.

[24] Y.-J. Lin, P.-H. Tseng, Y.-C. Chan, J. He, and G.-S. Wu, "A super-resolution-assisted fingerprinting method based on channel impulse response measurement for indoor positioning," *IEEE Transactions on Mobile Computing*, vol. 18, no. 12, pp. 2740–2753, 2018.

[25] S. Dayekh, S. Affes, N. Kandil, and C. Nerguizian, "Cooperative geolocation in underground mines: A novel fingerprint positioning technique exploiting spatio-temporal diversity," in *IEEE international symposium on personal, indoor and mobile radio communications*. Toronto, Canada: IEEE, 2011, pp. 1319–1324.

[26] ——, "Cooperative localization in mines using fingerprinting and neural networks," in *IEEE Wireless Communication and Networking Conference*. Sydney, NSW, Australia: IEEE, 2010, pp. 1–6.

[27] ——, "Cost-effective localization in underground mines using new SIMO/MIMO-like fingerprints and artificial neural networks," in *IEEE International Conference on Communications Workshops (ICC)*. Sydney, NSW, Australia: IEEE, 2014, pp. 730–735.

[28] Y.-M. Lu, J.-P. Sheu, and Y.-C. Kuo, "Deep learning for ultra-wideband indoor positioning," in *IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Helsinki, Finland: IEEE, 2021, pp. 1260–1266.

[29] W. Q. Malik and B. Allen, "Wireless sensor positioning with ultrawideband fingerprinting," in *First European Conference on Antennas and Propagation*. Nice, France: IEEE, 2006, pp. 1–5.

[30] L. Chen, I. Ahriz, D. Le Ruyet, and H. Sun, "Probabilistic indoor position determination via channel impulse response," in *IEEE Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Bologna, Italy: IEEE, 2018, pp. 829–834.

[31] J. Choi, "Sensor-aided learning for Wi-Fi positioning with beacon channel state information," *IEEE Transactions on Wireless Communications*, 2022.

[32] R. Klus, L. Klus, J. Talvitie, J. Pihlajasalo, J. Torres-Sospedra, and M. Valkama, "Transfer learning for convolutional indoor positioning systems," in *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Lloret de Mar, Spain: IEEE, 2022, pp. 1–8.

[33] G. Wang, A. Abbasi, and H. Liu, "Wifi-based environment adaptive positioning with transferable fingerprint features," in *IEEE Annual Computing and Communication Workshop and Conference (CCWC)*. NV, USA: IEEE, 2021, pp. 0123–0128.

[34] J. Tan and H. Zhao, "UAV localization with multipath fingerprints and machine learning in urban NLOS scenario," in *IEEE International Conference on Computer and Communications (ICCC)*. Chengdu, China: IEEE, 2020, pp. 1494–1499.

[35] P.-H. Tseng, Y.-C. Chan, Y.-J. Lin, D.-B. Lin, N. Wu, and T.-M. Wang, "Ray-tracing-assisted fingerprinting based on channel impulse response measurement for indoor positioning," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 5, pp. 1032–1045, 2017.

[36] K. Preusser and A. Schmeink, "Robust channel modeling of 2.4 ghz and 5 ghz indoor measurements: Empirical, ray tracing, and artificial neural network models," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 1, pp. 559–572, 2022.

[37] H. B. Eldeeb, M. Elamassie, S. M. Sait, and M. Uysal, "Infrastructure-to-vehicle visible light communications: Channel modelling and performance analysis," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2022.

[38] F. Fuschini, M. Barbiroli, G. Bellanca, G. Cal, J. Nanni, and V. Petruzzelli, "A ray tracing tool for propagation modeling in layered media: A case study at the chip scale," *IEEE Open Journal of Antennas and Propagation*, vol. 3, pp. 249–262, 2022.

[39] The MathWorks, Inc., "Matlab," https://se.mathworks.com/, accessed: 2022-02-15.

[40] *IEEE Draft Standard for Information technology - Telecommunications and information exchange between systems Local and metropolitan area networks - Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 3: Enhancements for positioning*, IEEE Std., 2021.

[41] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, http://www.deeplearningbook.org.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., vol. 25. Curran Associates, Inc., 2012.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*. IEEE Computer Society, 2016, pp. 770–778. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[45] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 6105–6114. [Online]. Available: https://proceedings.mlr.press/v97/tan19a.html

[46] Trimble, Inc., "Sketchup," https://www.sketchup.com/, accessed: 2022-02-15.