



HAL
open science

Convolutional Neural Networks based Denoising for Indoor Localization

Wafa Njima, Marwa Chafii, Ahmad Nimr, Gerhard Fettweis

► **To cite this version:**

Wafa Njima, Marwa Chafii, Ahmad Nimr, Gerhard Fettweis. Convolutional Neural Networks based Denoising for Indoor Localization. The 2021 IEEE 93rd Vehicular Technology Conference: VTC2021-Spring, Apr 2021, Helsinki (on line), Finland. hal-03170579

HAL Id: hal-03170579

<https://hal.science/hal-03170579v1>

Submitted on 16 Mar 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Convolutional Neural Networks based Denoising for Indoor Localization

Wafa Njima^{*}, Marwa Chafii^{*}, Ahmad Nimr[†], Gerhard Fettweis[†]

^{*}ETIS, UMR8051, CY Cergy Paris Université, ENSEA, CNRS, France

[†]Vodafone Chair Mobile Communication Systems, Technische Universität Dresden, Germany

Email: {wafa.njima, marwa.chafii}@ensea.fr, {ahmad.nimr, gerhard.fettweis}@ifn.et.tu-dresden.de

Abstract—Indoor localization can be based on a matrix of pairwise distances between nodes to localize and reference nodes. This matrix is usually not complete, and its completion is subject to distance estimation errors as well as to the noise resulting from received signal strength indicator measurements. In this paper, we propose to use convolutional neural networks in order to denoise the completed matrix. A trilateration process is then applied on the recovered euclidean distance matrix (EDM) to locate an unknown node. This proposed approach is tested on a simulated environment, using a real propagation model based on measurements, and compared with the classical matrix completion approach, based on the adaptive moment estimation method, combined with trilateration. The simulation results show that our system outperforms the classical schemes in terms of EDM recovery and localization accuracy.

Index Terms—Convolutional Neural Networks (CNN); Indoor localization; Matrix completion; Received Signal Strength Indicator (RSSI); Trilateration.

I. INTRODUCTION

Technological advances provide several location based services (LBS) applications. More recently and with the appearance of connected objects, the location information has aroused more interest in order to allow optimal use of the environmental measurements, reported by various deployed objects. Therefore, localization techniques will have an important role in such applications and can be integrated into the infrastructure or the object itself [1].

The common approaches of obtaining localization information are based on global navigation satellite systems (GNSSs), especially the global positioning system (GPS). However, these solutions are not appropriate for indoor environments, where obstacles such as walls and ceilings block the satellites signals. Alternatively, the propagation parameters associated with wireless communication systems can be exploited. Namely, the angle of arrival (AoA) [2], time of arrival (ToA) [3], time difference of arrival (TDoA) [4], channel state information (CSI) [5] and received signal strength indicator (RSSI) [6]. Due to its simplicity, RSSI is the most commonly used parameter in indoor localization, as it does not require additional hardware for time or phase synchronization and can be easily acquired. The existing methods based on RSSI can be classified into two

categories; solutions based on fingerprinting [7], [8] and others based on trilateration [9].

In trilateration, a node can localize itself relying on the signals detected from three reference nodes (RNs) at least. The measured RSSI value are converted to pairwise distances using the propagation models between the nodes. For multiple-node localization, the Euclidean distance matrix (EDM) is constructed from the pairwise distances between the nodes to be localized and the RNs placed at known positions. However, because of the limitation of signal range and power outage of sensor nodes, only a small number of pairwise distances are available. Thus, there might not be enough distance information to accurately localize all the unknown nodes. To overcome this problem, several matrix completion approaches have been proposed [10] [11] in order to complete the missing data. In our previous work [12], a matrix completion process is formulated as an optimization problem, which is solved by gradient descent and its variants. On the other hand, the accuracy of the RSSI measurements are influenced by the wireless channel conditions such as multi-path reflections, pathloss and fading. As a result, the known parameters of the EDM are noisy, and this noise will be enhanced during the completion process leading to reduced localization accuracy. To overcome this problem, we propose performing denoising on the completed EDM. Thus, a module ensuring correction of the completed matrix, to be used for localization, can be added after the traditional matrix completion module. Once the EDM is completed and corrected, localization takes place.

In this paper, convolutional neural networks (CNN) based denoising approach is proposed. Thus, we apply CNN on pairwise distance matrices determined from multiple RSSI measurements in order to minimize both the measurements noise and the estimation error of noisy completed measurements for indoor localization tasks. CNNs are a class of neural networks (NNs) that reduce the complexity of traditional networks and the number of parameters to learn by its weight sharing structure. It also brings better generalization and robustness. Due to the translational invariance of CNN, it can be used to model temporal correlations which coincide with the temporal dependency of RSSI fingerprints and consequently the

derived pairwise distances. An extension of this work has been published in [13] using advanced completion schemes.

The remainder of this paper is organized as follows: In Section II, the algorithm is presented and each step of the developed localization framework is explained. Section III-A describes the background of CNN including main concepts and different aspects. Simulation results are presented and discussed in Section IV. Finally, the conclusion is presented in Section V.

II. SYSTEM MODEL

We consider a set of sensor nodes including N unknown nodes placed at unknown positions and M RNs with known positions, as illustrated in Fig. 1. Each sensor node collects T successive RSSI measurements received from all other sensors (RN and the other unknown nodes) in the considered area. Collecting multiple RSSI measurements at each position helps to minimize temporal RSSI fluctuations caused by the shadowing and fading effects. All these measurements are sent to a central unit that ensures the preprocessing of data. Then, the central unit localizes a cloud of unknown nodes and send them their predicted coordinates periodically or only when requested.

A. Data preprocessing

The central unit arranges the measured RSSI measurements in form of T RSSI matrices, each of size $(N + M) \times (N + M)$. The pairwise distances between each pair of sensor nodes are then computed using a pathloss propagation model. Afterwards, the EDMs containing the available observed distance information $\mathbf{D}^{(t)} \in \mathbb{R}^{+(N+M) \times (N+M)}$, $t = 0, \dots, T - 1$ are built. As shown in Fig. 1, the entries of $\mathbf{D}^{(t)}$ are organized in the form

$$\mathbf{D}^{(t)} = \begin{bmatrix} \mathbf{D}_{11}^{(t)} & \mathbf{D}_{12}^{(t)} \\ \mathbf{D}_{21}^{(t)} & \mathbf{D}_{22}^{(t)} \end{bmatrix}, \quad (1)$$

where $\mathbf{D}_{12}^{(t)} = \mathbf{D}_{21}^{(t)T} \in \mathbb{R}^{M \times N}$, $\mathbf{D}_{11}^{(t)} \in \mathbb{R}^{N \times N}$, and $\mathbf{D}_{22}^{(t)} \in \mathbb{R}^{M \times M}$ represents the distances between the unknown nodes and RNs, the distances between unknown nodes, and the distances between the RNs, respectively. Note that, $\mathbf{D}_{22}^{(t)}$ is exactly known, whereas $\mathbf{D}_{12}^{(t)}$ and $\mathbf{D}_{11}^{(t)}$ are computed based on the RSSI values, and thus, they include noisy and missing entries. Note that, using the information in $\mathbf{D}_{11}^{(t)}$ brings more distance information than relying only on $\mathbf{D}_{12}^{(t)}$. Once the whole matrix is completed, we use only the sub matrices $\{\mathbf{D}_{12}^{(t)}\}$ for denoising and to perform localization. This is because trilateration requires only the pairwise distances between the unknown node and the RNs as well as the coordinates of the RNs.

B. EDM completion

This step targets completing the missing elements of $\mathbf{D}^{(t)}$ from a small number of known and noisy entries. This problem is formulated as an optimization problem solved by the adaptative moment estimation (ADAM) [14] advanced method. More details on the algorithms are presented in our previous work [12]. However, even if recovered, the EDM is still damaged by measurements noise and prediction errors. Hence, the idea in this paper is to integrate a denoising process to minimize the effect of noise on the estimated EDM in order to improve the localization accuracy. The denoising approach is discussed in Section III.

C. Target Localization based on Trilateration

The trilateration technique [9] is based on a combination of pairwise distances between the unknown nodes and the RNs. To perform trilateration, at least three RNs have to be detected by the target. The distance information are represented in the matrix $\hat{\mathbf{D}}_{12}$, which is obtained after the denoising. Let

$$\hat{d}_{m,n}^2 = \hat{\mathbf{D}}_{12}[n, m] = (x_m - \hat{x}_n)^2 + (y_m - \hat{y}_n)^2 \quad (2)$$

$$p_m^2 = x_m^2 + y_m^2,$$

where (\hat{x}_n, \hat{y}_n) , $n = 1 \dots N$ and (x_m, y_m) , $m = 1 \dots M$ are the estimated coordinates of the unknown node and the RNs, respectively. The coordinates of the n -th unknown node, $\hat{\mathbf{c}}_n = [\hat{x}_n, \hat{y}_n]^T$, are estimated using

$$\hat{\mathbf{c}}_n = (\mathbf{E}^T \mathbf{E})^{-1} \mathbf{E}^T \mathbf{a}_n. \quad (3)$$

Here, $\mathbf{a}_n \in \mathbb{R}^{M-1 \times 1}$, $\mathbf{E} \in \mathbb{R}^{M-1 \times 2}$ defined as

$$\mathbf{a}_n[m] = \frac{1}{2} \left(p_{m+1}^2 - p_1^2 - \hat{d}_{m+1,n}^2 + \hat{d}_{1,n}^2 \right) \quad (4)$$

$$\mathbf{E}[m, :] = [x_{m+1} - x_1 \quad y_{m+1} - y_1].$$

III. PROPOSED DENOISING PROCESS AND ESTIMATION ERROR MINIMIZATION USING CNN

In this section, we aim to exploit CNN tool to reduce the effect of noise on the distance parameters of $\{\mathbf{D}_{12}^{(t)}\}$, which are used in the trilateration step. First, given a set of completed distance matrices considered for training, features are extracted to build a model that predicts the corresponding real distance based on all the others. Once trained, we verify if the model works well or not using validation data. Thus, we use the trained model to predict the real distance and we compare the prediction to the real distance.

The matrices $\{\mathbf{D}_{12}^{(t)}\}$ are reorganized in a matrix \mathbf{D} of size $T \times MN$, such that

$$\mathbf{D}[t, :] = \text{vec} \left\{ \mathbf{D}_{12}^{(t)} \right\}, \quad (5)$$

where $\text{vec} \{\cdot\}$ is the vectorization operation, and $\mathbf{D}[t, :]$ is the t -th row of \mathbf{D} . This matrix is forwarded to the CNN column-wise, as depicted in Fig. 2. Note that each

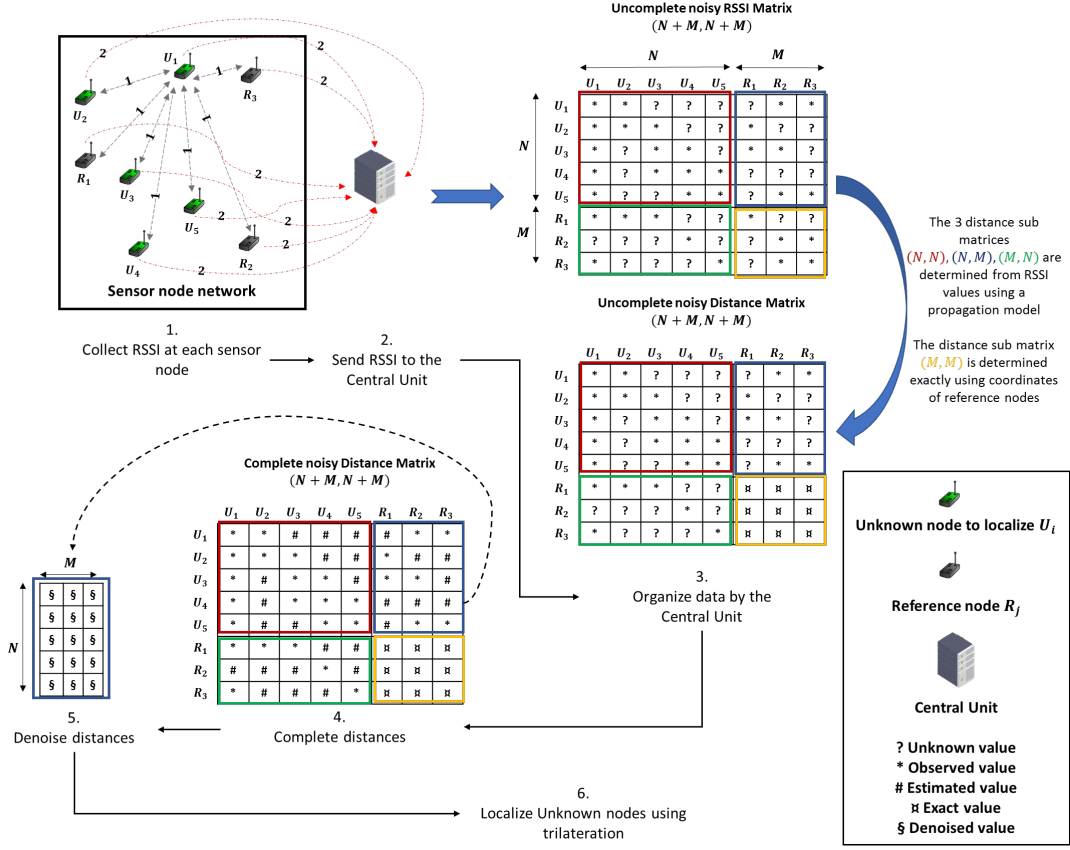


Fig. 1: System model.

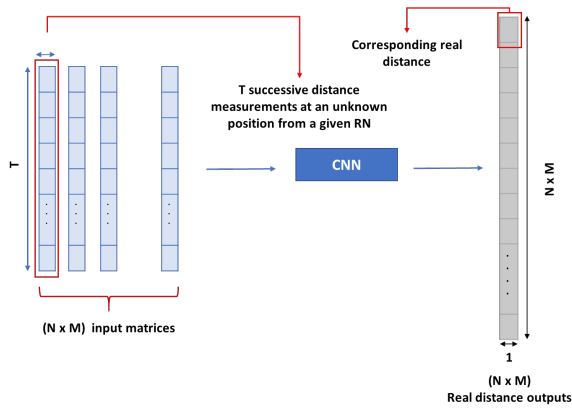


Fig. 2: Structure of CNN at the training phase for noise minimization on distance information.

column of D represents T successive distances between a given unknown node and a given RN.

Then, we train the CNN model in order to construct the best architecture which allows a good denoised distance prediction which is very close to the expected real distance. Once trained, we give as input the completed distances corresponding to a node to localize and we receive as output the estimated corresponding real

distance.

A. Deep CNN architecture overview

The structure of CNN designed for minimization of error estimation on noisy distance measurements consists of specialized layers ensuring different functions. It consists of convolutional layers followed by one or more fully-connected layers. The detailed role of each layer of the denoising module is explained in Fig. 3. There are two phases in the proposed CNN model: offline training, in which the CNN model compares the prediction and the true real distance, and online error minimization. Distance matrices are constructed at

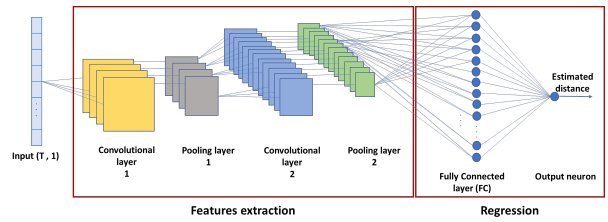


Fig. 3: CNN architecture.

every unknown node presenting the inputs of our CNN network. To which, we apply filters initialized randomly

of given dimension to extract their characteristics. The convolution operation consists of sliding filters across the input. Each convolutional output will be sent to the rectified linear unit (ReLU) activation function which replaces negative values with 0. It is given by

$$g_{\text{ReLU}} = \begin{cases} v & \text{if } v > 0 \\ 0 & \text{otherwise} \end{cases}, \quad (6)$$

where v denotes the outputs of the convolutional layer. After the convolution, comes the pooling layer which allows a spatial reduction which sub-samples the outputs of the previous convolutional layer to reduce the size of the data and lighten the calculation. A common pooling operation is performed dividing the input of the layer into regions called windows and computing the average (average pooling) or the maximum (max pooling) of each region values. When we work with small matrices and want to learn all the features, the pooling layer can be eliminated. After a feature extraction module which can contain several layers of convolution and pooling, comes the fully connected layers which are responsible for determining the predicted denoised corresponding real distance.

IV. SIMULATION RESULTS AND DISCUSSIONS

In this section, we evaluate our system and we present different obtained simulation results.

A. Evaluation metrics

The performance of our developed system is evaluated in terms of distance matrix reconstruction error and localization accuracy. Thus, we define the following two metrics.

- Distances recovery: it measures the error in estimating the distance, and it is given by

$$\text{MSE}_{\text{dis}} = \frac{\text{E} \left[\|\mathbf{D}_{12} - \hat{\mathbf{D}}_{12}\|_F^2 \right]}{NM}, \quad (7)$$

where \mathbf{D}_{12} and $\hat{\mathbf{D}}_{12}$ denote the exact and estimated pairwise distances between the unknown nodes and the RNs, respectively, and $\|\cdot\|_F$ refers to the Frobenius norm.

- Localization accuracy: it measures the average error in the coordinate estimation such that

$$\text{MSE}_{\text{loc}} = \frac{1}{N} \sum_{n=1}^N \text{E} \left[\|\mathbf{c}_n - \hat{\mathbf{c}}_n\|_2^2 \right], \quad (8)$$

where \mathbf{c}_n and $\hat{\mathbf{c}}_n$ are the exact and estimate coordinates at n -th node.

B. Environmental setup and hyper parameters settings

We consider a set of sensor nodes placed randomly in the studied environment of 400 m². These nodes include $M = 10$ RNs and N nodes placed at unknown positions as illustrated in Fig. 4. Two scenarios are evaluated, in the first $N = 25$ and $N = 50$ in the second scenario. The number of measurements used for

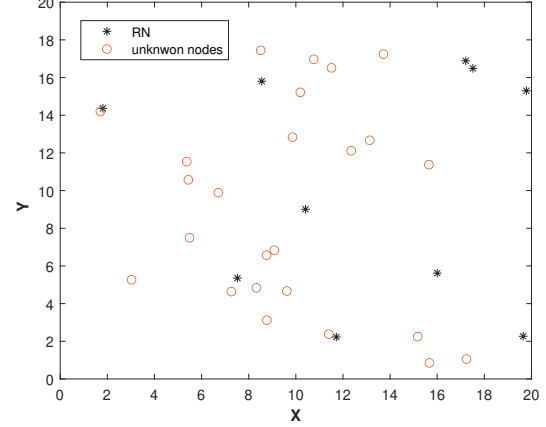


Fig. 4: Configuration of the wireless sensor network. Here, the first scenario, $N = 25$, is shown.

localization is set to $T = 10$ such that, T RSSI values are collected at each position (reference positions and unknown positions). Let $P_{r_{ij}}$, $i \neq j$ be the RSSI value corresponding to the signal received from node j at node i , with $i, j \in \{1, \dots, M + N\}$, which is given by

$$P_{r_{ij}} = P_t - P_{L_{ij}} + B_\sigma \quad [\text{dBm}]. \quad (9)$$

Here, B_σ is a Gaussian random variable that represents the shadowing effects, P_t is the transmitted power by a sensor node and $P_{L_{ij}}$ is the pathloss measurement calculated using the following equation:

$$P_{L_{ij}} = P_{L_0} + 20 \log_{10}(f) + 10\mu \log_{10}\left(\frac{d_{ij}}{d_0}\right), \quad (10)$$

where P_{L_0} is the pathloss value at a reference distance d_0 , f is the frequency, μ is the pathloss exponent and d_{ij} is the distance between sensor node i and sensor node j . In this paper, we use realistic propagation parameters from measurements conducted in our laboratory, which have been verified and validated; $P_t = 20$ dBm, $d_0 = 1$ m, $f = 2.4$ GHz and $\mu = 3.23$. The results are obtained when working in a noisy environment considering $\text{E}[B_\sigma]^2 = 2$.

After completing the distance matrix, we organize the set of CNN inputs: 80% from data for training and the remaining 20% for validation. Our experiments were conducted on a PC with Intel(R) Core(TM) i5-2520M CPU @2.5 GHz. MATLAB R2019b has an advanced Neural Network Toolbox. It is a very efficient framework for us to implement our CNN models.

The optimization of CNN parameters and the choice of the best trained architecture are the most important steps in CNN’s construction and deployment trying to satisfy a good trade-off between the obtained localization accuracy and the online computational complexity. This choice is based on an empirical process requiring several experiments. For both studied scenarios, we have tried different numbers of convolutional and fully connected layers in order to select the best architectures. In our case, we eliminate the pooling layers because we work with small size matrices and we want to save the whole distance information. This choice was verified by simulations. The CNN architecture chosen for both scenarios are summarized in Table I. The terms $\text{conv}(p, q)$ denotes a convolutional layer presenting p filters of size (q, q) and $\text{FC}(z)$ referees to a fully-connected layer with z neurons. We consider the whole training data for each optimization step. We use 0.005 as learning rate and 250 epochs during the training process.

TABLE I: Details of the trained networks.

CNN architecture for the first scenario ($N = 25$)
$\text{conv}(30, 3), \text{conv}(10, 2), \text{conv}(20, 2), \text{FC}(20)$
CNN architecture for the second scenario ($N = 50$)
$\text{conv}(50, 2), \text{conv}(10, 2), \text{FC}(20)$

C. Distances recovery performance

As first step, we begin by investigating the performance in terms of distance matrix recovery when only using a completion matrix and then when adding the proposed denoising process to completed distances. The different results are shown in Table II. It can be noticed that the best reconstruction accuracy is obtained when correcting distances is based on a CNN model. Instead of having 14.59 m or 10.09 m as matrix recovery error, the deployment of CNN improves the precision of distances prediction by 2.92 m and 1.84 m for the first and second scenario, respectively. Such results demonstrate and prove the benefits of the integration of such denoising process in order to accurately estimate distances. As an indication of the complexity, the CNN-based denoising step requires only 1.67×10^{-4} of the whole localization process considering a cloud of 25 unknown sensor nodes.

TABLE II: RMSE Distance recovery [meters].

	First scenario	Second scenario
Without denoising	14.592	10.095
With denoising	11.678	8.256

D. Localization accuracy

In this part, we evaluate our proposed localization scheme (denoted as method 3) and compare it with the

classical trilateration (denoted as method 1) as well as with the classic approach of matrix completion based localization (denoted as method 2). Thus, we compare the localization accuracy of three possible methods, namely

1. Classic trilateration using only the available observed distances between the unknown node and the detected RNs without any distance completion or denoising .
2. Trilateration after matrix completion only, i.e. without denoising.
3. Trilateration after matrix completion and CNN denoising process, which is the subject of this work.

A summary of the performance is given in Table III for both scenarios. It is clear that relying on available measurements only achieves the worse localization accuracy compared to the other two methods that involve completion. In this case, the mean error is about 4.7 m for both scenarios. When using the matrix completion, the localization error is reduced by 25%, with 1.22 m for the first and 1.11 m for the second scenario. Moreover, the denoising with CNN correction process further reduces this error to 0.5 m, which is about 10% of the error of the first method. This validates experimentally the gain of using completion and distances correction. On the other hand, this process gives the opportunity to localize all unknown nodes even when less than three RNs are detected unlike the classic trilateration.

TABLE III: Localization accuracy [meters].

	First scenario	Second scenario
Method 1	4.741	4.791
Method 2	1.2269	1.11
Method 3	0.555	0.537

The cumulative distribution function (CDF) of the localization error which is given by

$$\text{CDF}(x) = P(\text{error}_{\text{loc}} \leq x), \quad (11)$$

is shown in Fig. 6 for the first scenario, whereas Fig. 5 shows the CDF corresponding to the second scenario. Where $\text{error}_{\text{loc}}$ is the localization error.

For the first scenario, our developed system based on distance matrix completion and correction can ensure a localization error less than 0.37 m for 50% of unknown positions, outperforming the classical trilateration and trilateration with matrix completion having 3.8 m and 0.96 m, respectively. Also, we notice that 80% of unknown positions have a localization error less than 0.81 m, outperforming the classical trilateration and trilateration combined with completion reaching this error at 40% and 21% of unknown nodes, respectively. Therefore, integrating a denoising process of completed distances greatly improves the performance of trilateration.

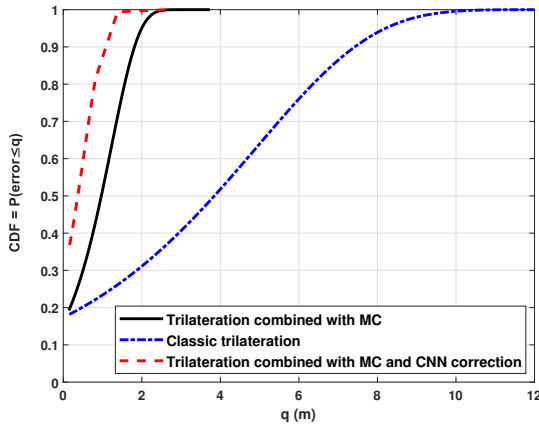


Fig. 5: The CDF corresponding to scenario 1 when using 10 RN and 25 unknown nodes with sigma shadowing equal to 2.

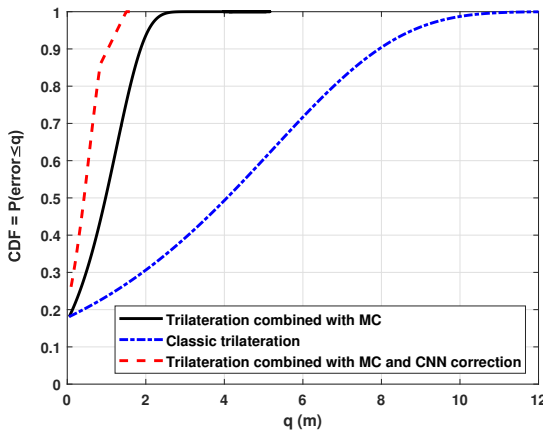


Fig. 6: The CDF corresponding to scenario 2 when using 10 RN and 50 unknown nodes with sigma shadowing equal to 2.

V. CONCLUSION

In this paper, we have presented a distance matrix completion combined with a minimization error process, based on convolutional neural networks (CNN), for indoor localization. To perform localization, we introduce the trilateration technique exploring all completed and corrected pairwise distances between nodes to localize and reference nodes (RNs). To validate the proposed system, different simulations were carried out in a noisy environment. Obtained results indicate that trilateration combined with distance matrix completion and correction outperforms both the classic trilateration and the trilateration combined with traditional matrix completion, in terms of localization accuracy. We have also shown that the combination of matrix completion and the minimization error process improves significantly the accuracy of distance matrix recovery compared to the traditional matrix completion. To validate the merits

of the proposed localization scheme, we intend to do real experiments in order to support simulation results in future work.

ACKNOWLEDGMENT

This work was supported by CY Initiative of Excellence (Investissement d’Avenir Grant: ANR-16-IDEX-0008 and ASIA chair of Excellence Grant: PIA/ANR-16-IDEX-0008).

REFERENCES

- [1] M. Ficco, F. Palmieri, and A. Castiglione, “Hybrid indoor and outdoor location services for new generation mobile terminals,” *Personal and ubiquitous computing*, vol. 18, no. 2, pp. 271–285, 2014.
- [2] I. Guvenc and C.-C. Chong, “A survey on toa based wireless localization and nlos mitigation techniques,” *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.
- [3] T. C. Karalar and J. Rabaey, “An rf tof based ranging implementation for sensor networks,” in *2006 IEEE International Conference on Communications*, vol. 7. IEEE, 2006, pp. 3347–3352.
- [4] X. Cheng, A. Thaler, G. Xue, and D. Chen, “Tps: A time-based positioning scheme for outdoor wireless sensor networks,” in *IEEE INFOCOM 2004*, vol. 4. IEEE, 2004, pp. 2685–2696.
- [5] Z. Yang, Z. Zhou, and Y. Liu, “From rssi to csi: Indoor localization via channel response,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 2, pp. 1–32, 2013.
- [6] P. Kumar, L. Reddy, and S. Varma, “Distance measurement and error estimation scheme for rssi based localization in wireless sensor networks,” in *2009 Fifth international conference on wireless communication and sensor networks (WCSN)*. IEEE, 2009, pp. 1–4.
- [7] F. Zafari, I. Papapanagiotou, and K. Christidis, “Microlocation for internet-of-things-equipped smart buildings,” *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 96–112, 2015.
- [8] W. Njima, I. Ahriz, R. Zayani, M. Terre, and R. Bouallegue, “Comparison of similarity approaches for indoor localization,” in *2017 IEEE 13th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2017, pp. 349–354.
- [9] Z. Yang and Y. Liu, “Quality of trilateration: Confidence-based iterative localization,” *IEEE Transactions on parallel and distributed systems*, vol. 21, no. 5, pp. 631–640, 2009.
- [10] H. Zhang, Y. Liu, and H. Lei, “Localization from incomplete euclidean distance matrix: Performance analysis for the svd-mds approach,” *IEEE Transactions on Signal Processing*, vol. 67, no. 8, pp. 2196–2209, 2019.
- [11] L. T. Nguyen, J. Kim, S. Kim, and B. Shim, “Localization of iot networks via low-rank matrix completion,” *IEEE Transactions on Communications*, vol. 67, no. 8, pp. 5833–5847, 2019.
- [12] W. Njima, R. Zayani, I. Ahriz, M. Terre, and R. Bouallegue, “Beyond stochastic gradient descent for matrix completion based indoor localization,” *Applied Sciences*, vol. 9, no. 12, p. 2414, 2019.
- [13] W. Njima, M. Chafii, A. Nimr, and G. Fettweis, “Deep learning based data recovery for localization,” *IEEE Access*, vol. 8, pp. 175 741–175 752, 2020.
- [14] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.