Received September 9, 2020, accepted September 17, 2020, date of publication September 24, 2020, date of current version October 6, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3026615

Deep Learning Based Data Recovery for Localization

WAFA NJIMA[1], (Member, IEEE), MARWA CHAFII[1], (Member, IEEE), AHMAD NIMR[2], (Member, IEEE), AND GERHARD FETTWEIS[2], (Fellow, IEEE)

[1]ETIS , UMR8051, CY Cergy Paris Université, ENSEA, CNRS, 95000 Cergy, France
[2]Vodafone Chair Mobile Communication Systems, Technische Universitat Dresden, 01069 Dresden, Germany

Corresponding author: Wafa Njima (wafa.njima@ensea.fr)

This work was supported by the Paris Seine Initiative (CY Cergy Paris Université, CNRS, ENSEA, and ETIS Laboratory) Project through the ASIA Chair of Excellence under Grant PIA/ANR-16-IDEX-0008.

ABSTRACT In this paper, we study the problem of euclidean distance matrix (EDM) recovery aiming to tackle the problem of received signal strength indicator sparsity and fluctuations in indoor environments for localization purposes. This problem is addressed under the constraints required by the internet of things communications ensuring low energy consumption and reduced online complexity compared to classical completion schemes. We propose EDM completion methods based on neural networks that allow an efficient distance recovery and denoising. A trilateration process is then applied to recovered distances to estimate the target's position. The performance of different deep neural networks (DNN) and convolutional neural networks schemes proposed for matrix reconstruction are evaluated in a simulated indoor environment, using a realistic propagation model, and compared with traditional completion method based on the adaptative moment estimation algorithm. Obtained results show the superiority of the proposed DNN based completion systems in terms of localization mean error and online complexity compared to the classical completion.

INDEX TERMS Convolutional neural networks (CNN), deep neural networks (DNN), indoor localization, matrix completion, received signal strength indicator (RSSI), trilateration.

I. INTRODUCTION

With the wide-scale proliferation of wireless communication, location-based services (LBS) are attracting a high interest especially related to internet of things (IoT) applications. The most fundamental common need for these LBS applications is to accurately estimate the target's position, based on the collected environmental data by a network of sensor nodes [1], [2]. The wide variety of LBS services and applications can include both outdoor and indoor situations. The global positioning system (GPS) can guarantee accurate and precise localization information in the outdoor area. However, its performance suffers from a drastic degradation in indoor environments due to the attenuation and the blockage of satellite signals. Consequently, the location demand for indoor positioning scenarios is more challenging than that in outdoor scenarios [3], [4]. Thus, LBS brings new challenges for developing promising high accuracy indoor localization techniques while respecting time-critical and energy efficiency constraints, in order to satisfy IoT requirements.

The majority of existing indoor localization works focus on the 2-dimensional (2-D) or 3-dimensional (3-D) coordinates [5], while some other works provide different location information (floor identification, zone identification, moving status, etc) [6], [7] [8]. The recently developed indoor localization techniques are essentially based on radio frequency (RF) technologies (e.g. Wi-Fi [9], [10], Bluetooth [11], radio frequency identification (RFID) [12], ultra wide-band (UWB) [13]) and emerging technologies (e.g. visible light [14], ultrasound [15], magnetic field [16]). The most popular among these techniques are the Wi-Fi based localization methods operated through software implementations and firmware upgrades on Wi-Fi based communication systems. Existing Wi-Fi based indoor positioning systems are mainly based on geometric mapping methods [17], [18] or on location fingerprints [19]. For geometric mapping schemes, spatial parameters like distance or direction, are derived from physical measurements essentially angle of arrival (AoA) [20], time of arrival (ToA) [21] and

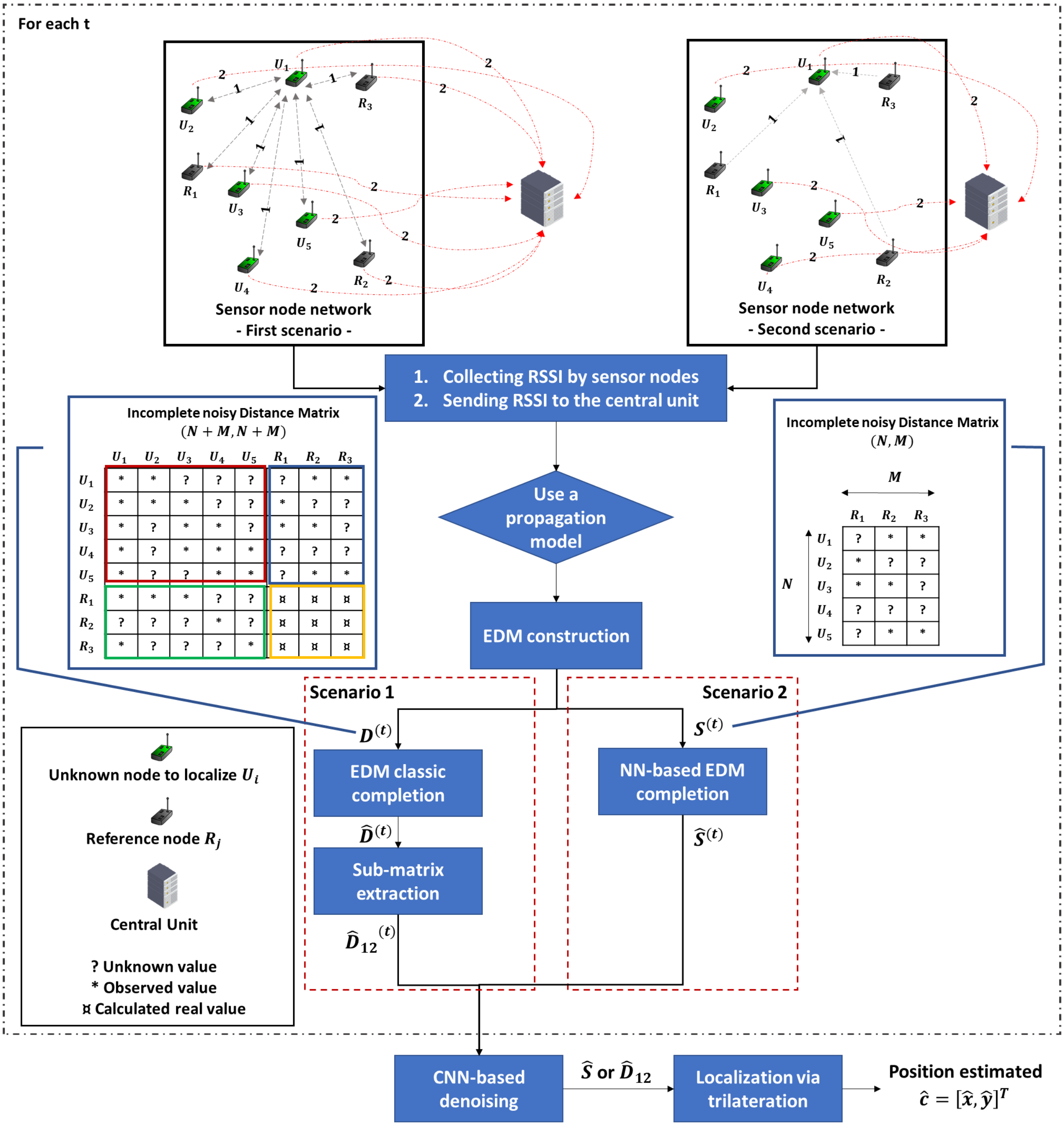
The associate editor coordinating the review of this manuscript and approving it for publication was Lei Guo.

received signal strength indicator (RSSI) [22]. The target's position is then provided using geometric algorithms (triangulation or trilateration). The localization performance of such algorithms is heavily affected by the limitation of communication range and multipath and shadowing effects caused by reflectors such as walls and other obstacles resulting from human movement [23] for example. These indoor propagation conditions complicate propagation modeling, and consequently, the relation between the measured received signal parameters and distances is perturbed. Therefore, pairwise sensor-node distances are noisy and only partially known, which makes accurate indoor localization challenging. To provide a reliable and efficient distance information for localization aims considering radio signal fluctuations and sparsity, the distance matrix completion and correction is proposed to recover the complete euclidean distance matrix (EDM) from a small number of noisy measured distance entries.

Several algorithms and approaches have been proposed and studied to recover the EDM approximating its missing entries and minimizing the noise on the observed entries from its noisy version, for RSSI based schemes. The classical matrix completion problem is formulated as a rank minimization problem solved by advanced optimization algorithms as described in [24]. When providing such recovering method, approximated and calculated data still needs enhancement, which can be achieved by means of a refinement process. In addition, a fully connected wireless network is required such that all nodes communicate with each others (reference nodes (RN) and unknown nodes to be localized), and the completion preprocessing step needs to be performed online. Online multi-decision making schemes have been used in [25]–[27], to solve a related problem. However, indoor localization systems require low energy consumption and low computational complexity to meet the IoT constraints. Therefore, several deep learning methods have been explored recently to shift the online complexity to an offline phase. The convolutional neural networks (CNN) have been used to overcome noise and uncertainty during RSSI-based localization by developing localization frameworks using CNN-based data augmentation [28], [29]. Authors in [30] deploy a cascade of multiple deep neural networks (DNN) to recover the original EDM from the noisy observed matrix which can achieve an accurate reconstruction performance of the EDM. Special types of neural networks (NN) have been explored, essentially autoencoders [31], [32] and generative adversarial networks (GAN) [33], to regenerate the complete noise-free original data in several fields. Currently, such deep learning (DL) methods are widely applied to reconstruct clean data from its noisy version for image denoising [34], diagnosis improvements of ECG signals [35], text correction and completion [36] and EDM or RSSI completion and denoising for localization accuracy improvements [37]. Authors in [38] tackle the problem of RSSI fluctuation and the sparsity of Wi-Fi signals in dynamic environments to improve the localization accuracy using a stacked denoising autoencoder to extract RSSI measurements based on Wi-Fi. Authors in [39] and [40] explore GAN to deal with the limited RSSI samples resulting from non-covered spaces or irregular RSSI distributions in complex indoor environments. The different mentioned existing DL-based methods, which are dedicated for data completion and correction to improve the indoor localization accuracy, are based on unsupervised learning and require a full connection between sensor nodes (communication between all unknown nodes to be localized and all RN) which increases the energy consumption and the time-response. To perform localization, authors in [41] apply a factorization process to the recovered distances matrix in order to obtain the estimated coordinates of unknown nodes. This factorization requires that the matrix is semidefinite positive (SDP). However, if the matrix is not SDP, the problem can also be solved by introducing semidefinite relaxations (SDR) [42]. The computational complexity of SDR-based localization techniques is closely related to the problem size. Therefore, it is only suitable to medium size networks due to the high required running time. The localization task can be also ensured by the fingerprinting technique. The main drawback of such technique is that an offline radio map should be constructed and frequently updated due to the environmental changes.

To overcome the aforementioned challenges, this paper addresses the matrix completion and correction problem under the constraints required by IoT communications. We propose to deploy supervised DL based methods for EDM completion as a preprocessing step for indoor localization. Consequently, we shift the online complexity to an offline phase ensuring a real-time response even for large networks, while achieving a centimeter-level accuracy. Thus, an offline-construction and designing of a generic and representative model is required. Such learned model is applied online to get the predicted complete data. We propose different architectures of DNN and CNN to ensure accurate distance regeneration in order to efficiently reconstruct distances data used for localization. Simulation results show that the proposed DL based schemes achieve higher localization accuracy than state-of-the-art methods that consider classical data completion schemes. Moreover, the different proposed schemes are compared in order to find the best recovery method taking into account the trade-off between localization accuracy and online complexity.

Thus, we design recovery schemes that exploit the correlations between measurements in order to fill the missing information and minimize the error on observed noisy measurements. After distance recovery, the localization is performed by means of trilateration technique using the complete pairwise inter-sensor nodes distances instead of only relying on a partial number of pairwise distances derived from measured RSSI. Applying this localization method does not need any extra requirements to be used once the EDM is completed. This technique provides the nodes' coordinates by combining measurements from different RN nodes and can be applied if we have at least three detected RN. By using



	U_1	U_2	U_3	U_4	U_5	R_1	R_2	R_3
U_1	*	*	?	?	?	?	*	*
U_2	*	*	*	?	?	*	?	?
U_3	*	?	*	*	?	*	*	?
U_4	*	?	*	*	*	?	?	?
U_5	*	?	?	*	*	?	*	*
R_1	*	*	*	?	?	¤	¤	¤
R_2	?	?	?	*	?	¤	¤	¤
R_3	*	?	?	?	*	¤	¤	¤

	R_1	R_2	R_3
U_1	?	*	*
U_2	*	?	?
U_3	*	*	?
U_4	?	?	?
U_5	?	*	*

FIGURE 1. System model mentioning different system's steps.

the recovered EDM, we aim to improve the localization accuracy of trilateration. From the numerical experiments, we demonstrate that using CNN and DNN-based recovery schemes achieves 62% and 79% reduction in the localization mean error over the conventional completion approach, respectively. Moreover, DL-based completion schemes are associated with a low online complexity and require a partial inter-nodes connection (communication between unknown nodes to be localized and RN only) in order to be adapted to indoor localization for an IoT context, where the energy consumption is limited.

The remainder of this paper is organized as follows: In Section II, we give a detailed description of the system model and the considered scenarios. We present the proposed DNN and CNN schemes used for data completion and denoising in Section III and Section IV, respectively. Obtained simulation results are presented and discussed in Section V. Finally, we conclude our work in Section VI.

II. SYSTEM DESCRIPTION

We consider an indoor environment where U sensor nodes are deployed, including M RN with known positions and N unknown nodes to be localized, such that $U = M + N$. As depicted in Fig. 1, the localization scheme is performed through three steps; (i) data acquisition, (ii) data preprocessing, and (iii) node localization.

We consider two studied inter-nodes communication scenarios:

- Full inter-nodes communication: all sensor nodes communicate with each other.
- Partial inter-nodes communication: unknown nodes to localize communicate only with RNs.

The RSSI measurements are sent to a central unit, which performs data preprocessing and localization. Afterwards, the estimated coordinates are sent to the considered sensor nodes periodically or only when requested. Therefore, the complexity is shifted to the central unit to meet the IoT energy constraints at the devices. In this section, we explain in details the system model and the inter-nodes communication conditions for two studied scenarios.

A. DATA ACQUISITION

Let $P_{r_{ij}}$ be the RSSI measured at the i-th sensor of the signal transmitted by the j-th sensor. It can be expressed as

$$P_{r_{ij}} = P_t - P_{L_{ij}} + B_\sigma \quad [\text{dBm}], \tag{1}$$

where P_t refers to the transmitted power, which is considered constant for all sensors, B_σ is a Gaussian random variable representing the shadowing effects, and $P_{L_{ij}}$ is the pathloss calculated using the model

$$P_{L_{ij}} = P_{L_0} + 20\log_{10}(f) + 10\mu\log_{10}(\frac{d_{ij}}{d_0}), \tag{2}$$

where P_{L_0} denotes the pathloss value at a reference distance d_0, f the carrier frequency, μ the pathloss exponent, and d_{ij} the distance between the sensor pair (i, j).

When considering the first communication scenario, $i = 1, 2, \ldots, U$ and $j = 1, 2, \ldots, U$. Considering the shadowing effects, T RSSI measurements are collected by each sensor node received from all other nodes following (1). However, when considering the second scenario, we work with a partially connected network. T RSSI are measured by each unknown node of signals received only from RN. Thus, $i = 1, 2, \ldots, N$ and $j = 1, 2, \ldots, M$.

B. DATA PREPROCESSING

Collected data is transferred to a central unit which performs three preprocessing tasks, namely, EDM construction, EDM completion, and EDM refinement.

1) EDM CONSTRUCTION

We consider two indoor propagation scenarios with different inter-nodes communication conditions.

- **Full inter-nodes communication:** This scenario is considered when we apply classical approaches to complete the EDM containing squared pairwise distances between all sensor nodes $\boldsymbol{D}^{(t)}$, $t \in \{1, 2, \ldots, T\}$,

$$\boldsymbol{D}^{(t)} = \begin{bmatrix} \boldsymbol{D}_{11}^{(t)} & \boldsymbol{D}_{12}^{(t)} \\ \boldsymbol{D}_{21}^{(t)} & \boldsymbol{D}_{22} \end{bmatrix}, \tag{3}$$

  where $\boldsymbol{D}_{12}^{(t)} \in \mathbb{R}^{+N\times M}$ and $\boldsymbol{D}_{21}^{(t)} \in \mathbb{R}^{+M\times N}$ represent the distances between the unknown nodes and RNs measured at the unkonwn nodes and the RNs, respectively. Moreover, $\boldsymbol{D}_{11}^{(t)} \in \mathbb{R}^{+N\times N}$ denotes the distances between unknown nodes, whereas $\boldsymbol{D}_{22} \in \mathbb{R}^{+M\times M}$ contains the distances between the RNs. Note that, $\boldsymbol{D}_{22}$ is exactly known i.e. it is fixed. That is why we remove the index t, since it is the same matrix for each t. Whereas $\boldsymbol{D}_{12}^{(t)}$, $\boldsymbol{D}_{21}^{(t)}$ and $\boldsymbol{D}_{11}^{(t)}$ are computed based on the RSSI values, replacing (2) in (1), and thus, they include noisy and missing entries.
- **Partial inter-nodes communication:** This completion scheme is applied when we consider the second communication scenario. The EDM to be recovered $\boldsymbol{S}^{(t)} = \boldsymbol{D}_{12}^{(t)} \in \mathbb{R}^{+N\times M}$, obtained from collected RSSI values and applying (1), contains only distances between the unknown nodes to localize and RNs at the t-th measurement. It corresponds to the sub-matrix $\boldsymbol{D}_{12}^{(t)}$ considered in the first scenario.

2) EDM COMPLETION

- **Classical completion:** This completion scheme is considered in the first inter-nodes communication scenario. Completed matrix is denoted as $\hat{\boldsymbol{D}}^{(t)}$. The completion problem is formulated as an optimization problem which aims to minimize the rank of the distance matrices solved by the adaptative moment estimation (ADAM) [43] advanced method. More details on the problem formulation and used algorithms are presented in our previous work [24].
- **NN-based completion:** The completion of each $\boldsymbol{S}^{(t)}$, denoted as $\hat{\boldsymbol{S}}^{(t)}$ is achieved based on a trained model using DNN or CNN as detailed in Section III and Section IV, respectively. Such scheme allows both distance completion and noise minimization on observed distances, since we train the model considering real distances as outputs. Thus, our model allows to estimate distances by applying completion if the distance is not observed or correction for noisy distances.

3) EDM REFINEMENT

Completed distances are introduced to a proposed refinement process. If the completed distances are still damaged by noise, a refinement process is performed as explained in Section IV-C. Finally, the central unit applies the trilateration on obtained preprocessed distances contained on $\hat{\boldsymbol{S}}$ to provide the estimates of unknown positions. We notice that $\hat{\boldsymbol{S}}$ represents the output of the CNN-refinement process when applied. If not applied, $\hat{\boldsymbol{S}}$ refers to the mean of the T completed distance matrices.

C. TRILATERATION FOR NODES LOCALIZATION

After completing and denoising the distances obtained from the RSSI measurements using the signal propagation model expressed in (1), the localization is performed using trilateration [44]. It is a geometric method that combines distances between the node to localize and RN, in order to accurately estimate the target's position. Trilateration explores the equations of spheres centered at an unknown node to determine its accurate position in 2-D or 3-D. In this paper, we consider localization in a 2-D plane. For localization purposes, only the sub-matrix $\hat{\boldsymbol{D}}_{12}$ of $\hat{\boldsymbol{D}}$ is used concerning the first communication scenario. Whereas the matrix $\hat{\boldsymbol{S}}$ is used for the second scenario. Let $\hat{\boldsymbol{S}}$ be the matrix containing pairwise distances between unknown nodes and RN, obtained after the preprocessing task, we define:

- $\hat{c}_n = [\hat{x}_n, \hat{y}_n]^T$, $n = 1 \cdots N$ are the estimated coordinates of the n-th unknown node.
- and (x_m, y_m), $m = 1 \cdots M$ are the real known coordinates of RN.

The coordinates $\hat{c}_n$ are estimated as follows:

$$\hat{c}_n = (\boldsymbol{G}^T\boldsymbol{G})^{-1}\boldsymbol{G}^T\boldsymbol{a}_n. \tag{4}$$

Here, $v_m^2 = x_m^2 + y_m^2$. $\boldsymbol{a}_n \in \mathbb{R}^{M-1\times 1}$, $\boldsymbol{G} \in \mathbb{R}^{M-1\times 2}$ defined as:

$$\begin{aligned}\boldsymbol{a}_n[m] &= \frac{1}{2}\left(v_{m+1}^2 - v_1^2 - \hat{d}_{m+1,n}^2 + \hat{d}_{1,n}^2\right)\\ \boldsymbol{G}[m,:] &= [x_{m+1} - x_1 \;\; y_{m+1} - y_1]\,.\end{aligned} \tag{5}$$

where $\hat{d}_{m,n}^2$ are the entries of the preprocessed squared distance matrix $\hat{\boldsymbol{S}}$ expressed as:

$$\hat{d}_{m,n}^2 = \hat{\boldsymbol{S}}[n,m] = (x_m - \hat{x}_n)^2 + (y_m - \hat{y}_n)^2. \tag{6}$$

III. PROPOSED DNN-BASED COMPLETION

In this section, we consider the second inter-nodes communication scenario. We aim to exploit DNN to complete unknown distances and correct observed noisy distances. In the following, we explain in details different explored DNN architectures.

A. OVERVIEW OF DNN

DNN, which can model complex problems using a deep-layered structure, are capable of learning high-level features. They have been widely investigated recently in different fields due to its promising performance in many benchmark problems. In this paper, we use such NN architectures to complete partially known distances where distance data received at the input layer is propagated to one or more middle layers called *hidden layers*. The weighted sums from these hidden layers are propagated to the output layer, which presents the estimated complete distance data. The weights of such network are updated iteratively by error back-propagation using an optimization algorithm. Determining the weights and biases of the network means training the network. Once trained and validated, the DNN model can perform completion by computing the output of the network using the weights determined during the training phase.

In order to perform completion using a DNN, we introduce an incomplete distance vector ($N_0 \times 1$) as input while the DNN's output corresponds to the complete associated distance vector ($N_{L+1} \times 1$), as illustrated in Fig. 2. Let N_l be the number of neurons for the l^{th} layer, $0 \leq l \leq L+1$ where L is the number of hidden layers. $\boldsymbol{b}_l \in \mathbb{R}^{N_l\times 1}$ and $\boldsymbol{W}_l \in \mathbb{R}^{N_l\times N_{l-1}}$ denotes the biases and the weights matrix, respectively. The input incomplete vector is denoted by $\boldsymbol{i}_{(0)} \in \mathbb{R}^{N_0\times 1}$ and the output estimated vector is $\boldsymbol{o} \in \mathbb{R}^{N_{L+1}\times 1}$. The output vector of the l^{th} layer can be expressed as

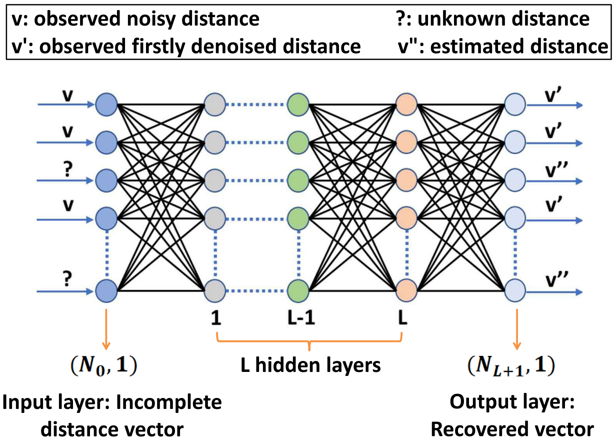

v: observed noisy distance
?: unknown distance
v': observed firstly denoised distance
v": estimated distance
v
v
?
v
?
v'
v'
v''
v'
v''
1
L-1
L
$(N_0, 1)$
L hidden layers
$(N_{L+1}, 1)$
Input layer: Incomplete distance vector
Output layer: Recovered vector

FIGURE 2. Structure of DNN at test phase for correction and completion distances matrix.

$$\boldsymbol{o}(l) = g_l(\boldsymbol{b}_{(l)} + \boldsymbol{W}_{(l)}\boldsymbol{i}_{(l-1)}), \quad \boldsymbol{i}_{(l)} = \boldsymbol{o}_{(l)}, \tag{7}$$

where the input vector $\boldsymbol{i}_{(l)}$ undergoes a linear transformation represented by $\boldsymbol{W}_{(l)}$, a bias vector $\boldsymbol{b}_{(l)}$, and then a nonlinear activation function $g_{(l)}$ is applied elementwise. The activation function used in this paper is Relu, which is defined as

$$g_l(\boldsymbol{x}_{(l)})\,[n] = \max\left(\boldsymbol{x}_{(l)}\,[n]\,, 0\right). \tag{8}$$

The gap between the ideal correct distances and the estimated complete distances computed by the generated DNN based on its current weights is referred to as the loss. The loss function $loss(\lambda) = loss(W, B)$ measures the difference between $\boldsymbol{o}_{(L+1)}^{(P)}$ and $\boldsymbol{o}_{(L+1)}^{(R)}$ denoting predicted outputs and real outputs, respectively. It corresponds to the normalize mean square error between $\boldsymbol{o}_{(L+1)}^{(P)}$ and $\boldsymbol{o}_{(L+1)}^{(R)}$. Thus, the goal of training DNN is to find a set of weights to minimize the average loss over a large training set using a back-propagation method. To minimize the loss, optimization algorithms are used to iteratively update λ. Then, the model is validated using a validation data set. For evaluation aims, the algorithm must accurately complete the unknown components of the distance vector in a test set of incomplete distance vectors. The obtained output estimated distances are then considered by our system to perform trilateration.

When working with DNN, the online complexity represents the number of mathematical operations (multiplications and summations) needed to activate neurons in all DNN layers. We can neglect the cost of summations since it is not complex compared to multiplications. The number of such multiplication operations $N_{mul,DNN}$ required to transit from the 1st layer to the output layer can be given by:

$$N_{mul,DNN} = \sum_{l=1}^{L+1} N_{l-1}N_l. \tag{9}$$

B. DNN SCHEMES FOR DATA COMPLETION

In this part, we aim to exploit DNN to complete and denoise the distance information contained in $\{\boldsymbol{S}^{(t)}\}$. First, given a set of observed incomplete distance matrices which are used for training, a model is built to predict complete correct distances. During training, we aim to find the best DNN architecture which allows a good distance recovery. Thus, the trained model is used to complete and denoise distances and we compare the recovered distances to the accurate distances. Once verified, we give as input the observed distances corresponding to an unknown node to localize and as output we receive the estimated complete and denoised corresponding distances. Matrices to be completed can be reorganized in two different ways to construct the inputs and the outputs of the DNN model.

- **DNN for a single node distance completion:** The matrices $\{\boldsymbol{S}^{(t)}\}$ are reorganized in NT vectors of size $M \times 1$. These vectors are introduced to the DNN input. Note that each vector represents distances between all RN and a given unknown node at a given t. We obtain NT estimated distance vectors of size $M \times 1$, as model outputs. Considering (7), we have $\boldsymbol{i}_0^{(t,n)}[m] = \boldsymbol{S}^{(t)}[n, m]$. In this case, $N_0 = M$ and $N_{L+1} = M$.
- **DNN for all nodes distance completion:** The matrices $\{\boldsymbol{S}^{(t)}\}$ are reorganized in T vectors of size $NM \times 1$. These vectors are introduced to the DNN. Note that each vector represents distances between all unknown nodes and all RN at t. As outputs, we obtain the completed vectors representing $\hat{\boldsymbol{S}}^{(t)}$. Considering (7), we have $\boldsymbol{i}_0^{(t)}[1 : N_0] = \text{vec}\left\{\boldsymbol{S}^{(t)}\right\}$, where $\text{vec}\{\cdot\}$ is the vectorization operation. In this case, $N_0 = NM$ and $N_{L+1} = NM$.

IV. PROPOSED CNN-BASED COMPLETION AND REFINEMENT

In this section, we exploit CNN to reconstruct partially known and noisy distances considering the second inter-nodes communication scenario. We also propose a CNN-based completed distances refinement process which can be applied after completion in the two mentioned inter-nodes communication conditions. In the following, we explain in details different explored tools and architectures.

A. OVERVIEW OF CNN

CNN are a variant of NN that have been widely used in different promising domains, for classification and regression problems. The architecture of such networks is composed essentially of:

- **A feature extraction module** which can contain one or many convolutional layers and one or many pooling layers. The convolution operation aims to extract the inputs' features using filters initialized randomly with a predefined size. The pooling layers sub-sample the outputs of the previous convolutional layer reducing their resolution, by pooling over local neighborhood, to reduce the complexity. The use of sub-sampling layers is not mandatory, it depends on the studied problem and on the size of used data.
- **Fully connected layers** which are responsible of determining the network's outputs based on an appropriate activation function.

A variety of CNN architectures can be derived. In this part, we describe a 2-D CNN architecture and we remove the pooling layers since we aim to save all extracted features. In this CNN architecture, we first apply 2-D linear convolution with a bank of Q_l filters each of size $P_l \times P_l$ to construct feature maps as convolutional layers outputs. The input matrices to the l-th layer where $1 \leq l \leq L+1$ are denoted as $\boldsymbol{I}_{(r_{l-1})}$, $r_{l-1} = 0 \cdots R_{l-1} - 1$. Here, R_{l-1} and R_l denotes the number of input and output matrices of the l-th layer, respectively. It can be computed with relation to Q_l as

$$R_l = Q_l R_{l-1}, \text{ where } R_0 = 1. \tag{10}$$

Each input matrix $\boldsymbol{I}_{(r_{l-1})} \in \mathbb{R}^{N_{l-1} \times M_{l-1}}$ is convolved with a filter $\boldsymbol{W}_{(q_l)} \in \mathbb{R}^{P_l \times P_l}$, $q_l = 0 \cdots Q_l - 1$, to produce the matrix $\bar{\boldsymbol{O}}_{q_l + r_l Q_l} = \boldsymbol{W}_{(q_l)} * \boldsymbol{I}_{(r_{l-1})} \in \mathbb{R}^{N_l \times M_l}$ defined by

$$\bar{\boldsymbol{O}}_{q_l + r_l Q_l}[m, n] = \sum_{u=0}^{P_l - 1} \sum_{v=0}^{P_l - 1} \boldsymbol{W}_{(q_l)}[u, v] \cdot \boldsymbol{I}_{(r_{l-1})}[m-u, n-v]. \tag{11}$$

Accordingly, $N_l \times M_l$ is the size of output matrices and it can be calculated from the input size $N_0 \times M_0$ using $M_l = \max(M_{l-1} - P_l + 1, 1)$ and $N_l = \max(N_{l-1} - P_l + 1, 1)$, $l = 1 \cdots L$. We notice that when convolving the matrix by the filter, it slides across the image by a stride $= 1$. Thus, the obtained outputs of the l-th convoluitonal layer are smaller than its inputs. When the input size is smaller than the filter, the problem cannot be handled. In this case, a zero padding process can be introduced (it is the case of the CNN used for refinement), and the output of the corresponding dimension is of size 1. After adding a bias term $\boldsymbol{B}_{(l)}$, an activation function is used to get the output such that

$$\begin{aligned} \boldsymbol{O}_{(q_l + r_l Q_l)} &= g(\boldsymbol{W}_{(q_l)} * \boldsymbol{I}_{(r_{l-1})} + \boldsymbol{B}_{(l)}), \\ \boldsymbol{I}_{(q_l + r_l Q_l)} &= \boldsymbol{O}_{(q_l + r_l Q_l)}, \ l = 1 \cdots L + 1. \end{aligned} \tag{12}$$

The ReLU activation function is used and it is defined by

$$g\left(\boldsymbol{X}[m, n]\right) = \max\left(\boldsymbol{X}[m, n], 0\right). \tag{13}$$

The outputs of the last convolutional layer, i.e. the entries of the matrices $\{\boldsymbol{O}_{r_L} \in \mathbb{R}^{N_L \times N_{L+1}}\}$ are flattened in a vector $\boldsymbol{i}_0$ of size $R_L N_L M_L \times 1$. Afterwards, a fully connected (FC) layer similar to DNN, as given by (7), with only one hidden layer of Z_{L+1} neurons is used to obtain the output of size N_{L+2} elements.

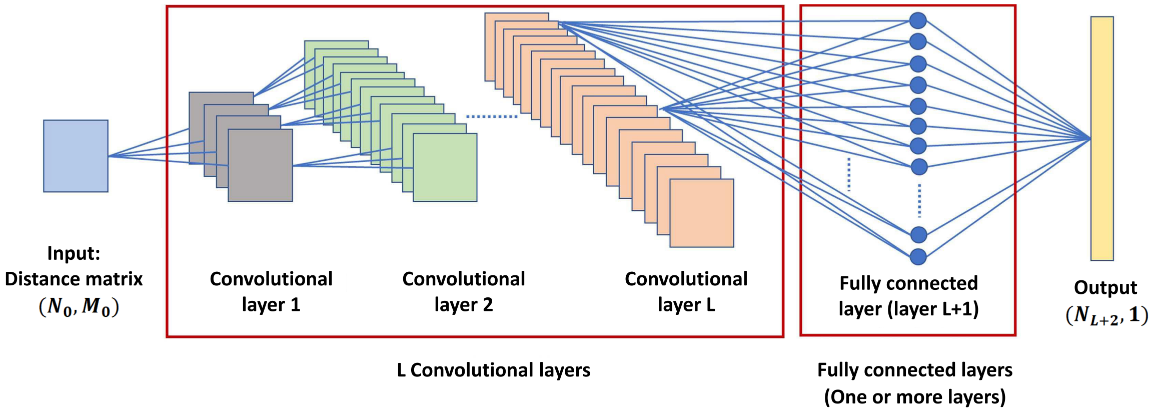


FIGURE 3. **Structure of generic CNN at test phase for distance recovery or refinement considering *L* convolutional layers and one fully connected layer.**

The online complexity of CNN includes the multiplications required by the convolutional layers $N_{mul-conv,CNN}$ given by

$$N_{mul-conv,CNN} = \sum_{l=1}^{L} R_{l-1} P_l^2 Q_l M_l N_l, \tag{14}$$

in addition to the number of multiplications at fully connected layers $N_{mul-fc,CNN}$, which can be calculated using (9) considering the corresponding input vector $\boldsymbol{i}_0$ with $N_0 = R_L N_L M_L \times 1$, namely,

$$N_{mul-fc,CNN} = R_L N_L M_L N_{L+1} + N_{L+1} N_{L+2}. \tag{15}$$

B. CNN SCHEME FOR DATA COMPLETION

In this part, we use CNN to recover inter-nodes pairwise distances contained in $\{\boldsymbol{S}^{(t)}\}$ and we remove the pooling layers since we aim to save all extracted features. During training, features are extracted to build a CNN model, based on a set of incomplete distance matrices, that predicts the complete correct distances. Once trained, we validate the model using a validation dataset. Then, the trained model is applied to recover the whole distance information. To evaluate the performance of such model, we compare the predicted distance information to the correct ones.

A set of incomplete distance data is generated to construct a 2-D CNN model conceived for data completion mission. From each, 80 % of data is considered for training to which we associate the corresponding real distance vectors as outputs. The remaining 20 % of data is used for model validation, where we forward the distances with missed entries and we receive as outputs the corresponding estimated entirely known vectors. Estimated outputs are compared to real outputs in order to evaluate the trained model and choose a suitable one. Once trained and validated, we introduce the T successive incomplete distance matrices $N \times M$ corresponding to the target to be localized, to the CNN model as inputs, where the outputs represent the estimated corresponding complete real distances. Considering (12), $I_{(0)} = S^{(t)} \in \mathbb{R}^{N \times M}$, $O_{L+2} \in \mathbb{R}^{NM \times 1}$, thus $N_0 = N$, $M_0 = M$ and $N_{L+2} = NM$. Obtained completed distances are then forwarded to a refinement process which aims to improve the estimation of complete distances.

C. CNN SCHEME FOR DATA REFINEMENT

After the completion, a combination of the T cleaned-up measurements is applied to get the final estimate before the trilateration. This refinement can be performed by averaging T corresponding measurements which work well when the cleaned measurements correspond to the complete clean distances. But, when the cleaned measurements are still noisy, a CNN-based refinement process is able to improve the accuracy obtained by simple averaging. In this part, we describe the CNN applied on completed distances for distance refinement aims. Only the sub-matrix $\hat{\boldsymbol{D}}_{12}^{(t)}$ considered in the first inter-nodes communication scenario or $\hat{\boldsymbol{S}}^{(t)}$ regenerated in the second scenario, to be used for localization, are considered by the refinement process.

The completed matrices $\{\hat{\boldsymbol{S}}^{(t)}\}$ are reorganized in a matrix $\boldsymbol{I}_0$ of size $T \times MN$, such that

$$\boldsymbol{I}_0[t, :] = \text{vec}\left\{\hat{\boldsymbol{S}}^{(t)}\right\}, \tag{16}$$

where vec $\{\cdot\}$ is the vectorization operation, and $\boldsymbol{I}_0[t, :]$ is the t-th row of $\boldsymbol{I}_0$. Note that each column of $\boldsymbol{I}_0$ represents T successive distances between a given unknown node and a given RN. During the training phase, we associate at the output of the CNN, to each input vector which corresponds to a column from I_0, the corresponding accurate distance. For the testing phase, we obtain the predicted distances. Corrected distances obtained at the output of the CNN are used for localization based on trilateration. Once predicted, location coordinates can be sent to nodes periodically or when requested. In this case, $N_0 = T$, $M_0 = 1$ and $N_{L+2} = 1$.

V. SIMULATION RESULTS

In this section, we evaluate our proposed DL based localization schemes (denoted as method 3, 4, 5, 6, 7, 8) and compare

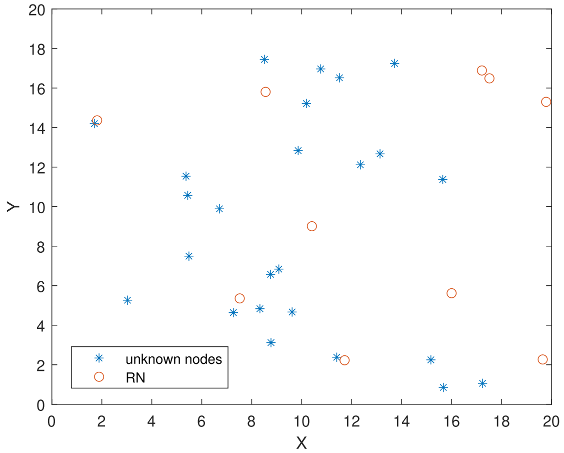


FIGURE 4. Configuration of the wireless sensor network considering $N = 25$.

TABLE 1. Details of the trained networks.

NN model	NN architecture	#calls	#Mult./call×10
CNN1$(10 \times 1, 1)$	$\text{conv}_1(30,3)$- $\text{conv}_2(10,2)$- $\text{conv}_3(20,2)$- $\text{FC}(20)$	250	146912
CNN2 $(10 \times 1, 1)$	$\text{conv}_1(30,3)$- $\text{conv}_2(20,2)$- $\text{conv}_3(20,2)$- $\text{FC}(20)$	250	245552
CNN3$(10 \times 1, 1)$	$\text{conv}_1(30,3)$- $\text{conv}_2(30,2)$- $\text{FC}(20)$	250	47792
CNN4$(25\times 10, 250)$	$\text{conv}_1(120,3)$- $\text{conv}_2(40,3)$- $\text{FC}(100)$	10	6614692
DNN1$(10, 10)$	$L_1(30)$-$L_2(10)$	250	700
DNN2$(250, 250)$	$L_1(45)$	10	2250

them with the classical trilateration (denoted as method 1) as well as with the analytical approach of matrix completion based localization (denoted as method 2). The performance is assessed in terms of distance matrix regeneration error and localization accuracy. Thus, we define the following two metrics:

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

$$\text{MSE}_{\text{dis}} = \frac{1}{MN} \sum_{n=1}^{N} \sum_{m=1}^{M} \text{E}\left[\left| \boldsymbol{S}[n,m] - \hat{\boldsymbol{S}}[n,m] \right|^2 \right], \quad (17)$$

  where $\boldsymbol{S}$ and $\hat{\boldsymbol{S}}$ denote the exact and estimated pairwise distances between the unknown nodes and the RNs, respectively. If the CNN-based completed distances refinement is applied, $\hat{\boldsymbol{S}}$ is the output of such process. Otherwise, it is obtained by averaging T recovered matrices $\hat{\boldsymbol{S}}^{(t)}$ over T measurements.
- Localization accuracy: it measures the average error in the coordinate estimation and it is defined by:

$$\text{MSE}_{\text{loc}} = \frac{1}{N} \sum_{n=1}^{N} \text{E}\left[\|\boldsymbol{c}_n - \hat{\boldsymbol{c}}_n\|^2 \right], \quad (18)$$

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

A. ENVIRONMENTAL SETUP AND DL ARCHITECTURES

To evaluate the performance of the proposed schemes in terms of distance matrix recovery and localization accuracy, we have a set of sensor nodes placed randomly in the studied environment of 400 m^2. These nodes include $M = 10$ RN and $N = 25$ nodes to be localized as depicted in Fig. 4.

The number of measurements used for localization is set to $T = 10$ such that, T RSSI values are collected at each position (reference positions and unknown positions) for the first communication scenario and T RSSI values from M RN are collected at each unknown position considering the second communication scenario. These values are collected following (1). In this paper, we use realistic propagation parameters obtained from measurements conducted in our laboratory, which have been verified and validated. These experiments are performed using $P_t = 20$ dBm, $d_0 = 1$ m, $f = 2.4$ GHz and $\mu = 3.23$. The results presented in this paper are obtained when working in a noisy environment considering $\text{E}[B_\sigma]^2 = 2$. The data collected from the sensors are simulated data on Matlab.

Based on intensive experiments, the proposed CNN and DNN architectures satisfy a good trade-off between the obtained performance and the online computational complexity. Identifying the optimal values of the NN parameters and best architectures is defined by an empirical process requiring several experiments. Data were trained with different architectures, varying the number of layers and the number of neurons in each layer, to find the best architecture. Table 1, summarizes the chosen architectures and the online complexity in terms of number of multiplications per input without considering the number of calls. CNN$(N_0 \times M_0, N_{L+2})$ and DNN(N_0, N_{L+1}) refer to the chosen architecture with the corresponding input and output sizes. The terms $\text{conv}_l(Q, P)$ denotes the l-th convolutional layer with Q filters of size (P, P) and FC(Z) referees to a fully-connected layer with Z neurons. The notation '$-$' means there is another following layer. We present the architectures of six NN models, namely.

- CNN1: CNN architecture for completed data refinement when using classical matrix completion.
- CNN2: CNN architecture for completed data refinement when using DNN-based matrix completion.
- CNN3: CNN architecture for completed data refinement when using CNN-based matrix completion.
- CNN4: CNN architecture for matrix completion and error minimization.
- DNN1: DNN architecture for matrix completion and error minimization based on a single node distance completion.
- DNN2: DNN architecture for matrix completion and error minimization based on all nodes distance completion.

TABLE 2. Hyper-parameters corresponding to each NN model.

NN model	Learning rate	Max epochs
CNN1	0.005	2500
CNN2	0.005	2000
CNN3	0.005	2500
CNN4	0.05	200
DNN1	0.05	–
DNN2	0.01	–

We consider ADAM as an optimization algorithm to obtain convergence for CNN and DNN. The other hyper-parameters such as the mini batch size, the number of epochs and the learning rate were optimized for each model during the training phase to retain the best configurations illustrated in Table 1. Experimentally, the use of a dropout rate is not beneficial, thus, we did not use it. We use full batch training, e.g, we consider the whole training data for each optimization step because it has been verified experimentally based on our training data, that the full batch training significantly outperforms the mini batch training. In Table 2, we present the learning rate and the max epochs corresponding to each NN model. Note that when using DNN, the number of epochs is not fixed in the beginning. The algorithm keeps running until convergence.

B. DISTANCE RECOVERY PERFORMANCE

We investigate the distance recovery performance for the two inter-nodes communication scenarios, exploring proposed recovering methods adapted to each scenario. The first scenario is considered when applying the classical completion and the second scenario is considered when using NN models for completion. We present in Table 3, the error of distances prediction corresponding to each recovery method.

As first step, we begin by investigating the performance in terms of distance matrix recovery when only using the classical completion matrix and then when adding the proposed CNN-based refinement process to completed distances. The different results are shown in Table 3. It can be noticed that the reconstruction accuracy is better when correcting distances using a CNN model. Instead of having 14.59 m as matrix recovery error, the deployment of CNN improves the precision of distances prediction by 2.92 m. Such results demonstrate and prove the benefits of the integration of such refinement process in order to accurately estimate distances considering the first inter-nodes communication scenario.

Considering the second studied scenario when we have a partial inter-nodes communication, we deploy a DL model for distances regeneration. We easily notice that such methods are associated to a better recovery rate even when applying the simple average to completed distances instead of the CNN-based refinement process. Among these methods, the performance of CNN is worse than DNN. In order to improve the 9.39 m RMSE distance recovery, the deployment of another CNN for distance refinement improves considerably the precision of distances prediction by 3.43 m. When using DNN for a single node distance completion, for distance reconstruction combined with a CNN-based refinement process, we obtain 4.89 m as a matrix recovery error. The best distance recovery rate 3.1 m is obtained when exploring DNN for all nodes distance completion. This is due to the fact that DNN considers the correlation between data to complete and it also corrects observed data.

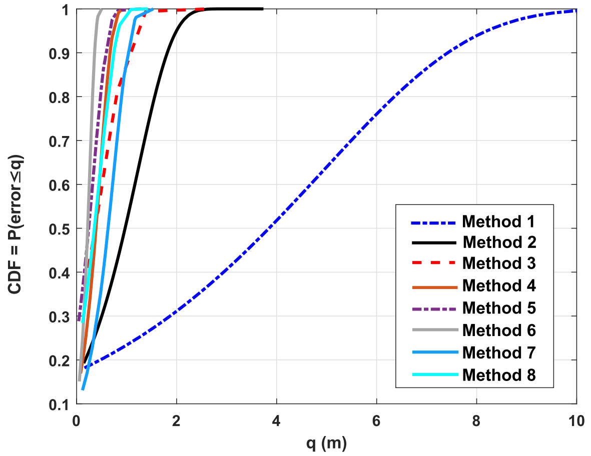


FIGURE 5. The CDF when using 10 RN and 25 unknown nodes with sigma shadowing equal to 2.

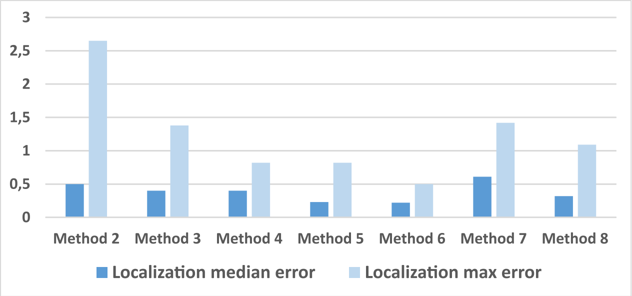


FIGURE 6. Localization median errors and localization max errors corresponding to each method with sigma shadowing equal to 2.

C. LOCALIZATION ACCURACY

A summary of the localization accuracy performance of eight schemes is given in Table 3 presenting the localization mean errors. The CDF of the localization error which is given by

$$\mathrm{CDF}(x) = P(\mathrm{error}_{\mathrm{loc}} \leq x), \tag{19}$$

is shown in Fig. 5. Where $\mathrm{error}_{\mathrm{loc}}$ is the localization error. Fig. 6 presents the localization median errors and the localization max errors. 50% of unknown nodes have an error less than the localization median error, which is determined considering $\mathrm{CDF}(q) = 0.5$ and projecting on the x-axis in order to find the corresponding error value q. The localization max error refers to the maximum reached localization error. This value is also determined based on CDFs.

It is clear that relying on available measurements only achieves the worse localization accuracy compared to the other methods that explore completion. In this case, the mean

TABLE 3. Comparison of algorithms' performances when considering 25 unknown nodes placed in an indoor area of 400 m^2 .

Method	Completion method	Refinement method	RMSE distance recovery [m²]	Loc. NMSE [m]	Online complexity decrease relative to Method 2 [%]
1	–	–	–	4.741	–
2	classical	Simple average	14.592	1.23	0
3	classical	CNN1	11.678	0.55	−17.66
4	DNN1	Simple average	6.03	0.47	99.91
5	DNN1	CNN2	4.89	0.34	70.4
6	DNN2	Simple average	3.1	0.26	99.98
7	CNN4	Simple average	9.39	0.74	68.19
8	CNN4	CNN3	5.959	0.47	62.49

error is about 4.7 m and it can achieve 9.94 m. Obtained results validate experimentally the beneficial use of distance matrix recovery and it ensures the localization of all unknown nodes even if we have less than three detected RN. The completed distances refinement with CNN correction process reduces the localization mean error introduced by the classical completion to 0.5 m, which is about 10% of the error of the first method. These results validate the gain of combining the classical completion and CNN-based refinement.

We can easily notice that reconstructing distances based on DL methods is always better than the classical completion even with a simple average for refinement, except for CNN-based completion which achieves 0.74 m as a mean error, which is about 0.19 m higher than the third method. This is due to the fact that DL methods approximate unknown matrix entries and correct the known noisy entries. When using CNN-based completion, the localization error can reach 1.42 m and 1.09 m with simple average based refinement and with CNN-based refinement, respectively.

The best localization accuracies are obtained when using DNN methods for distance recovery. The DNN for a single node distance completion with simple average based refinement is associated to 0.47 m as a localization mean error which is equal to the one achieved by CNN when followed by the CNN-based refinement module. The use of CNN correction process reduces this localization mean error to 0.34 m. But, the best accuracy 0.26 m is obtained when using DNN for all nodes distance completion, is about 20% of the mean error of the second method. Moreover, all unknown nodes placed in the studied area have a localization error less than 0.5 m and 50% of the targets have a localization error less than 0.22 m. These results confirm that DL methods especially DNN architectures are potential candidates for distance information recovery for localization improvement.

D. ONLINE COMPLEXITY OF DL-BASED MATRIX RECOVERY SCHEMES

In this section, we analyze the computational complexity of the proposed DL-based distance reconstruction schemes. The use of DL methods for matrix completion aims to minimize the computational cost. As mentioned before, all of them achieve a good localization accuracy and the choice of which method to explore depends on our computational and storage capacities. For this, we evaluate the online computational complexity. The complexity corresponding to each NN model used for matrix completion or matrix refinement is given in Table 1. We notice that we consider the preprocessing of only one input when presenting the number of multiplications without intervening the "calls" number.

We present in Table 3 the online complexity decrease of the NN-based distance recovery schemes in terms of number of multiplications, per T measurements, relative to the complexity of the classical completion (method 2), which is calculated directly as provided in [24]. It is expressed as follows:

$$N_{mul,MC} = I \times (6U^3 + 9U^2), \tag{20}$$

where I the number of iterations required by the optimization algorithm ADAM to reach the convergence. U is the total number of sensor nodes $U = N + M$. Using (20), the classical completion complexity, in terms of multiplications for a single distance matrix considering $T = 1$ is equal to 207913.125×10^3.

Considering the classical completion, method 3 which includes a distance's refinement based on CNN is better than a simple average based refinement, requiring 1.17 multiplications instead of 1 multiplication, which increases the complexity by 17%. Considering the trade-off between localization accuracy and online complexity, all NN-based schemes outperform the classical completion for all refinement methods used. The cascade of CNN4 and CNN3 used for distance completion and refinement, respectively, is beneficial in terms of distance recovery and localization accuracy with an additional low-complexity compared to simple refinement. However, DNN-based recovery schemes is still better in terms of localization accuracy and online complexity. We can easily notice that DNN2 with averaging is associated with the best localization accuracy and a significant low online complexity decreasing the classical complexity by 99.98%. Thus, even without CNN based refinement, such recovery scheme outperforms the others since this DNN model make use of all distances correlation.

VI. CONCLUSION

In this paper, we investigate the challenging distance completion issue for indoor localization. To perform localization, we use the trilateration based on collected RSSI measurements. For several indoor propagation conditions, these measurements as well as the derived distances are partially known and cannot serve efficiently for localization. Consequently, we have proposed to complete and correct distances considering two different inter-nodes communication scenarios namely full sensor network connection and partial sensor network connection. We have proposed different recovery schemes using classical matrix completion scheme based on ADAM optimizer and DL methods. Assuming that all nodes communicate between them, the classical matrix completion performs well, especially when combined with a CNN-based completed distances refinement module. However, a full inter-nodes communication is required and the whole process is online. Thus, we have proposed to shift the online recovering complexity and respect the requirements of IoT devices and real-time responses, using DL supervised methods. The gain of using such methods for distance recovery in a noisy indoor environment has been verified experimentally, especially DNN. This efficient completion introduces an important improvement of localization accuracy. The best localization accuracy equal to 0.26 m is obtained when using DNN with averaging, represents only 5% of the error obtained by the classical trilateration without completion. This recovery scheme is also associated with a low online complexity requiring only 1.08×10^{-4} multiplications instead of 1 for classical completion.

REFERENCES

[1] J. Schiller and A. Voisard, *Location-Based Services*. Amsterdam, The Netherlands: Elsevier, 2004.

[2] B. Rao and L. Minakakis, "Evolution of mobile location-based services," *Commun. ACM*, vol. 46, no. 12, pp. 61–65, Dec. 2003.

[3] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[4] D. Dardari, P. Closas, and P. M. Djuric, "Indoor tracking: Theory, methods, and technologies," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1263–1278, Apr. 2015.

[5] L. Chen, I. Ahriz, and D. Le Ruyet, "AoA-aware probabilistic indoor location fingerprinting using channel state information," *IEEE Internet Things J.*, early access, Apr. 24, 2020, doi: 10.1109/JIOT.2020.2990314.

[6] W. Njima, I. Ahriz, R. Zayani, M. Terre, and R. Bouallegue, "Deep CNN for indoor localization in IoT-sensor systems," *Sensors*, vol. 19, no. 14, p. 3127, Jul. 2019.

[7] B. A. Akram, A. H. Akbar, and O. Shafiq, "HybLoc: Hybrid indoor Wi-Fi localization using soft clustering-based random decision forest ensembles," *IEEE Access*, vol. 6, pp. 38251–38272, 2018.

[8] R. Elbakly, H. Aly, and M. Youssef, "TrueStory: Accurate and robust RF-based floor estimation for challenging indoor environments," *IEEE Sensors J.*, vol. 18, no. 24, pp. 10115–10124, Dec. 2018.

[9] S. He and S.-H.-G. Chan, "Wi-Fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 466–490, 1st Quart., 2016.

[10] Y. Ma, G. Zhou, and S. Wang, "WiFi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, pp. 1–36, Jul. 2019.

[11] S. Liu, Y. Jiang, and A. Striegel, "Face-to-face proximity estimation using Bluetooth on smartphones," *IEEE Trans. Mobile Comput.*, vol. 13, no. 4, pp. 811–823, Apr. 2014.

[12] L. M. Ni, Y. Liu, Y. Cho Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," in *Proc. 1st IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2003, pp. 407–415.

[13] B. Kempke, P. Pannuto, B. Campbell, and P. Dutta, "Surepoint: Exploiting ultra wideband flooding and diversity to provide robust, scalable, high-fidelity indoor localization," in *Proc. 14th ACM Conf. Embedded Netw. Sensor Syst. (CD-ROM)*, 2016, pp. 137–149.

[14] A. B. M. M. Rahman, T. Li, and Y. Wang, "Recent advances in indoor localization via visible lights: A survey," *Sensors*, vol. 20, no. 5, p. 1382, Mar. 2020.

[15] Y. Sun, J. Chen, C. Yuen, and S. Rahardja, "Indoor sound source localization with probabilistic neural network," *IEEE Trans. Ind. Electron.*, vol. 65, no. 8, pp. 6403–6413, Aug. 2018.

[16] W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao, and A. Crivello, "Indoor positioning based on fingerprint-image and deep learning," *IEEE Access*, vol. 6, pp. 74699–74712, 2018.

[17] B. T. Fang, "Trilateration and extension to global positioning system navigation," *J. Guid., Control, Dyn.*, vol. 9, no. 6, pp. 715–717, Nov. 1986.

[18] G. Han, J. Jiang, C. Zhang, T. Q. Duong, M. Guizani, and G. Karagiannidis, "A survey on mobile anchor node assisted localization in wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2220–2243, 3rd Quart., 2016.

[19] D. Han, S. Jung, M. Lee, and G. Yoon, "Building a practical Wi-Fi-based indoor navigation system," *IEEE Pervasive Comput.*, vol. 13, no. 2, pp. 72–79, Apr./Jun. 2014.

[20] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "SpotFi: Decimeter level localization using WiFi," in *Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM)*, 2015, pp. 269–282.

[21] N. Tadayon, M. T. Rahman, S. Han, S. Valaee, and W. Yu, "Decimeter ranging with channel state information," *IEEE Trans. Wireless Commun.*, vol. 18, no. 7, pp. 3453–3468, Jul. 2019.

[22] L. Wu, C.-H. Chen, and Q. Zhang, "A mobile positioning method based on deep learning techniques," *Electronics*, vol. 8, no. 1, p. 59, Jan. 2019.

[23] F. Xiao, W. Liu, Z. Li, L. Chen, and R. Wang, "Noise-tolerant wireless sensor networks localization via multinorms regularized matrix completion," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2409–2419, Mar. 2018.

[24] W. Njima, R. Zayani, I. Ahriz, M. Terre, and R. Bouallegue, "Beyond stochastic gradient descent for matrix completion based indoor localization," *Appl. Sci.*, vol. 9, no. 12, p. 2414, Jun. 2019.

[25] Z. Ning, K. Zhang, X. Wang, L. Guo, X. Hu, J. Huang, B. Hu, and R. Y. K. Kwok, "Intelligent edge computing in Internet of vehicles: A joint computation offloading and caching solution," *IEEE Trans. Intell. Transp. Syst.*, early access, Jun. 5, 2020, doi: 10.1109/TITS.2020.2997832.

[26] Z. Ning, R. Y. K. Kwok, K. Zhang, X. Wang, M. S. Obaidat, L. Guo, X. Hu, B. Hu, Y. Guo, and B. Sadoun, "Joint computing and caching in 5G-envisioned Internet of vehicles: A deep reinforcement learning-based traffic control system," *IEEE Trans. Intell. Transp. Syst.*, early access, Feb. 5, 2020, doi: 10.1109/TITS.2020.2970276.

[27] X. Wang, Z. Ning, S. Guo, and L. Wang, "Imitation learning enabled task scheduling for online vehicular edge computing," *IEEE Trans. Mobile Comput.*, early access, Jul. 28, 2020, doi: 10.1109/TMC.2020.3012509.

[28] R. S. Sinha and S.-H. Hwang, "Comparison of CNN applications for RSSI-based fingerprint indoor localization," *Electronics*, vol. 8, no. 9, p. 989, Sep. 2019.

[29] R. S. Sinha and S.-H. Hwang, "Improved RSSI-based data augmentation technique for fingerprint indoor localisation," *Electronics*, vol. 9, no. 5, p. 851, May 2020.

[30] S. Kim, L. T. Nguyen, and B. Shim, "Deep neural network based matrix completion for Internet of Things network localization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3427–3431.

[31] F. Strub, R. Gaudel, and J. Mary, "Hybrid recommender system based on autoencoders," in *Proc. 1st Workshop Deep Learn. Recommender Syst. (DLRS)*, 2016, pp. 11–16.

[32] P. Yazdanian and V. Pourahmadi, "DeepPos: Deep supervised autoencoder network for CSI based indoor localization," 2018, *arXiv:1811.12182*. [Online]. Available: http://arxiv.org/abs/1811.12182

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[34] P. Barsocchi, A. Crivello, D. La Rosa, and F. Palumbo, "A multisource and multivariate dataset for indoor localization methods based on WLAN and geo-magnetic field fingerprinting," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Oct. 2016, pp. 1–8.

[35] S. Nurmaini, A. Darmawahyuni, S. Mukti, A. Noviar, M. N. Rachmatullah, F. Firdaus, and B. Tutuko, "Deep learning-based stacked denoising and autoencoder for ECG heartbeat classification," *Electronics*, vol. 9, no. 1, p. 135, 2020.

[36] S. Ghosh and P. O. Kristensson, "Neural networks for text correction and completion in keyboard decoding," 2017, *arXiv:1709.06429*. [Online]. Available: http://arxiv.org/abs/1709.06429

[37] J. Fan and T. Chow, "Deep learning based matrix completion," *Neurocomputing*, vol. 266, pp. 540–549, Nov. 2017.

[38] R. Wang, Z. Li, H. Luo, F. Zhao, W. Shao, and Q. Wang, "A robust Wi-Fi fingerprint positioning algorithm using stacked denoising autoencoder and multi-layer perceptron," *Remote Sens.*, vol. 11, no. 11, p. 1293, May 2019.

[39] H. Zou, C.-L. Chen, M. Li, J. Yang, Y. Zhou, L. Xie, and C. J. Spanos, "Adversarial learning-enabled automatic WiFi indoor radio map construction and adaptation with mobile robot," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 6946–6954, Aug. 2020.

[40] C. Zhu, L. Xu, X.-Y. Liu, and F. Qian, "Tensor-generative adversarial network with two-dimensional sparse coding: Application to real-time indoor localization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[41] T. Nguyen and Y. Shin, "Matrix completion optimization for localization in wireless sensor networks for intelligent IoT," *Sensors*, vol. 16, no. 5, p. 722, May 2016.

[42] X. Guo, L. Chu, and X. Sun, "Accurate localization of multiple sources using semidefinite programming based on incomplete range matrix," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5319–5324, Jul. 2016.

[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[44] Z. Yang and Y. Liu, "Quality of trilateration: Confidence-based iterative localization," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 631–640, May 2010.

WAFA NJIMA (Member, IEEE) was born in Tunisia, in 1991. She received the engineer's degree from the Institut National des Sciences Appliquées et de Technologies de Tunis, in 2015, and the Ph.D. degree in the field of radio-communications from the Conservatoire National des Arts et Métiers in Paris, in 2019, in collaboration with the Ecole Supérieure des Communications de Tunis, under the supervision of Prof. Michel Terré and Prof. Ridha Bouallegue. She is currently a Temporary Assistant Professor and Researcher at the ETIS Laboratory, ENSEA. Her research interests are in signal processing, wireless communications, sparse data, indoor localization, and machine learning for communications.

MARWA CHAFII (Member, IEEE) received the master's degree in the field of advanced wireless communication systems (SAR) and the Ph.D. degree in telecommunications from Centrale Supélec, France, in 2013 and 2016, respectively. From 2014 and 2016, she has been a Visiting Researcher at the Poznan University of Technology, Poland; University of York, U.K.; Yokohama National University, Japan; and the University of Oxford, U.K. She joined the Vodafone Chair Mobile Communication Systems at the Technical University of Dresden, Germany, in February 2018, as a Research Group Leader. Since September 2018, she has been a Research Projects Lead at Women in AI and an Associate Professor at ENSEA, France, where she holds a Chair of Excellence from CY Paris Initiative. She received the 2018 Ph.D. prize in the field of Signal, Image & Vision in France. She has been listed under the top 10 Rising Stars at N2Women. Her research interests include signal processing for digital communications, advanced waveform design for wireless communications, and machine learning for communications. She is the Vice-Chair of the IEEE ComSoc Emerging Technology Initiative on Machine Learning for Communications. She is also currently managing the Gender Committee of the AI4EU Community. She currently serves as an Associate Editor for the IEEE COMMUNICATIONS LETTERS.

AHMAD NIMR (Member, IEEE) received the Diploma degree in communication engineering, in 2004, and the M.Sc. degree in communications and signal processing from TU Ilmenau, in 2014. He worked as a Software and Hardware Developer from 2005 to 2011. He has been a member of the Vodafone Chair at TU Dresden since October 2015. His research activities focus on multicarrier waveforms and multiple access techniques. He is also involved in the design and implementation of real-time communication systems. He received the Best Graduate Student Award for his excellent master's grades.

GERHARD FETTWEIS (Fellow, IEEE) received the Ph.D. degree from RWTH Aachen, in 1990, under the supervision of H. Meyr. After one year at IBM Research, San Jose, CA, USA, he moved to TCSI Inc., Berkeley, CA, USA. He coordinates the 5G Lab Germany, and two German Science Foundation (DFG) centers at TU Dresden, i.e., cfaed and HAEC. He has been a Vodafone Chair Professor at TU Dresden since 1994, and heads the Barkhausen Institute since 2018. His research focusses on wireless transmission and chip design for wireless/IoT platforms, with 20 companies from Asia/Europe/US sponsoring his research. He is a member of the German Academy of Sciences (Leopoldina), the German Academy of Engineering (acatech), and received multiple IEEE recognitions as well has the VDE ring of honor. In Dresden, his team has spun-out sixteen start-ups, and setup funded projects in volume of close to EUR 1/2 billion. He co-chairs the IEEE 5G Initiative, and has helped organizing IEEE conferences, most notably as the TPC Chair of ICC 2009 and of TTM 2012, and as the General Chair of VTC Spring 2013 and DATE 2014.