Energy efficient data gathering schema for Wireless Sensor Network: A Matrix Completion based approach

Manel Kortas, Oussama Habachi, Ammar Bouallegue, Vahid Meghdadi, Tahar Ezzedine, Jean-Pierre Cances

Energy efficient data gathering schema for Wireless Sensor Network: A Matrix Completion based approach

Manel Kortas*†, Oussama Habachi†, Ammar Bouallegue*, Vahid Meghdadi†, Tahar Ezzedine *
and Jean-Pierre Cances†
*Communications Systems Laboratory, National Engineering School of Tunis,
University of Tunis El Manar, Tunisia
†Univ. Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France
manel.kortas@unilim.fr, oussama.habachi@xlim.fr, ammar.bouallegue@enit.rnu.tn,
meghdadi@ensil.unilim.fr, Tahar.Ezzedine@enit.rnu.tn and cances@ensil.unilim.fr

***Abstract*—In this paper, we seek to address the data gathering in the continually growing Wireless Sensor Networks (WSNs) with the intention to save the nodes' energy. In order to address usual WSN problems, such as data losses, collisions and re-transmissions, a twofold data compression pattern is proposed. We consider that a restricted number of sensor nodes are selected to be active and represent the whole network, while the rest of nodes remain idle and do not participate at all in the data sensing and transmission. Furthermore, the set of active nodes' readings is efficiently reduced, in each time slot, according to the cluster scheduling. Relying on the existing Matrix Completion (MC) techniques, the sink node is unable to recover the entire data matrix due to the existence of completely empty rows that correspond to the inactive nodes, which can be considered as absent nodes for a very long period, or nodes that do not exist at all. Thereby, we propose a complementary interpolation technique, based on a minimization problem that benefits from sensor nodes inter-correlation, to guarantee the reconstruction of all the empty rows, despite their large number. The simulations confirm the efficiency of the proposed approach and show that it outperforms the existing one by up to 70.101% of Normalized Mean Absolute Error on all missed elements, when the number of active nodes is of about 10% of the total number of sensor nodes.**

I. Introduction

During the last decades, the Internet of Things (IoT) has emerged as a new business paradigm composed of billions of devices that can communicate with each other. Thus, it has gained much attention in both the industry and the scientific community. Yet, the integration of the IoT into the fifth generation cellular systems (5G) and their evolution still represent a formidable technical challenge due to the large number of sensor nodes and generated data. However, this is a burdensome task since the wireless resources as well as sensors' capabilities are limited. A motivating proposal, Compressive Sensing (CS), has been proposed to reduce the number of active agents at a given time slot, while remaining capable to recover the missing data [1]. Generally, Wireless Sensor Networks (WSNs) consist of a large set of sensor nodes that are self-organising and geographically distributed across the network area. In most cases, these devices operate in an unattended mode. Then, they are unable to renew their batteries. Therefore, energy efficiency is of prime importance in these networks. Indeed, reducing the number of transmitting sensors, using methods such as CS, is not only useful to avoide the collisions but also crucial for sensors who need to sleep to prolong their lifetimes. Over the past years, a plenty of works has managed the data gathering problems in wireless networks by the integration of the CS technique, which made attractive progress in the network energy consumption [1]-[4]. Recently, it has been proven that the integration of Matrix Completion (MC), as an extension of CS, has significantly enhanced WSNs' performances. Since MC treats the data in its matrix form, it can fully capture the signal correlation in both time and space, and hence achieves a good interpolation quality with a higher compression ratio (fewer delivered readings) [4]. Therefrom, many researches about data gathering schemes based on MC theory have been introduced [4]-[9].

In some applications, especially the densely deployed WSNs, nodes that are monitoring the same geographic region can be arranged into cluster to enhance the network management. Moreover, the sampled data is in general highly correlated between nodes that belong to the same cluster. Indeed, gathering raw data from all cluster nodes becomes wasteful for the energy and thus inefficient. Therefore, in this paper, we suppose that only a sub-set of nodes will be selected from each cluster to be the representative of the whole network. These active sensor nodes send their readings to the sink under a sampling ratio guaranteed by the MC theory, while the rest of sensor nodes remain silent and do not contribute in the data sensing and transmission operations. By

way of explanation, we propose a twofold compression pattern. First, we suppose that a part of nodes does not sense the environment at all. We can proceed as if these inactive nodes are inexistent or absent for a very long period. The second compression level is that, at each time slot, only a sub-set of the active nodes, called transmitting nodes, deliver their sensing data to the sink. However, this atypical high-loss scenario leads to a large number of empty rows in the received data matrix, which totally disagrees with the MC fundamentals (a row is called empty if and only if sensing data are missing over all the time slots, which corresponds to an inactive node). Indeed, since MC schemes are based on the minimization of the matrix rank, they become useless when there is any empty column or empty row in the matrix [10]. In the state-of-art of MC-based approaches for the WSNs, to the best of our knowledge, [11] is the only paper who dealt with the case where there are some missed rows by applying a spatial pre-interpolation technique, which recovers data from neighboring nodes. Nevertheless, as the number of empty rows (inactive nodes) gets bigger, we face with absent nodes having themselves absent neighbor nodes as well. Subsequently, this framework becomes unable to recover the data rows of these $isolated$ nodes. Even though this method is interesting, it seems not to be well suited for the addressed scenario and fails to take into consideration the existence of the $isolated$ nodes (absent nodes whose all neighbors are absent too). In fact, they basically focused on the case of MC reconstruction with the existence of successive missed or/and corrupted data, and treating a considerable number of empty rows was out of the scope of their work. In this context, we develop our approach, which, firstly, schedules the compression pattern after efficiently clustering the nodes and identifying the representative ones. Secondly, it addresses the case of high data loss ratios with a significant number of inactive sensors (empty rows) using a sequence of three different interpolation techniques.

The paper is organized as follows. The next section provides a brief overview on the MC theory. Section III introduces the system model. We present, in section IV, how to efficiently identify the clusters as well as the data sensing and transmission schedule. Section V is dedicated to the reconstruction framework. Before concluding the paper in section VII, we illustrate in section VI the performance of the proposed approach.

II. Overview of Matrix Completion

Recently, MC technique has emerged to benefit from the signal low-rank feature in order to fill the missing data using a limited number of matrix entries [12]. That is, a partially unknown matrix $M \in \mathbb{R}^{N\times T}$ of rank $r \ll min\{N,T\}$ can be entirely recovered, if a sub-set of its entries M_{ij} as well as their indices $(i,j) \in \Omega$ are known by the receiver. The entry-wise partial observation operator $P_\Omega : \mathbb{R}^{N\times T} \rightarrow \mathbb{R}^{N\times T}$ is defined as follows:

$$[P_\Omega(X)]_{ij} = \begin{cases} X_{ij} & (i,j) \in \Omega \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

According to [12], if Ω holds enough information and if M is a low rank matrix, we can recover the unknown entries by solving the following rank minimization problem:

$$\text{minimize} \quad rank(X) \quad s.t \quad P_\Omega(X) = P_\Omega(M). \quad (2)$$

However, problem (2) is not convex, and algorithms that can solve it are doubly exponential. Fortunately, the nuclear norm $\| X \|_*$ minimization problem, which is a convex relaxation, can be solved. Indeed, it is used as an alternative to the NP-hard rank minimization problem. Hence, we have:

$$\text{minimize} \quad \| X \|_* \quad s.t \quad P_\Omega(X) = P_\Omega(M). \quad (3)$$

In the literature, several solvers for this type of systems have been proposed. For example, the Singular Value Thresholding (SVT), which optimizes an approximation of (3) by adding a Frobenius-norm term to the objective function [13]:

$$\begin{aligned} \text{minimize} \quad & \tau \| X \|_* + \frac{1}{2} \| X \|_F^2 \\ s.t \quad & P_\Omega(X) = P_\Omega(M). \end{aligned} \quad (4)$$

Low rank matrix fitting (LMaFit) [14], Sparsity Regularized SVD (SRSVD) and Sparsity Regularized Matrix Factorization (SRMF) [5], among other schemes, have used the matrix factorization method. Different from (3), matrix factorization technique has been suggested to replace (2) rather than the nuclear norm.

III. System Model

Consider a WSN that consists of a set $\mathcal{N} = \{1,\dots,N\}$ of N sensor nodes. Let $X \in \mathbb{R}^{N\times T}$ denote the data matrix that holds measurements gathered by the set $\mathcal{N}$ during a sensing period of length T time slots. Precisely, the entry $x_{i,t}$ of X represents the t^{th} data reading sensed by the i^{th} node. The considered scenario targets to estimate the full nodes' readings, X, using a small sub-set $\mathcal{N}_{rep} = \{1,\dots,N_{rep} \ll N\}$ of active nodes, referred to as representative nodes. It is noteworthy that the number of active nodes is relatively small compared to the number of the inactive ones. Specifically, reducing the number N_{rep} generates a set of absent nodes, which have also all their neighbors absent as well. These nodes are denoted by $isolated$ nodes (IS).

To figure out how would be the performance of the proposed approach, we generate a synthetic signal that is composed of different Gaussians, each of which presents a portion of the whole controlled geographic area. Each portion of the signal is correlated in time and space, where the temporal correlation as well as the spatial

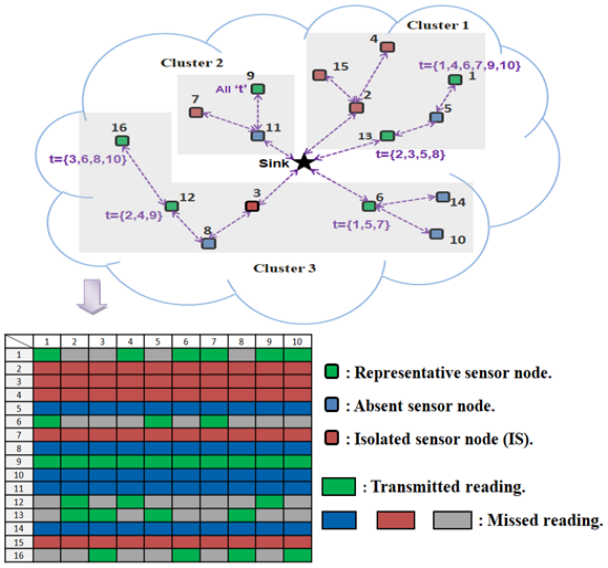

Fig. 1. An illustrative miniature WSN with the resulting delivered data matrix M.

correlation parameters differ from one portion to another. The performed signal model was inspired from [15], which has introduced the solution of reproducing a signal retaining the behavior of a given real world data by adjusting the correlations' parameters. The number of the generated Gaussians as well as their distribution on the field can be defined according to the kind of the signal one wants to reproduce.

Figure 1 depicts an example of a WSN composed of $N = 16$ nodes, among which $N_{rep} = 6$ nodes are chosen to be active. The proposed reconstruction scheme aims to fill all the missed entries that correspond to the non-delivered readings.

IV. Sampling Pattern

A. Clusters Detection

In this part, we investigate the partition of the deployed sensor nodes into J clusters. The main reason for clustering the network is to involve all the detected clusters in the data sensing and transmission. Usually, in conventional MC, the transmitting nodes are chosen randomly during the T time slots. This kind of selection can disregard nodes that belong to the small clusters, which deteriorates the reconstruction process. If we make all the clusters participate in the data sampling process, we fortify the diversity in the sent data set. Thus, for each time slot t, according to a given sampling ratio and using the same percentage, a set of sensors is selected from each cluster to constitute the sensing and transmission schedule. To do so, in this subsection, we aim to partition nodes into different clusters having different readings, when in the same group nodes have similar readings, i.e. we seek for minimizing the inter-cluster similarities and maximizing the intra-cluster similarities. Such an efficient grouping can be realized using the Normalized Spectral Clustering (SPC) [16]. We propose to perform the algorithm of Ng, Jordan and Weiss, whose steps are detailed in [17].

We suppose that the whole network is organized as follows: $\mathcal{N} = \bigcup_{j=1}^{J} CL_j$ and $N = \sum_{j=1}^{J} cl_j$, where $cl_j = |CL_j|$, CL_j is the cluster j and cl_j is the number of nodes that belong to CL_j. To cluster the nodes, the sink relies on their received readings[1] and considers the set of data vectors, $\chi_{init} = \{x_{init\,1}^T, x_{init\,2}^T, \ldots, x_{init\,N}^T\}$. $x_{init\,i} \in \mathbb{R}^{1\times T_{init}}$ denotes a T_{init}-dimensional data points, containing the readings sent by node i during the learning period. The SPC technique performs data clustering and considers it as a graph partitioning problem. It transforms the set χ_{init} into a weighted graph $G = (V, E)$ using a similarity matrix $A \in \mathbb{R}^{N\times N}$, where each vertex v_i represents $x_{init\,i}$, and each edge between two vertices v_j and v_i represents the similarity $a_{j,i} \geq 0$. In this work, we opted for the Gaussian kernel to measure the similarities between the data points $\{x_{init\,i}\}$ [17].

Commonly, identifying the number J of clusters in an optimal way is the main concern of all the clustering algorithms. In this work, we apply the eigengap heuristic that determines J after finding a drop in the magnitude of the Laplacian eigenvalues, $\{\lambda_1, \lambda_2, \ldots, \lambda_N\}$, computed according to the used SPC technique [16]. That is:

$$J = arg\ \max_i(\lambda_{i+1} - \lambda_i). \tag{5}$$

B. Sensing and Transmission Schedule

In this part, we illustrate how we take into consideration the detected clusters in the representative sensor node selection as well as in the transmission schedule.

In order to cover all the detected clusters, $\mathcal{N}_{rep}$, the set of the representative nodes, consists of the combination of J sub-sets, $(\mathcal{N}_{rep_j})_{j=1,\ldots,J}$, where $\mathcal{N}_{rep_j}$ includes N_{rep_j} representative nodes randomly selected from cluster CL_j using the same shared percentage pct_{Nrep}. That is:

$$N_{rep} = \sum_{j=1}^{J} N_{rep_j},\ \text{where}\ N_{rep_j} = pct_{Nrep}\% \times cl_j. \tag{6}$$

In (6), if $pct_{Nrep}\% \times cl_j$ is not an integer, we round it to the nearest integer greater than or equal to the value of that element. Given the example of Figure 1, we can note the existence of three clusters within the network. We assume that $pct_{Nrep} = 30$. Hence, 30% of sensors will be selected from each cluster to be active. That is to say that we should select $N_{rep_1} = 2$ nodes from CL_1, $N_{rep_2} = 1$ node from CL_2 and $N_{rep_3} = 3$ nodes from

[1] At the initialization, all the sensor nodes transmit their information during a short learning period $T_{init} \ll T$.

CL_3. That is, in total we have $N_{rep} = 6$ representative nodes.

Once the set $\mathcal{N}_{rep}$ of representative nodes is assigned, the sink moves to the sensing and transmitting schedule $\Omega_M \in \mathbb{R}^{N\times T}$, that is, $\Omega_{M(i,t)} = 1$ if $x_{i,t}$ will be sensed and 0 otherwise. Thus, the incomplete received data matrix $M \in \mathbb{R}^{N\times T}$ can be expressed as a dot product between X and Ω_M.

The representative nodes do not send their raw data to the sink. Instead, they trade on the data sensing and transmission along the T time slots and deliver a part of their readings, that is, $m < N_{rep}$ rather than N_{rep} readings per time slot. Therefrom, the sink assigns m transmitting nodes for each time slot t by picking them from the sub-set $\mathcal{N}_{rep}$. As it has been previously stated, in order to ensure the diversity in the transmitted data, the m transmitting nodes are selected randomly, with the same percentage pct_m, that is, m_j sensors from each sub-set $\mathcal{N}_{rep_j}$. Likewise (6) we have:

$$m = \sum_{j=1}^{J} m_j, \text{ where } m_j = pct_m\% \times N_{rep_j}. \quad (7)$$

Let us focus again on the example of Figure 1, we assume that $pct_m = 20$. Hence, for each t, 20% of nodes from each sub-set $\mathcal{N}_{rep_j}$ are randomly designated to transmit their readings to the sink. Since the used number N of this example is very small, we end with $m_j = 1$ transmitting node from each cluster, for each time slot t. Note that without enforcing the involvement of all the clusters in the data sensing and transmission process, cluster 2 that contains only sensor 9, could be totally ignored.

V. Reconstruction Pattern

In this section, we focus on how to estimate the entire data matrix $X \in \mathbb{R}^{N\times T}$ based on the limited amount of received readings. Isolating $(N - N_{rep})$ inactive sensor nodes from the sampling schedule entails the existence of $(N - N_{rep})$ totally empty rows in the received data matrix M, which impedes the MC technique and makes it completely unable to recover the original matrix. Thus, the use of other complementary interpolation techniques becomes needful. In this context, we develop a structured MC-based reconstruction algorithm that is able to ensure the recovery of the entire data matrix X.

Stage 1: Obviously, it is not possible to directly apply the MC method with the existence of the empty rows. Thus, firstly, we remove these rows from M. We denote the resultant matrix as $M_{MC} \in \mathbb{R}^{N_{rep}\times T}$ that contains the partially received readings of the active sensor nodes. We carry on with the same removal from Ω_M to get $\Omega_{MC} \in \mathbb{R}^{N_{rep}\times T}$. Then, using (4) or any other method proposed for the MC resolution, we fill the missed entries of M_{MC} that correspond to the non-delivered readings of the N_{rep} nodes. We denote $X' \in \mathbb{R}^{N_{rep}\times T}$ as the MC based estimation data. Finally, we update $X' \in \mathbb{R}^{N\times T}$ by adding the $(N - N_{rep})$ empty rows and placing them in their proper corresponding locations of M. It is noteworthy that the MC, as the first step in the reconstruction process, is an important part since the performance of the subsequent interpolation techniques depends on the recovery accuracy of the MC.

Stage 2: After filling the random missed readings, remain the $(N - N_{rep})$ fully empty rows corresponding to the inactive nodes. In this stage, we carried on with the spatial pre-interpolation method of [11], which estimates the data of an empty row using the available data of the neighboring nodes. They used an $N \times N$ binary symmetric matrix Y, where both rows and columns denote nodes. The sink assigns 1 to $Y(i, j)$ if it finds that node i and node j are 1-hop neighbors. However, according to the signals nature that we consider, and to avoid untrustworthy data recovery, we assume that even though two nodes are geographically close to each other, if they don't belong to the same cluster, they are not considered as neighbors.

As stated before, the number N_{rep} of the representative nodes is very small compared the total number N, which means that the $(N - N_{rep})$ inactive nodes represent the preponderant portion of the network. Thereby, there are several IS nodes in the network (having all their neighbors absent). Using the stated topology matrix Y, this interpolation technique can reconstruct data only for the absent nodes, whose neighbors belong to $\mathcal{N}_{rep}$. We assume that the network distribution contains N_{Is} $isolated$ nodes. Hence, the resulting data matrix $X'' \in \mathbb{R}^{N\times T}$, obtained following this stage, still holds N_{Is} empty rows to be estimated.

For the detailed steps of the above interpolation technique, the reader may refer to [11].

Stage 3: Since the pre-interpolation method is limited to estimate only a part of the total empty rows (absent nodes), we resort to a complementary spatial interpolation to recover the remaining part of the empty rows ($isolated$ nodes). Taking advantage of the spatial dependency among the sensors, we fill the remaining empty rows by minimizing the following problem:

$$\text{minimize } (fac_1 \times \|\widehat{X} - X''\|_F^2 + fac_2 \times \|S \times \widehat{X}\|_F^2), \quad (8)$$

where $S \in \mathbb{R}^{N\times N}$ represents the spatial constraint matrix, fac_1 and fac_2 are two tuning parameters and $\widehat{X} \in \mathbb{R}^{N\times T}$ is the final interpolated data matrix. The resolution of this minimization problem can be easily achieved using the semidefinite programming. To solve (8) and get $\widehat{X}$, we opted for the CVX package [18], which is implemented in Matlab, as an advanced convex programming solver. In (8), the matrix S reflects our knowledge about the spatial structure inherent in the data, as it is computed based on the data matrix $X_{init} = [x_{init\,1}^T, x_{init\,2}^T, \ldots, x_{init\,N}^T]^T \in \mathbb{R}^{N\times T_{init}}$, corresponding to the learning period. The matrix S expresses the

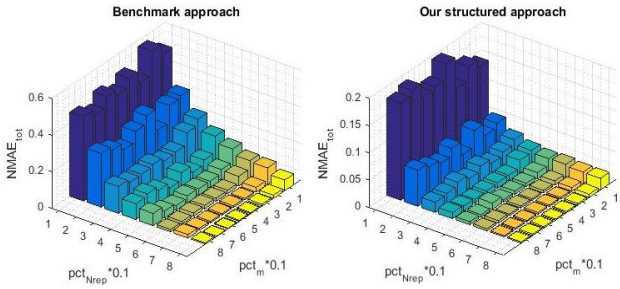

Fig. 2. $NMAE_{tot}$ for the proposed technique and for the benchmark.

inter-nodes' readings similarities. As a distance function, we used the Euclidean distance, computed in the nodes data domain, in order to model the similarity between the rows of X_{init}. Below are the steps to obtain S:

1-We start with an all-zeros matrix S.

2-For each row i of the learning data matrix X_{init}, we search for the set j_i' of indexes containing the K closest rows to i, that is, $j_i' = \{j_k \neq i \mid k = 1, ..., K\}$.

3-Assuming that the row i can be estimated using the linear combination of the rows of set j_i', we perform the linear regression to get the weight vector $W = [w_1, \ldots, w_K] \in \mathbb{R}^{1\times K}$ through the following equation:

$$W = X_{init}(i,:)X_{init}(j_i',:)^T[X_{init}(j_i',:)X_{init}(j_i',:)^T]^{-1}. \quad (9)$$

4-Finally, we attribute $-w_k$ to $S(i, j_k)$ and 1 to $S(i, i)$.

Once the above steps have been achieved for all rows i, we get the matrix S, with which we recover $\widehat{X}$.

Now, remains the last adjustment to perform, which is the scaling of the parameters, fac_1 and fac_2 of (8). These tuning parameters are introduced in order to make a tradeoff between a close fit to the estimated values of X'' and the intention of approximating the N_{Is} remaining empty rows. By running a large number of simulations, we found that adjusting these parameters nicely improves the reconstruction performance, and the founded values of fac_1 and fac_2 are independent of the size of the matrix (N and T) as well as the Gaussians' values composing the synthetic signal.

Focusing once again on the example of Figure 1. The dotted lines refer to the neighborhood relation between the nodes. As we can note, the nodes $\{5, 8, 10, 11, 14\}$ are each linked at least to a representative node. Thus, their data readings can be easily estimated using the spatial pre-interpolation technique of stage 2. Whereas, the nodes $\{2, 3, 4, 7, 15\}$ are considered as $isolated$ from the network. Thus, their readings are recovered thanks to the minimization (8) of stage 3.

VI. Numerical Results

In this section, we compare the data recovery performance of our proposed approach to that of a benchmark one, which was implemented basically on what was proposed in [11] and in line with our scenario requirements. In fact, at the end of their paper, Xie et al. assumed in [11] that there are few empty rows in M, that is, for $N = 196$, only 14 rows was empty, namely 7% of N (i.e. 93% of N of representative nodes). As we have already mentioned at the beginning of this paper, dealing with a large number of empty rows has not been the main focus of their work. Thereby, their approach hasn't taken into consideration the existence of the IS in the network. Yet, to the best of our knowledge, this is the unique paper that has dealt with a similar scenario using MC, and with which we can compare our approach. To measure the recovery error, we opted for the following metrics, where X and $\widehat{X}$ represent respectively the raw data matrix before compression and the finally recovered one:

1-$NMAE_{tot}$: The Normalized Mean Absolute Error on all missed elements:

$$NMAE_{tot} = \frac{\sum_{i,t:\Omega_M(i,t)=0} |X(i,t) - \widehat{X}(i,t)|}{\sum_{i,t:\Omega_M(i,t)=0} |X(i,t)|}. \quad (10)$$

2-$NMAE_{MC}$: The Normalized Mean Absolute Error on the partially missed elements that correspond to the non-delivered data readings of the N_{rep} representative nodes, where Ω_{mc} is the index set of the partially missed elements found in the received data matrix M:

$$NMAE_{MC} = \frac{\sum_{i,t:(i,t)\in\Omega_{mc}} |X(i,t) - \widehat{X}(i,t)|}{\sum_{i,t:(i,t)\in\Omega_{mc}} |X(i,t)|}. \quad (11)$$

3-$NMAE_{ER}$: The Normalized Mean Absolute Error on the missed data of the totally empty rows that correspond to the inactive nodes' readings, where Ω_{ER} is the index set of the $(N - N_{rep})$ empty rows found in the received data matrix M:

$$NMAE_{ER} = \frac{\sum_{i,t:i\in\Omega_{ER}} |X(i,t) - \widehat{X}(i,t)|}{\sum_{i,t:i\in\Omega_{ER}} |X(i,t)|}. \quad (12)$$

4-CR: The Compression Ratio, where $\mid \Omega \mid$ presents the number of received readings, whereas, $(N \times T)$ presents the total number of elements in X:

$$CR = \frac{N \times T - \mid \Omega \mid}{N \times T}. \quad (13)$$

To evaluate the proposed approach under different CRs, we vary pct_{Nrep} from 10 to 80, and for each given pct_{Nrep}, we vary pct_m from 10 to 80. It is evident that the range of the values of CR depends on the value of pct_{Nrep}. The higher pct_{Nrep}, the larger CR range can be used. Note that we are mostly interested in the small values of pct_{Nrep} and pct_m, as we are treating a high loss scenario.

For the network parameters, we consider that $N = 50$ sensor nodes are randomly deployed in a square observation area of size $100m \times 100m$, and we monitor the WSN throughout $T = 100$ time slots.

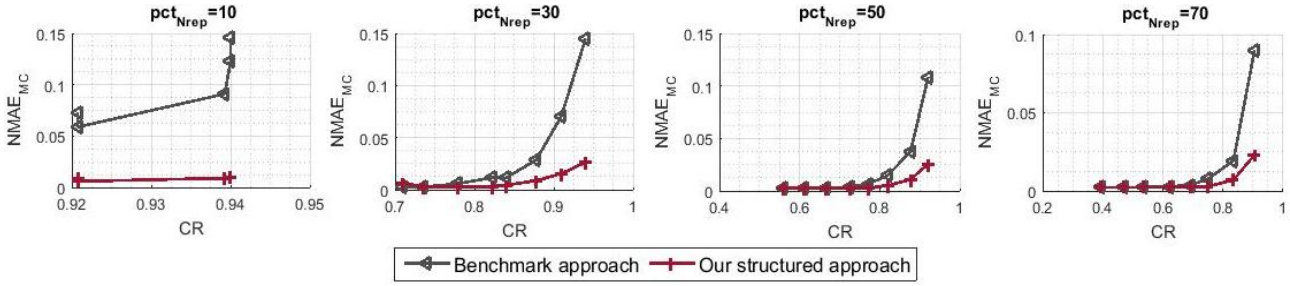

Fig. 3. $NMAE_{MC}$ for the proposed technique and for the benchmark.

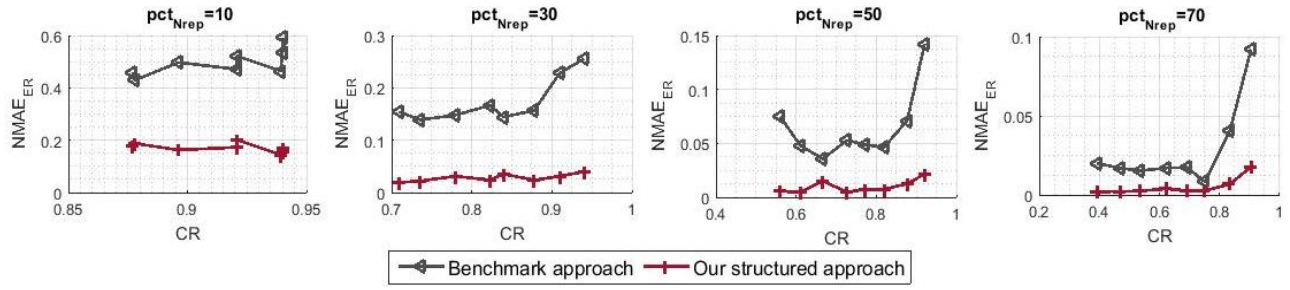

Fig. 4. $NMAE_{ER}$ for the proposed technique and for the benchmark.

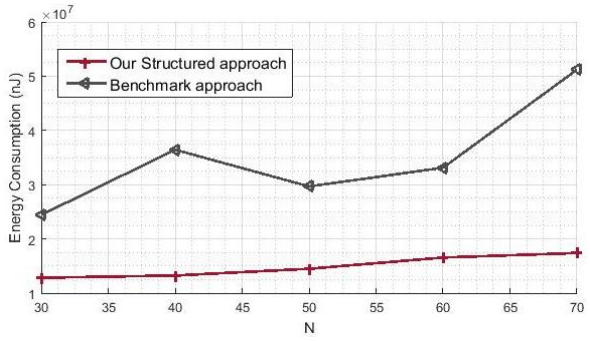

Fig. 5. Energy consumption for the proposed technique and for the benchmark.

To begin, we implement a benchmark approach based on what was proposed in [11]. The compression pattern of this scheme consists in selecting the set $\mathcal{N}_{rep}$ of representative nodes in a purely random way, which is the same as randomly choosing the empty rows. Likewise, for each time slot t, m sensor nodes are uniformly picked from the set $\mathcal{N}_{rep}$ to transmit their data readings to the sink. Here, neither the selection of the representative nodes nor the selection of the transmitting ones takes into consideration the detected clusters. Regarding the reconstruction pattern, to obtain the final reconstructed data matrix $\widehat{X}$, this approach applies the MC resolution then the spatial pre-interpolation. The temporal pre-interpolation was omitted here as we don't consider the existence of empty columns in the observed data matrix M. This is not the case with our scenario as, at every time slot t, we ensure the transmission of m data readings collected from different m locations. As we can notice from the 3-D bar graph of Figure 2, our proposed approach distinctly outperforms the benchmark one for all the pct_m values, and for different values of pct_{Nrep}. We are able to go up to 90% of N of missing rows ($pct_{Nrep} = 10$) with an attractive recovery performance, $NMAE_{tot}$ of $[0.14, 0.18]$. Whereas, the benchmark approach yields an $NMAE_{tot}$ of $[0.40, 0.56]$.

To evaluate separately the benefits of each building block of the proposed approach, we have disassembled the error ratios, and we have firstly measured, in Figure 3, the $NMAE_{MC}$ with the variation of CR. Since the benchmark approach proceeds regardless the existence of the different clusters, it cannot provide an equitable representation of the different regions composing the whole network, and sensor nodes that belong to small clusters can be totally ignored. Although both compared approaches apply the same MC resolution method, the $NMAE_{MC}$ of our approach is much lower than that of the benchmark, especially for the high CR values. This simulation shows how curiously interesting the clusters consideration is. For example, we can reach an improvement of 90.11% for ($pct_{Nrep} = 10$, $pct_m = 30$) (one passes from 0.091 to 0.009) and 82.069% for ($pct_{Nrep} = 30$, $pct_m = 10$) (one passes from 0.145 to 0.026).

For the convenience of comparison, Figure 4 highlights the error ratios on the recovery of the fully missed readings. Noticeably, we can detect a considerable gap in terms of $NMAE_{ER}$ between the curves of Figure 4. This difference across the entire ranges of CR, comes from the non-recovered readings of the IS with the benchmark approach. For example, we can reduce the recovery error of the empty rows up to 69.18% for ($pct_{Nrep} = 10$, $pct_m = 30$) and 84.706% for ($pct_{Nrep} = 30$, $pct_m = 10$). This simulation shows that the number N_{Is} of IS is significant for the high CRs. Therefore, adding a third interpolation technique, as the proposed minimization (8), becomes deeply needed. Otherwise, we are susceptible to end with a data matrix $\widehat{X}$, which is almost half built, even less.

Note that the proposed framework extremely minimizes the overall network energy consumption since we use a small set of active nodes for the data transmission. Moreover, compared to the benchmark approach, the proposed one can further improve the sensor nodes lifetime. Indeed, for a given $NMAE_{tot}$ target of 0.02 and $pct_{Nrep} = 60$, we measure the energy consumption during the T time slots for the both compared approaches depending on the number of nodes N. We consider that the energy consumption per emitted bit is $E_{bit} = 230nJ/bit$, and $E_{packet} = E_{bit} \times packet_{size}$ as the energy consumption per emitted packet. A sensor reading is of 16 bits, whereas the packet header size is fixed to 104 bits [2]. Figure 5 depicts the energy consumption for the proposed framework as well as for the benchmark one. It illustrates that our approach requires far less sensor nodes' readings, consequently much less energy consumption, to reach the same recovery performance. As a perspective, we can consider the residual energy of sensors when selecting the representative nodes as well as when assigning the sensing schedule. Accordingly, even though the energy consumption will not be reduced, the network lifetime can be extended.

VII. Conclusion

In this paper, we investigated how to tackle a challenging issue in the dense WSNs: how to isolate a significant number of sensor nodes from the monitoring area and let them remain idle throughout the whole sensing period. Then, relying on a MC-based approach, the sink approximates the missed readings using only the partially reported readings of the active sensor nodes (representative nodes). It is noteworthy that neatly identifying the clusters as well as the data sensing and transmission schedule deeply impact the recovery accuracy of the sensor nodes' readings. Moreover, by adding a minimization problem interpolation-based technique to the MC method, we succeeded to significantly improve, in comparison with the state-of-the-art method, the recovery of the inactive nodes' data readings. The simulation results confirm the efficiency of the proposed approach and show that it outperforms the existing one by up to 70.101% of $NMAE_{tot}$, when the number of active nodes is of about 10% of the total number of sensor nodes.

References

[1] M. Kortas, V. Meghdadi, A. Bouallegue, T. Ezzeddine, O. Habachi, and J.-P. Cances, "Routing aware space-time compressive sensing for wireless sensor networks," in *Personal, Indoor, and Mobile Radio Communications (PIMRC), 2017 IEEE 28th Annual International Symposium on*. IEEE, 2017, pp. 1–6.

[2] M. Hooshmand, M. Rossi, D. Zordan, and M. Zorzi, "Covariogram-based compressive sensing for environmental wireless sensor networks," *IEEE Sensors Journal*, vol. 16, no. 6, pp. 1716–1729, 2015.

[3] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, and S. Xiao, "Data gathering with compressive sensing in wireless sensor networks: a random walk based approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, 2015.

[4] M. Kortas, A. Bouallegue, T. Ezzeddine, V. Meghdadi, O. Habachi, and J.-P. Cances, "Compressive sensing and matrix completion in wireless sensor networks," in *Internet of Things, Embedded Systems and Communications (IINTEC), 2017 International Conference on*. IEEE, 2017, pp. 9–14.

[5] M. Roughan, Y. Zhang, W. Willinger, and L. Qiu, "Spatio-temporal compressive sensing and internet traffic matrices," *IEEE/ACM Transactions on Networking (ToN)*, vol. 20, no. 3, pp. 662–676, 2012.

[6] J. Cheng, Q. Ye, H. Jiang, D. Wang, and C. Wang, "Stcdg: An efficient data gathering algorithm based on matrix completion for wireless sensor networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 850–861, 2013.

[7] D. Wang, J. Wan, Z. Nie, Q. Zhang, and Z. Fei, "Efficient data gathering methods in wireless sensor networks using gbtr matrix completion," *Sensors*, vol. 16, no. 9, p. 1532, 2016.

[8] J. He, G. Sun, Z. Li, and Y. Zhang, "Compressive data gathering with low-rank constraints for wireless sensor networks," *Signal Processing*, vol. 131, pp. 73–76, 2017.

[9] K. Xie, L. Wang, X. Wang, G. Xie, and J. Wen, "Low cost and high accuracy data gathering in wsns with matrix completion," *IEEE Transactions on Mobile Computing*, vol. 17, no. 7, pp. 1595–1608, 2018.

[10] R. Du, C. Chen, B. Yang, N. Lu, X. Guan, and X. Shen, "Effective urban traffic monitoring by vehicular sensor networks," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 1, pp. 273–286, 2015.

[11] K. Xie, X. Ning, X. Wang, D. Xie, J. Cao, G. Xie, and J. Wen, "Recover corrupted data in sensor networks: A matrix completion solution," *IEEE Transactions on Mobile Computing*, vol. 16, no. 5, pp. 1434–1448, 2017.

[12] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational mathematics*, vol. 9, no. 6, p. 717, 2009.

[13] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[14] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Mathematical Programming Computation*, vol. 4, no. 4, pp. 333–361, 2012.

[15] D. Zordan, G. Quer, M. Zorzi, and M. Rossi, "Modeling and generation of space-time correlated signals for sensor network fields," in *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. IEEE, 2011, pp. 1–6.

[16] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[17] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[18] M. Grant and S. Boyd, "Cvx: Matlab software for disciplined convex programming, version 2.1. 2014 mar," http://www.cvxr.com/cvx,March2014.