Compressive Sensing Based Distributed Data Storage for Mobile Crowdsensing

SIWANG ZHOU, YI LIAN, DAIBO LIU, and HONGBO JIANG, Hunan University
YONGHE LIU, University of Texas at Arlington
KEQIN LI, State University of New York

Mobile crowdsensing systems typically operate centralized cloud storage management, and the environment data sensed by the participants are usually uploaded to certain central cloud servers. Instead, this article addresses the decentralized data storage problem in scenarios where cloud servers or network infrastructures do not work as expected and the sensing data have to be temporarily stored on the mobile devices carried by the participants. Considering that the sensing data are generally correlated, this article investigates a compressive distributed storage scheme for mobile crowdsensing. We notice a key observation: when a participant has a random walk in the target sensing area, his walking/sensing process can be considered as a random sampling for the entire area, although the activity of the participant may only have a local scope. We then propose an encoding algorithm based on compressive sensing theory. Each participant encodes the sensing data in their local trajectory, but the encoded CS measurement is capable of roughly reflecting the entire information of the whole area. While a participant stores a blurred global image of the target sensing area, the entire data can then be collaboratively stored by a certain number of participants. We further present a period-based data recovery algorithm to exploit the inter-period correlations, improving the recovery accuracy. Experimental results using real environmental data demonstrate the performance of the proposed compressive storage scheme. The test datasets and our source codes are available at https://github.com/siwangzhou/MCS-Storage.

CCS Concepts: • **Networks** → **Packet scheduling**; • **Human-centered computing** → **Ubiquitous and mobile computing design and evaluation methods;**

Additional Key Words and Phrases: Compressive sensing, distributed storage, mobile crowdsensing, wireless sensor network

ACM Reference format:
Siwang Zhou, Yi Lian, Daibo Liu, Hongbo Jiang, Yonghe Liu, and Keqin Li. 2022. Compressive Sensing Based Distributed Data Storage for Mobile Crowdsensing. *ACM Trans. Sen. Netw.* 18, 2, Article 25 (February 2022), 21 pages.
https://doi.org/10.1145/3498321

This work was supported in part by the National Science Foundation of China under grants 61902122 and 62172153, and Changsha Municipal Natural Science Foundation under grant kq2014057.

Authors' addresses: S. Zhou, Y. Lian, D. Liu (corresponding author), and H. Jiang, College of Computer Science and Electrical Engineering, Hunan University, Changsha, Hunan, China, 410082; emails: {swzhou, lisa_lian}@hnu.edu.cn, dbliu.sky@gmail.com, hongbojiang@hnu.edu.cn; Y. Liu, Department of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX; email: yonghe@cse.uta.edu; K. Li, Department of Computer Science, State University of New York, New Paltz, NY; email: lik@newpaltz.edu.

1550-4859/2022/02-ART25 $15.00
https://doi.org/10.1145/3498321

1 INTRODUCTION

Wireless sensor networks (WSNs) may be the most desirable way to collect environment information such as temperature, noise, air quality, and traffic condition, among others [13, 22]. However, deploying a large-scale WSN can be cost prohibitive. Taking Citysee [12], an air quality monitoring network, as an example, the organizer uses 100 dedicated sensors and 1,096 relay nodes in a block size of about 1 km^2 in Wuxi, a middle-sized city in China. If the network needs to be extended to the whole urban area of Wuxi city, totaling 560 km^2, one has to deploy at least 56,000 sensors and more than 0.6 million relay nodes to achieve a desirable coverage. And even worse, these nodes may fail unpredictably, risking data loss of the observation sites. **Mobile crowdsensing (MCS)**, along with the proliferation of various portable mobile devices with built-in sensors such as smartphones and wearables, is becoming an appealing paradigm for monitoring phenomena in the target sensing area [18, 29]. The involvement of the participants with their own mobile sensing devices is one of the most important differences between traditional WSNs and the MCS. This makes it possible to build environment monitoring systems without deploying dedicated sensor networks.

MCS systems typically employ centralized cloud-based data storage management [5]. The participants with portable sensing devices walk in the target area. The corresponding sensing data are uploaded to certain cloud servers via WiFi/4G/5G mobile network infrastructure. A number of MCS methods have been investigated to utilize the mobilities of the participants for data gathering, ensuring the cloud servers can obtain and store the complete information of the entire target sensing area [15, 28]. Unfortunately, in some scenarios, such as an earthquake or other unexpected events, network failures may occur, and the cloud servers may not receive the sensing data uploaded by the participants as usual.

Several decentralized storage methods for WSNs have recently been investigated (e.g., [11, 14, 21]), where sink nodes, functionally equivalent to the cloud servers in MCS systems, are not present and the sensing data have to be stored in individual sensors. In these methods, sensor readings are first disseminated over the network with various dissemination strategies. Each sensor can then receive the readings sensed by other nodes. In the process of receiving readings, the sensors encode the received readings and then store the encoded value to decrease the amount of data. Zhou et al. [37] further proposed a networked storage algorithm by employing **compressive sensing (CS)** theory to improve the data accuracy with less dissemination cost. However, crowdsensing is not a network in the conventional sense like WSNs. In MCS systems, the participants with sensing devices are perceived as the network nodes, and the sensing data are not easily disseminated among the participants due to their mobility and the randomness of their movements. Therefore, these WSN-related storage strategies are inapplicable to MCS systems.

In this context, we observe and introduce a significant problem of distributed data storage in MCS systems, where the participants are required to temporarily store their sensing data until the network infrastructure and the cloud servers are back up. However, when considering distributed storage in MCS systems, several challenges arise. First, the sensing data have huge data volumes, whereas mobile devices cannot store too much sensing data, since they are mostly privately owned and memory constrained. Second, predicting which sites a participant will arrive at is not practical without the support of central cloud servers. This means that one may not guarantee the full coverage in the absence of cloud severs, causing a number of monitoring holes. Third, moreover, some participants carrying the sensing data might leave the MCS system at any moment. In other words, the cloud servers need to be able to recover the entire dataset even if only a few participants contribute their data when the data communication and the clouds get back to normal.

In this article, we propose a distributed and compressive storage scheme for MCS. To the best of our knowledge, this is the first storage scheme for MCS systems without requiring a central cloud

server. To achieve this goal, we define a novel concept of the virtual sensor network to abstract a target sensing area so that we are capable of following the storage strategies in traditional WSNs and then investigate a new compressive distributed data storage scheme. With our scheme, the random movements of the participants are exploited to achieve full coverage in the encoding sense. Each participant stores an encoded CS measurement, and the complete data corresponding to the target area can then be recovered with the measurements stored by only a certain number of participants.

The contributions of this work are summarized in the following:

- We introduce a novel virtual sensor network model for abstracting the target sensing area without deploying dedicated sensors. In this model, observation sites are seen as virtual sensor nodes, and data communication among the virtual nodes is implemented utilizing the movement of the participants. In this way, we can follow the classical WSNs and finally give a new distributed storage scheme specially suited for crowdsensing.
- We present a robust compressive storage algorithm. Based on CS theory, the random local movement of a participant can be referred to as an encoding process for the entire sensing area. The partial data sensed by a single participant are encoded into a blurred CS measurement, which is capable of being considered as a globe image of the area. Each participant stores a blurred global image, and the entire dataset can then be recovered from a certain number of participants still surviving after the cloud server gets back to normal.
- Considering infinitely long time-series data, we further propose a period-based storage algorithm to improve the recovered data. The sensing data in each sensing period are encoded separately, but the reconstruction is on the entire data for exploring the inter-period correlation. We also give the mathematical analysis, indicating that the CS measurement matrix, formulated by our storage algorithm, can guarantee successful data recovery.
- The extensive experiments illustrate that the proposed compressive storage scheme can ensure the successful data recovery, even if only a fraction of measurements stored by the corresponding participants are utilized. Moreover, for multi-period time-series sensing data, our period-based algorithm achieves better recovery accuracy.

The remainder of the paper is organized as follows. Section 2 reviews the related work of storage approaches in WSNs and MCS. Section 3 introduces the basic concept of CS theory. Section 4 describes the virtual sensor network model and the compressive data storage scheme. Section 5 presents the mathematical foundation of our distributed storage algorithm, and Section 6 gives a brief discussion. Section 7 gives its performance through simulations. Finally, we conclude this paper in Section 8.

2 RELATED WORK

WSNs are traditional sensing infrastructures employed to monitor the environment. In WSNs, the sensor nodes perceive the environment information and produce sensor readings. One or several sink nodes are used as central servers, storing the data sensed by the sensor nodes in the network. A plethora of research efforts have been attracted to data gathering methods [10, 16, 39], in which the sensor readings are transmitted to the central sink nodes with low communication cost as much as possible.

It is worth noting that decentralized storage methods for WSNs have recently been introduced in several works [11, 14, 21, 32, 37], where sink nodes do not receive the sensing data as usual for various reasons. When a sensor node generates a reading, this sensor reading is disseminated throughout the network rather than being uploaded to the central sink nodes. Each node then receives and stores the readings disseminated from other nodes. For WSNs, the main concern is the

energy consumption of sensor nodes. Talari and Rahnavard [21] present a probabilistic broadcasting strategy for data dissemination, and the nodes employ CS techniques to encode the received data to save storage space with an acceptable energy cost. Yang et al. [32] propose to reduce the total number of data transmissions and receptions to save data dissemination cost. The cost is further reduced by designing a random walk based data dissemination algorithm [14]. In our earlier work [37, 38], we present a region-based **compressive networked storage (CNS)** algorithm, aiming at decreasing decoding ratio and improving data accuracy with less dissemination cost. In the work of Gong et al. [11], a storage method, called *ST-CNC*, is presented to exploit spatial and temporal correlations among sensor readings, which is more suitable for the storage of a spatial-temporal dataset. In practice, however, it is quite expensive to deploy and maintain a large-scale sensor network.

MCS is becoming a new paradigm to collect environmental data with the recent popularity of mobile devices and increasingly more powerful wireless network infrastructure. Considering the limited storage space in mobile sensing devices, typical MCS systems utilize cloud servers to store the sensing data uploaded by the participants. To obtain sufficient sensing data representing the whole target sensing area, the MCS campaign organizers often require the participants to meet the full-coverage requirement. A solution to implement full coverage is to design an effective scheduling algorithm, with which the cloud servers can select enough participants to collect data at the observation sites where necessary. Considering the dynamic environment of MCS systems, a checkpoint strategy is investigated in the work of Yuan et al. [33] to supervise the data collecting process. To lower overall scheduling cost, Wang et al. [24–26] introduce a sparse MCS method where the cloud servers only select a small number of observation sites for sensing while inferring the data of the remaining sites. In another work of Wang et al. [23], deep reinforcement learning theory is used to decide which sites are the best choice, improving data inference quality. In MCS systems, the cloud servers are responsible for scheduling the participants to guarantee high coverage so that it can receive and store the entire dataset corresponding to the target area.

Unfortunately, in some cases, such as the period after an earthquake, network outages may occur. At this moment, the clouds might not receive the sensing data properly, let alone scheduling the participants to achieve a desired coverage. In other words, cloud-based data storage in MCS systems may not work as usual at certain times. Inspired by the decentralized storage strategy with WSNs, in this work we will develop a new distributed storage scheme particularly suitable for MCS.

3 PRELIMINARIES OF CS

CS is a novel signal processing theory. It suggests that sparse signals are capable of being represented with much lower sampling rates, or significantly fewer CS measurements, than the traditional Shannon-Nyquist limit [4]. Looking at this from another perspective, one can recover an n-length signal accurately from a much lower number of measurements by employing CS theory. In this article, we are going to utilize CS theory to respond to the challenges of distributed storage in MCS systems.

CS processing includes two stages—a measuring process and a reconstructing process—that are sometimes referred to as the encoding process and the decoding process, respectively. Let $\boldsymbol{x}$ represent a sparse signal of size $1 \times n$ and ϕ be an $m \times n$ measurement matrix with $m \ll n$. According to CS theory, the measuring process is illustrated as

$$\boldsymbol{y} = \phi \boldsymbol{x}^{\mathsf{T}}, \tag{1}$$

where $\boldsymbol{x}^{\mathsf{T}}$ is the transpose of $\boldsymbol{x}$.

In CS theory, sparsity is the key aspect that enables recovery of signal $\boldsymbol{x}$ from CS measurements $\boldsymbol{y}$. Such sparsity is able to be with respect to some sparse transform φ. Denote α_n as the transform coefficients of $\boldsymbol{x}$ in terms of $\boldsymbol{\alpha}^\intercal = \varphi \boldsymbol{x}^\intercal$, and sort $\boldsymbol{\alpha}_n$ in descending order by magnitude (i.e., $|\alpha_n| \geq |\alpha_{n+1}|$). Define $\boldsymbol{\alpha}^K$ as a vector consisting of partial coefficients taken from $\boldsymbol{\alpha}$ by keeping the K largest coefficients, and set the rest to zero. Let $\boldsymbol{x}^{K\intercal} = \varphi^{-1}\boldsymbol{\alpha}^{K\intercal}$, where φ^{-1} is the inverse of matrix φ. Then one can say $\boldsymbol{x}$ is sparse in the transform domain and $\boldsymbol{x}^K$ can be approximate to $\boldsymbol{x}$,

$$\| \boldsymbol{x} - \boldsymbol{x}^K \|_2 \leq C_r R K^{-r}, \tag{2}$$

in the sense of

$$|\alpha_n| < R n^{-r}, \tag{3}$$

where $R < \infty, r \geq 1$, and C_r is a constant that depends only on r [7, 20]. When the magnitudes of the coefficients α_n have a power-law decay in terms of Equation (3), CS theory holds that the original signal $\boldsymbol{x}$ is capable of being recovered from $\boldsymbol{y} = \phi \boldsymbol{x}^\intercal$ with an acceptable accuracy. Specifically, the recovered signal, $\hat{\boldsymbol{x}}$, will meet

$$\| \hat{\boldsymbol{x}} - \boldsymbol{x} \|_2 \leq C \frac{\| \boldsymbol{x} - \boldsymbol{x}^K \|_1}{\sqrt{K}} \tag{4}$$

for some predetermined constant C. Note that real-world signals themselves, including natural images, temperature data, and **particulate matter (**PM**)** air concentration data we focus on in this article, are rarely sparse. But they are often approximately sparse in some transforming domain in terms of Equations (2), (3), and (4), and therefore can be applied to CS theory. We refer the reader to other works [7, 8, 14, 20] for detailed theoretical demonstration.

The CS reconstructing process is the reverse process of measuring. Let Φ is a matrix satisfying $\Phi = \phi\varphi^{-1}$. According to Equation (1), one has

$$\boldsymbol{y} = \Phi \boldsymbol{\alpha}^\intercal, \tag{5}$$

where $\boldsymbol{\alpha}^\intercal = \varphi^{-1}\boldsymbol{x}^\intercal$. It has been proved elsewhere [2, 7] that if measurement matrix $\boldsymbol{\phi}$ has its mutual coherence with sparsity basis φ, then $\boldsymbol{\alpha}$ can be reconstructed by solving the following equation,

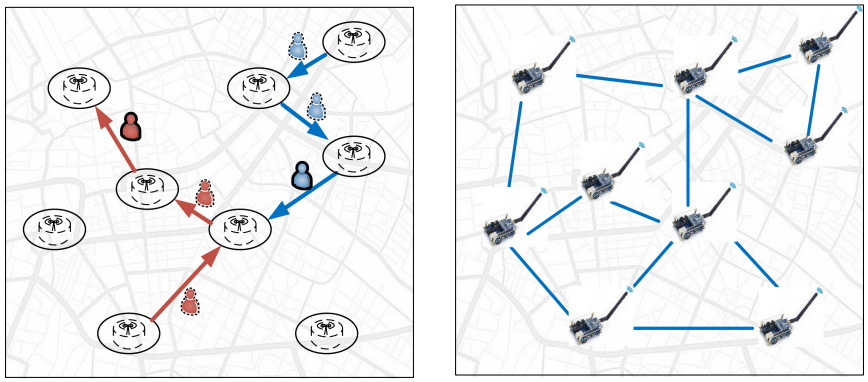
$$min \|\boldsymbol{\alpha}\|_1, \quad subject\ to\ \|\boldsymbol{y} - \Phi\boldsymbol{\alpha}^\intercal\|_2^2 \leq \lambda, \tag{6}$$

even if $m \ll n$. Here λ is a predefined small constant, and $\|\cdot|_1$ and $\|\cdot\|_2$ denote 1-norm and 2-norm, respectively. One can then recover the original $\boldsymbol{x}$ by using $\boldsymbol{x}^\intercal = \varphi\boldsymbol{\alpha}^\intercal$. Several CS reconstruction algorithms have been investigated, including the group-based GSR algorithm [34], matching pursuit [31], and the iterative D-AMP algorithm [17], among others.

Compared to general encoding and decoding algorithms, in CS the measuring process is quite simple, as shown in Equation (1). This is beneficial to the resource-limited mobile sensing device used to encode and store the sensing data in MCS systems. In contrast to the simple measuring operation, the reconstruction algorithm is often with high complexity. Fortunately, the reconstruction process runs on the cloud servers not involving the mobile devices. Once the network returns to normal, the data field corresponding to the target area is reconstructed on the powerful cloud servers.

4 COMPRESSIVE STORAGE STRATEGY FOR MCS SYSTEMS

This section first gives the definition of the virtual sensor network, then proposes a compressive encoding strategy to store the data in the target sensing area. After that, a period-based reconstruction algorithm is presented to improve the recovered data.



(a) A MCS system with nine observation sites and two participants

(b) A sensor network with nine dedicated sensors

Fig. 1. An instance of an MCS system from a virtual sensor network perspective and the corresponding sensor network. In a virtual sensor network, a geographical observation site is seen as a virtual node, and the movements of the participants form data links between the virtual nodes.

4.1 Defining a Virtual Sensor Network

This section defines a novel concept of virtual sensor network relating to the target sensing area so that one can apply the classical CS theory to the specific MCS storage scenarios by following the framework of traditional WSNs. As we know, WSNs may be the most desirable method to collect the environment information if one does not take the expensive deployment and maintenance cost as the consideration.

We believe that with an MCS system participated in by human beings, a target sensing area itself can be imagined as a sensor network. We term this imaginary network as a *virtual sensor network*. Figure 1 shows an instance of an MCS system from a virtual sensor network perspective. In Figure 1(a), an ellipse represents an observation site in the target sensing area, and a site is seen as a virtual node. It is observed from Figure 1(a) that the traditional sensor network can achieve full coverage of the sensing area, but Figure 1(a) shows that there exist monitoring holes in the MCS system since two observation sites are not visited by the participants.

The concepts of the proposed virtual sensor node and virtual sensor network are illustrated in Definitions 4.1 and 4.2 and Propositions 4.1 and 4.2, respectively.

Definition 4.1 (Virtual sensor node). A virtual sensor node is a geographical observation site in the target sensing area.

Proposition 4.1. *Virtual nodes generate sensor readings, which are the environment information associated with the corresponding observation sites. However, the virtual nodes are memoryless. In other words, virtual nodes can never store the readings since they are physically nonexistent.*

Definition 4.2 (Virtual sensor network). A virtual sensor network consisting of n nodes is a geographical sensing area with n corresponding observation sites.

Proposition 4.2. *In a virtual sensor network, data forwarding among the virtual nodes are implemented by the movement of the participant with mobile sensing devices. The sensor readings generated by a virtual node are stored by the participants only if their movement trajectories cover this node.*

Our idea of virtual sensor network stems from the fact that environment information actually relates to the observation sites, not the dedicated sensors. In other words, even without physical

sensor nodes, the observation sites still keep generating environment information. In this way, we can consider the target sensing area as a virtual sensor network, where the observation sites are virtual nodes and the participants moving around are responsible for data forwarding. The proposed network is a virtual one without intensive real sensor deployment. Its monitoring resolution depends on the number of the mobile devices carried by the participants. The number of mobile devices indicates the spatial resolution, whereas the minimum time interval for sensing process determines temporal resolution. The ongoing advancement of technology of mobile devices will enable our proposed virtual sensor network more practical solution.

By defining a virtual sensor network, we are capable of following the road map of decentralized storage in traditional WSNs to investigate a distributed storage scheme suited for MCS systems. Note that the virtual nodes are not physical sensors and do not have memory space to store the sensor readings. In other words, the sensor readings are in fact discarded if no participants pass the corresponding virtual nodes. It also makes our distributed storage algorithm for MCS systems different from that with traditional WSNs.

4.2 Compressive Storage with CS-Based Encoding

This section first presents an appropriate package structure for facilitating CS encoding operation. Then an encoding algorithm is proposed to implement CS-based compressive distributed storage in MCS systems.

4.2.1 Data Package Structure. We design a structure of the data package forwarded among the nodes in the virtual sensor network, saving not only the readings generated by virtual sensor nodes but also the traveling trajectories of the participants.

Let dp_j denote the data package carried by the j-th participant, P_j. Package dp_j includes two components, $dp_j\{0\}$ and $dp_j\{1\}$, where $dp_j\{0\}$ saves the IDs of the virtual nodes that participant P_j visits and the arrival time of P_j, and $dp_j\{1\}$ saves the corresponding readings. dp_j is initially defined by

$$dp_j : \begin{cases} dp_j\{0\} = \varnothing \\ dp_j\{1\} = \varnothing. \end{cases} \tag{7}$$

When participant P_j moves to virtual node s_p at time t_q, they update the data in dp_j by employing the following operations:

$$dp_j : \begin{cases} dp_j\{0\} = dp_j\{0\} \cup [s_p, t_q], \\ dp_j\{1\} = dp_j\{1\} \cup reading_{p,q}, \end{cases} \tag{8}$$

where $reading_{p,q}$ is the reading of node s_p at time t_q.

It is observed from Equation (8) that data package dp_j stores the values of all sensor readings associated with virtual nodes participant P_j passes through. What is more, dp_j actually records the moving trajectory of the j-th participant, which is indicated in $dp_j\{0\}$. The trajectory of the participant will be further investigated to implement CS distributed encoding in the following sections.

4.2.2 Encoding Algorithm for a Single Participant. Our aim has two sides: (1) compressing the readings to save storage space, and, more importantly, (2) extracting the key information from the local trajectory of a participant to recover the data field with regard to the entire sensing area. When a participant finishes the walk in the target sensing area, they perform the encoding algorithm and store the encoded measurement.

The sensor readings collected by a participant often have data correlation and can be compressed with a reasonable encoding algorithm. As shown in Figure 2, the scenario, where participant P_j carrying package dp_j walks through the target sensing area, is like forwarding dp_j from a virtual

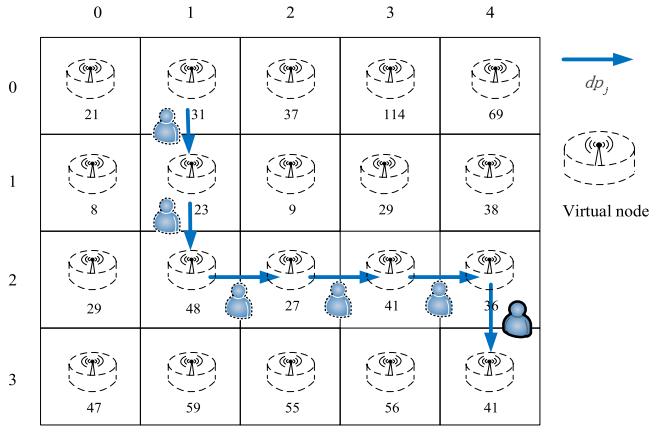


Fig. 2. An instance of the movement of package dp_j in a virtual sensor network, where each virtual node generates a $PM_{2.5}$ concentration datum, and the entire data is of size 20 consisting of a matrix of size 4×5. We say that the movement of dp_j indicates the encoding operation for the entire data, although the trajectory of dp_j is limited in a local scope— only seven nodes in this instance.

node to another. The sequence of the nodes forwarding dp_j is $\{s_{0,1}, s_{1,1}, s_{2,1}, s_{2,2}, s_{2,3}, s_{2,4}, s_{3,4}\}$, and the corresponding $PM_{2.5}$ readings, $\{51, 13, 48, 27, 41, 36, 41\}$, are saved by P_j. These readings are adjacent to each other and may be compressed by exploring the correlation therein. Here we omit the time dimension for simplicity.

Taking Figure 2 as an example, the route of data packet dp_j just covers a part of nodes, and only seven readings are saved in dp_j. In other words, the moving trajectory of participant P_j is limited in local scope. However, we argue that seven nodes forwarding dp_j actually record the key information recovering all 20 data across the entire sensing area. The local trajectory can be viewed as a global sampling for the sensing area. Toward this end, we formulate the sampling process as

$$\begin{pmatrix} 0 & 31 & 0 & 0 & 0 \\ 0 & 23 & 0 & 0 & 0 \\ 0 & 48 & 27 & 41 & 36 \\ 0 & 0 & 0 & 0 & 41 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \circ \begin{pmatrix} 21 & 31 & 37 & 114 & 69 \\ 8 & 23 & 9 & 29 & 38 \\ 29 & 48 & 27 & 41 & 36 \\ 47 & 59 & 55 & 56 & 41 \end{pmatrix}. \tag{9}$$

Here "$\circ$" is the Hadamard product representing the sampling operator, and

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

is a sampling matrix. From Equation (9), by constructing this sampling matrix with 0 and 1, the partial data saved by P_j can be seen as a sampling result for the entire data field.

We further generalize Equation (9) into a universal form. Let Ψ_j indicate the sampling matrix extracted from trajectory of P_j saved in $dp_j(0)$, X represents the entire data relating to the sensing area, and X_j is the partial sample result. The sampling process of P_j is then generalized as

$$\boldsymbol{X}_j = \Psi_j \circ \boldsymbol{X}. \tag{10}$$

We denote $\boldsymbol{x}_j$ as

$$\boldsymbol{x}_j = \mathrm{vec}(\boldsymbol{X}_j), \tag{11}$$

ALGORITHM 1: Encoding algorithm for participant P_j

Require: dp_j, ϕ_j;
Ensure: $\boldsymbol{y}_j$, Ψ_j;
1: Construct sampling matrix Ψ_j according to the trajectory of P_j stored in dp_j;
2: Compute the sampling result $\boldsymbol{X}_j$: $\boldsymbol{X}_j = \Psi_j \circ \boldsymbol{X}$;
3: $\boldsymbol{x}_j = \text{vec}(\boldsymbol{X}_j)$;
4: Compute the encoding result $\boldsymbol{y}_j$: $\boldsymbol{y}_j = \phi_j \boldsymbol{x}_j^\mathsf{T}$;

where vec$(\cdot)$ is a function reshaping a matrix to a row vector. Let ϕ_j be the j-th row of a predefined CS measurement matrix ϕ. By employing the measuring technique in CS theory, the encoding operation of participant P_j is shown as

$$\boldsymbol{y}_j = \phi_j \boldsymbol{x}_j^\mathsf{T}, \tag{12}$$

where $\boldsymbol{y}_j$ is the encoding result. Participant P_j then stores the encoded measurement, $\boldsymbol{y}_j$.

Algorithm 1 summarizes the encoding process for any participant P_j.

4.2.3 Distributed Storage from the Global Perspective. From a single participant perspective, a participant stores an encoded value. Let $\psi_j = \text{vec}(\Psi_j)$ and $\boldsymbol{x} = \text{vec}(\boldsymbol{X})$. According to the second, third, and fourth steps in Algorithm 1, the encoding process for participant P_j can be written as shown:

$$\begin{aligned} \boldsymbol{y}_j &= \phi_j \boldsymbol{x}_j^\mathsf{T} \\ &= \phi_j (\text{vec}(\Psi_j \circ \boldsymbol{X}))^\mathsf{T} \\ &= (\phi_j \circ \psi_j) \boldsymbol{x}^\mathsf{T} \\ &= \mathcal{A}_j \boldsymbol{x}^\mathsf{T}, \end{aligned} \tag{13}$$

where $\mathcal{A}_j = \phi_j \circ \psi_j$. Each participant performs the same encoding process.

From the global perspective, m participants in the target sensing area store m encoded measurements. The global encoding process is formulated as

$$\underbrace{\begin{pmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_j \\ \vdots \\ \boldsymbol{y}_{m-1} \end{pmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{pmatrix} \mathcal{A}_0 \\ \mathcal{A}_1 \\ \vdots \\ \mathcal{A}_j \\ \vdots \\ \mathcal{A}_{m-1} \end{pmatrix}}_{\mathcal{A}} \boldsymbol{x}^\mathsf{T}. \tag{14}$$

Here $\boldsymbol{x}$ is of size $1 \times n$ and $\mathcal{A}$ is an $m \times n$ matrix. In this way, the entire data field, $\boldsymbol{x}$, is stored by m participants in the form of encoded result, $\boldsymbol{y}$.

As mentioned earlier, the trajectories of the participants are often limited to a local region. However, from Equations (13) and (14), any $P_j (0 \le j < m)$ can make an encoding operation to the entire data field, not the local data collected by themselves, thanks to the CS theory.

4.3 Data Recovery from the Encoded Measurements

4.3.1 Data Recovery with Classical CS Theory. As illustrated in Equation (14), the distributed storing process by m participants in an MCS system is formulated as

$$\boldsymbol{y} = \mathcal{A} \boldsymbol{x}^\mathsf{T}, \tag{15}$$

where $\mathcal{A} = \psi \circ \phi$. Note that ϕ is a known CS measurement matrix, but ψ is constructed according to the traveling trajectories of the participants in the target area. In the next section, we will prove that matrix $\mathcal{A}$ formed in the process of distributed storage has a mathematical nature similar to a common CS measurement matrix. As a real-world signal, $\boldsymbol{x}$ is not sparse, but it shows sparsity in some transform domain. Let φ be a transform matrix, and define $\boldsymbol{\alpha}^\mathsf{T} = \varphi \boldsymbol{x}^\mathsf{T}$ and $\Phi = \mathcal{A}\varphi^{-1}$. One can easily derive a standard CS measuring equation from Equation (15) as

$$\boldsymbol{y} = \Phi \boldsymbol{\alpha}^\mathsf{T}. \tag{16}$$

It is observed that Equation (16) is exactly the same in form as Equation (5) of Section 3. CS theory holds that the coefficients of the transform domain, $\boldsymbol{\alpha}$, can be deduced from Equation (16) by solving a optimization problem shown in Equation (6) of Section 3. The original $\boldsymbol{x}$ can then be achieved with $\boldsymbol{x}^\mathsf{T} = \varphi^{-1}\boldsymbol{\alpha}^\mathsf{T}$.

We note that ϕ is constructed according to the traveling trajectories of the participants, but the transform matrix φ is tightly associated with the distribution of the sensing data. In other words, $\boldsymbol{\alpha}$ obtained with different ψ will show different sparsity. A plethora of research efforts have been introduced to exploit more efficient sparse transform according to the statistical distribution of a specific signal [19, 27]. Thiswork does not investigate a new sparse transform but instead focuses on how to construct an efficient storage-related measurement matrix. In the experiment, the well-known BIOR 1.5 wavelet is used as sparse transform basis that is applied to both our storage scheme and the competing ones for a fair comparison.

4.3.2 Period-Based Encoding but Entire-Data Recovery. Considering the ever-increasing amount of data in MCS systems, we propose to encode the sensing data separately according to the sensing period but take the entire dataset into consideration in the decoding processing for improving the recovered data.

The environmental data of the target sensing area are generated continuously throughout the life of MCS systems. It is not practical to store all time-series sensing data at once. Intuitively, the data storage process should be divided into multiple sensing periods, and the data in each period can then be reconstructed using a CS reconstruction algorithm independently. A period-independent recovery strategy is reasonable, but it ignores the inter-period correlation, and there is still room for improvement. Our idea of entire-data recovery comes from block-based CS theory [6, 9, 36], where a large image is spatially partitioned into several blocks to decrease reconstruction complexity while guaranteeing the image quality. By contrast, in the MCS system, the sensing data are temporally divided into a number of sensing periods, making our scheme more practical to store the very long time-series data.

Suppose that the entire data, $\boldsymbol{x}^{entire}$, are divided into L sensing periods taking the form of $\boldsymbol{x}^{entire} = (\boldsymbol{x}^0, \ldots, \boldsymbol{x}^{L-1})$. According to Equation (15), the storage process of the target sensing area for the l-th period is rewritten as

$$\boldsymbol{y}^l = \mathcal{A}^l \, \boldsymbol{x}^{l\mathsf{T}}. \tag{17}$$

The sensing data in all L sensing periods are first recovered period by period using a general CS reconstruction algorithm. Let $\hat{\boldsymbol{x}}^l(0)$ represent the initial result of the l-th period. The direct combining of all L initial results, $\hat{\boldsymbol{x}}^{entire}(0) = \{\hat{\boldsymbol{x}}^l(0)\}_{l=0}^{L}$, may cause the artifacts due to period division.

Given this, we present a reconstruction algorithm to exploit inter-period correlation by using the denoising technique and projection onto the convex set. The proposed algorithm consists of two stages. In the first stage, the denoising operation is performed on the entire data to ameliorate the period-partition artifacts by exploiting the correlation across the sensing periods. The denoising

ALGORITHM 2: Period-based data reconstruction algorithm

Require: $\hat{\boldsymbol{x}}^l, \boldsymbol{y}^l, \mathcal{A}^l, k, num, Threshold$;
Ensure: $\hat{\boldsymbol{x}}^{entire}(k+1)$;
1: **for** $k = 0$ to $num - 1$ **do**
2: **for** $l = 0$ to $L - 1$ **do**
3: $\hat{\boldsymbol{x}}^l = \mathcal{P}(\hat{\boldsymbol{x}}^l, \boldsymbol{y}^l, \mathcal{A}^l)$;
4: **end for**
5: $\hat{\boldsymbol{x}}^{entire}(k+1) = \mathcal{D}(\hat{\boldsymbol{x}}^{entire}(k))$;
6: $error = \frac{\|\hat{\boldsymbol{x}}^{entire}(k+1) - \hat{\boldsymbol{x}}^{entire}(k)\|_2}{\|\hat{\boldsymbol{x}}^{entire}(k)\|_2}$;
7: **if** $error < Threshold$ **then**
8: break;
9: **end if**
10: **end for**

processing in the k-th iteration is shown in

$$\hat{\boldsymbol{x}}^{entire}(k+1) = \mathcal{D}(\hat{\boldsymbol{x}}^{entire}(k)), \tag{18}$$

where $\mathcal{D}(\cdot)$ represents a general denoiser.

In the second stage, projection onto the convex set is employed to approximate the original sensing data, since denoising operation involves a loss of precision. To find the approximation closest to the original data, we use Equation (19) presented in the work of Candes and Romberg [3]:

$$\mathcal{P}(\boldsymbol{x}^l, \boldsymbol{y}^l, \mathcal{A}^l) = \boldsymbol{x}^{l\mathsf{T}} + \mathcal{A}^{l\mathsf{T}}(\mathcal{A}^l\mathcal{A}^{l\mathsf{T}})^{-1}(\boldsymbol{y}^l - \mathcal{A}^l\boldsymbol{x}^{l\mathsf{T}}). \tag{19}$$

These two stages are combined iteratively, improving the recovered data. The proposed sensing period based approach is somewhat analogous to our strategy presented in earlier work [37]. In that case, a WSN is spatially divided into ϖ regions, whereas in this work the sensing data are temporally divided into L sensing periods and $\mathcal{A}$ is not a general random matrix but the one generated by our compressive storage scheme. Suppose that *num* is the number of iterations, *Threshold* is the predefined threshold of the reconstruction accuracy. The detailed description of this process is shown in Algorithm 2.

We can observe from Algorithm 2 that the denoiser $\mathcal{D}$ acts on the entire data $\hat{\boldsymbol{x}}^{entire}$, not the sensing data $\hat{\boldsymbol{x}}^l$ corresponding to a single sensing period. The artifacts because of period partition are then ameliorated. At the same time, the data precision lost in the denosing process is compensated by projection onto the convex set shown in Equation (19). Denoising and projection operations are performed iteratively, and the recovered data, $\hat{\boldsymbol{x}}^{entire}$, are thus improved.

5 THEORETICAL ANALYSIS

In Section 4, we formulated the distributed storage in MCS systems as CS encoding and decoding processes. This section demonstrates that the proposed compressive storage strategy is capable of guaranteeing successful data recovery in MCS systems.

Suppose that ϕ is an $m \times n$ measurement matrix satisfying the condition of successful CS reconstruction. In other words, ϕ is incoherent with any given sparsity transform basis with a high probability [1, 2]. Let $\psi = \{\psi_j\}_{j=0}^{m-1}$ be a matrix extracted from the packages $\{dp_j\}_{j=0}^{m-1}$ generated by all m participants. As illustrated in Section 4.2.2, the j-th row of ψ, ψ_j, represents the moving trajectory of participant P_j, and ψ is a binary matrix consisting of the elements of 0 or 1. At this moment, one has $\mathcal{A} = \psi \circ \phi$. Then we have the following theorem.

Theorem 5.1. *If matrix ϕ has the mutual coherence with any given sparsity basis, φ (i.e., $\mu(\phi, \varphi) < \gamma$, where γ is a positive constant), then matrix $\mathcal{A}$ generated by the participants in an MCS system has the mutual coherence with the basis ψ as well. In other words,*

$$\mu(\mathcal{A}, \varphi) < c\gamma, \tag{20}$$

where c is a predefined positive constant satisfying $c \leq 1$.

Proof. Suppose that ϕ_j represents the j-th row of CS measurement matrix ϕ of size $m \times n$, and φ_i is the i-th column of sparsity basis φ of size $n \times n$. Then one has

$$\mu(\phi, \varphi) = \max_{0 \leq j < m, 0 \leq i < n} \frac{|\langle \phi_j, \varphi_i \rangle|}{\|\phi_j\|_2 \|\varphi_i\|_2} < \gamma, \tag{21}$$

where $\langle \cdot, \cdot \rangle$ represents an inner product, $|\cdot|$ and $\|\cdot\|_2$ denote the operations of absolute value and 2-norm, respectively.

Since $\mathcal{A} = \psi \circ \phi$, the mutual coherence between $\mathcal{A}$ and φ is shown as

$$\mu(\mathcal{A}, \varphi) = \max_{0 \leq j < m, 0 \leq i < n} \frac{|\langle \psi_j \circ \phi_j, \varphi_i \rangle|}{\|\psi_j \circ \phi_j\|_2 \|\varphi_i\|_2}. \tag{22}$$

As mentioned in the previous section, the movements of the participants in MCS systems are often limited in local scope. As a consequence, most elements in matrix ψ are 0. Let $\frac{1}{p}$ represent the proportion of 1 in ψ_j, where p is larger than 2. Then one has $|\langle \psi_j \circ \phi_j, \varphi_i \rangle| \simeq \frac{1}{p} |\langle \phi_j, \varphi_i \rangle|$ and $\|\psi_j \circ \phi_j\|_2 \simeq \sqrt{\frac{1}{p}} \|\phi_j\|_2$. So, $\frac{|\langle \psi_j \circ \phi_j, \varphi_i \rangle|}{\|\psi_j \circ \phi_j\|_2 \|\varphi_i\|_2} = \sqrt{\frac{1}{p}} \frac{|\langle \phi_j, \varphi_i \rangle|}{\|\phi_j\|_2 \|\varphi_i\|_2}$ with a high probability. Let $c = \sqrt{\frac{1}{p}}$. According to Equation (21), one then has

$$\mu(\mathcal{A}, \varphi) \simeq \sqrt{\frac{1}{p}} \mu(\phi, \varphi) < c\gamma. \tag{23}$$

□

In our distributed storage scheme, ϕ is a predefined CS measurement matrix. So Theorem 5.1 can show that matrix $\mathcal{A}$, formed by the proposed storage strategy, satisfies the condition of CS reconstruction in theory. We believe that it is the randomness of the movements of the participants that ensures the excellent property of our matrix $\mathcal{A}$. In Section 7, we further validate it through numerical experiments.

6 DISCUSSION

As mentioned in Section 1, in distributed storage for MCS systems, three major challenges have to be faced. The first challenge is the limited storage resource of the mobile sensing devices. In MCS systems, these devices are privately owned by participants, so the MCS organizers do not have to plan more budgets for deploying dedicated sensors, offering significant cost savings in contrast to traditional WSNs. However, this also makes it impractical for the participants to spend too much additional storage space to store sensing data. Data compression is an intuitive solution. But general compression algorithms do not consider the data correlation between a participant and the other ones, and the existing distributed algorithms are also not suitable for the storage in MCS systems without the support of central cloud servers. Given this, this article presents a CS-based distributed encoding algorithm, in which the encoded measurements, not the raw sensing data, are stored to decrease the amount of the sensing data in the target area.

The second challenge we are facing is monitoring holes, which lead to data loss in the target sensing area. Unfortunately, the monitoring holes themselves are unavoidable, since the participants cannot be scheduled to certain observation sites to implement full coverage in the absence of central cloud servers. This is additional proof that general compression algorithms are unfit for the storage scenario in MCS systems. The proposed CS-based encoding algorithm and the corresponding data recovery algorithm consider the problem of monitoring holes. Our idea is to reconstruct the missing data by utilizing the correlation between the collected data and the missing ones. We base our idea on CS theory and prove that measurement matrix $\mathcal{A}$, formed by our distributed encoding algorithm, satisfies a certain mathematical property. Thus, the original data field in the target sensing area can be recovered with a reasonable level of precision.

The third challenge is the possibility of participants leaving the MCS systems at will. If so, the encoded measurements of the sensing data stored in these unreliable participants may disappear into the ether. Our solution for this situation is to make no distinction about normal participants and unreliable ones. It is our hope that no participants are essential to the storage in MCS systems, and any participant may be replaced by someone else. This is achieved by our proposed CS-based encoding strategy, in which the encoded measurement of each participant reflects the global information of the entire target area regardless of their specific moving trajectories. Therefore, in the encoding sense, any participant is equivalent to another one. In this way, the entire data field corresponding to the target area can be recovered, as long as the number of participants satisfies the requirement of the decoding ratio without waiting for a certain participant until the cloud servers get back to normal.

As mentioned earlier, we utilize CS theory and follow the line of WSNs to address these challenges. For this purpose, the concept of the virtual sensor network is defined to abstract the MCS systems. A geographical area is seen as a virtual sensor network, where the observation sites are the network nodes while the movements of the participants form the data links in the network. The performance indexes of the virtual sensor network, such as throughput, bandwidth, and packet transfer rate, depend on the movements of the participants as well as the hardware parameters of their mobile devices. In this article, we omit the network performance analysis of the virtual sensor network itself for simplicity since our emphasis is to introduce the distributed storage scheme in MCS systems.

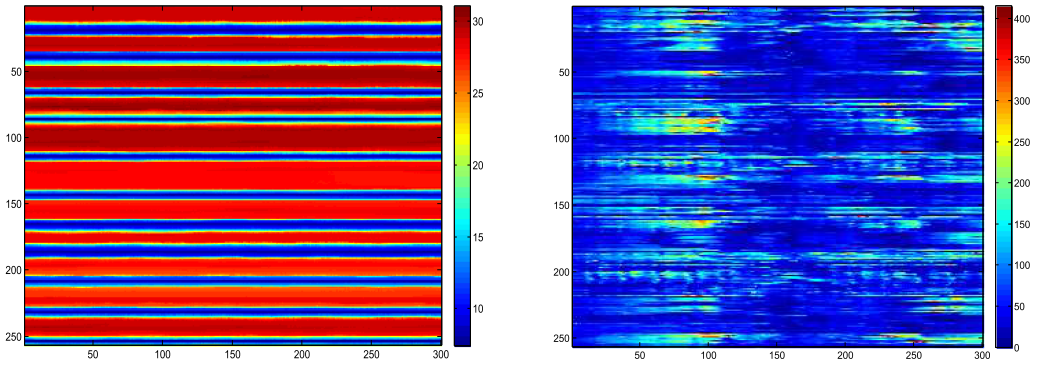
7 PERFORMANCE EVALUATION

This section first gives the simulation scenario and the criteria of performance evaluation. Next, the recovery performance of the proposed compressive storage scheme is evaluated by using temperature and air quality datasets. Then, our scheme is compared to two competing methods with dedicated WSNs—that is, CNS (CNS-WSNs) [37] and spatio-temporal compressive storage (ST-CNC-WSNs) [11].

7.1 Experimental Setup

The simulated target sensing area consists of S observation sites, each of which keeps generating time-series data. For traditional WSNs, S sensor nodes have to be deployed to the corresponding S sites, and the environmental data are collected and stored by these nodes. As for MCS, a number of participants walk around with their mobile sensing devices, gathering and storing the data generated from these S sites.

To perform the numerical simulation, we use the real-world sea surface and subsurface temperature datasets from the National Data Buoy Center [30] as well as air quality datasets from China National Environmental Monitoring Centre [35]. The visual maps of temperature data and $PM_{2.5}$ concentration data used in the simulation are shown in Figure 3. The relative square error and



(a) Temperature data (°C) (b) $PM_{2.5}$ concentration data ($\mu g/m^3$)

Fig. 3. Visual maps of two test datasets of size 256×300.

mean absolute error are employed to test the quality of the recovered data field of the target area. The decoding ratio is used to evaluate the compression performance of the storage algorithms. Suppose that x represents the original spatial-temporal data of size $R \times C \times T$ in terms of the target area, where $R \times C$ denotes two spatial dimensions and T is the temporal dimension, and $\hat{x}$ is the recovered data field from the stored values in WSNs or MCS. The storage process is divided into P sensing periods, simulating the scenarios where one has to store a large amount of data. Let $n = R \cdot C \cdot T$ and $m = \sum_{l=1}^{P} m_p$, where m_p represents the number of CS measurements in the p-th sensing period of the storage process (i.e., the number of encoded measurements by sensor nodes or the participants). In other words, m indicates the number of the participants walking in the target sensing area and n represents the size of the data field being recovered. The relative square error $RSE(\boldsymbol{x}, \hat{\boldsymbol{x}})$, mean absolute error $MAE(\boldsymbol{x}, \hat{\boldsymbol{x}})$, and the decoding ratio DR are given by

$$RSE(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}\|_2}, \tag{24}$$

$$MAE(\boldsymbol{x}, \hat{\boldsymbol{x}}) = \frac{\|\boldsymbol{x} - \hat{\boldsymbol{x}}\|_1}{n}, \tag{25}$$

$$DR = \frac{m}{n}, \tag{26}$$

respectively. Here $\|\cdot\|_2$ denotes 2-norm and $\|\cdot\|_1$ is 1-norm. We say that the original data field is successfully recovered if the mean absolute error is lower than a given threshold.

All the computation is done using the MATLAB R2015b simulator on a server platform configured with 256 GB of memory and two 3.2-GHz Intel CPUs. The same CS reconstruction algorithm, D-AMP [17] combined with a BM3D denoiser and BIOR 1.5 wavelet transform, is utilized to recover the data field stored by both our storage scheme and the competing ones for a fair comparison.

7.2 Performance Evaluation of the Proposed Storage Scheme

The target sensing area consists of 900 observation sites with a spatial 30×30 grid setup. Each observation site generates 700 time-series data. A participant starts his or her walk from an observation site at time t at random, then walks more than s steps. In the simulation, t satisfies $0 < t < 700 - s$ and s is equal to 300, 400, and 500, respectively. During the period of the random

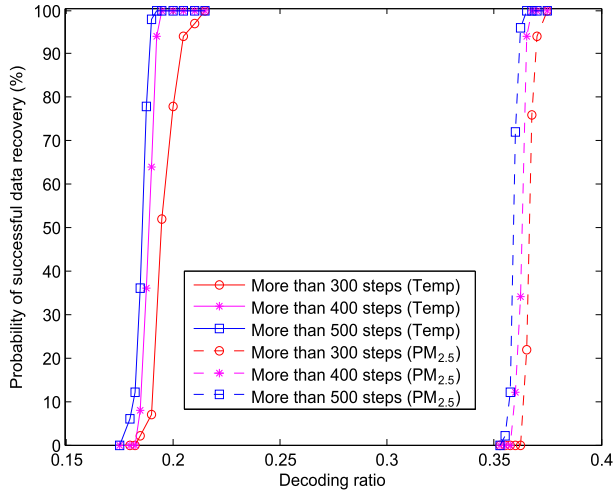
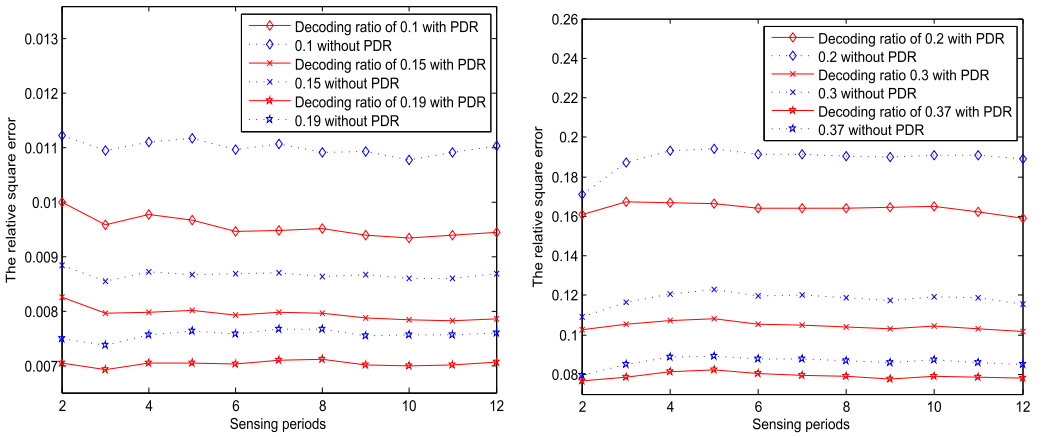


Fig. 4. Data recovery probability. The temperature and $PM_{2.5}$ data can be 100% recovered when the decoding ratio exceeds 0.19 and 0.37 for the scenario with 500 steps, respectively.

walk, the participants encode and store their sensing data generated by the corresponding observation sites. We designate to recover the data of size $16 \times 16 \times d$ in the central area of $30 \times 30 \times 700$ data space, focusing on the storage performance regardless of the region border.

In the storing process, d is set to 300 consisting of six sensing periods. At this point, one has $n = 16 \times 16 \times 300 = 76{,}800$. We perform 50 repeated simulations and use the probability of successful data recovery to evaluate the performance of the proposed storage scheme. Figure 4 shows that we are capable of attaining 100% recovery when the decoding ratio exceeds 0.21 for temperature data and 0.37 for $PM_{2.5}$ concentration data for all scenarios with various steps, respectively. Here, the threshold of the mean absolute error with regard to temperature data is set to 0.1°C. In other words, if the mean absolute error of the recovered temperature data is less than 0.1°C, then we say that the desired temperature field is successfully reconstructed. As for $PM_{2.5}$ concentration data, the corresponding threshold is set to 5 $\mu g/m^3$. From Figure 4, temperature data require a lower decoding ratio to achieve successful recovery than $PM_{2.5}$ concentration data. The reason is that temperature data are more correlated than air concentration, as can be seen from the visual maps in Figure 3. It can also be observed from Figure 4 that as the steps of the participants walking in the targeted area increase, one can get slightly better data reconstruction performance. In the scenarios where the participants walk more than 500 steps in the target sensing area, the decoding ratio is only 0.19 for temperature data. This is due to the distribution of 0 and 1 in the CS measurement matrix. Fewer steps of a participant mean a lower number of 1 in the corresponding row in the measurement matrix, which may impair the performance of the reconstruction algorithm.

Figure 5 indicates the recovery accuracy of the proposed storage scheme with and without period processing. Here the minimum steps of the participants are set to 500. The storage process consists of 12 sensing periods, and in each period a participant stores 50 data. For temperature data, we choose three decoding ratios, 0.1, 0.15, and 0.19, whereas for $PM_{2.5}$ concentration, the three decoding ratios are 0.2, 0.3, and 0.37, respectively. It is observed from Figure 5 that the proposed **period-based data recovery (PDR)** algorithm can boost the recovery performance on all three decoding ratios. This is because our proposed compressive storage scheme exploits not only the spatial-temporal correlation within one sensing period but also inter-period temporal



(a) Temperature data (b) $PM_{2.5}$ concentration data

Fig. 5. Recovery error comparison. Our scheme always has a smaller error than that without the technique of PDR for k sensing periods partition. Here $2 \leq k \leq 12$.

correlation. Figure 5 shows that the lower the decoding ratio is, the more the relative error decreases. This indicates that the algorithm benefits more if the lower decoding ratio is employed. Looking at this from another perspective, exploiting inter-period correlation enables us to recover the data field with fewer participants, decreasing the monetary cost of recruiting the participants.

Figure 6 illustrates how the recovery precision changes with the number of participants and the number of steps each participant needs to walk. The target sensing area is set to a two-dimensional grid size of 30×30 without considering the time dimension for simplicity. The minimum number of steps is set to 10. Note that the decoding ratio is defined as $DR = m/n$, where m is the number of participants and $n = 900$ represents the size of data being recovered. It can be observed from Figure 6 that when the number of steps exceeds 50, the steps do not affect data recovery much, and the recovery error is primarily determined by the number of participants. This is because, with the proposed CS-based scheme, each participant generates and stores only one measurement regardless of how many steps the participant walks. This result agrees with that shown in Figure 4, where the time dimension of the data field is further considered. It can also be observed from Figure 6 that fewer steps indicate slightly larger error values. This may be due to the distribution of 0 and 1 in the CS measurement matrix. Fewer steps of a participant mean a lower number of 1 in the corresponding row of the measurement matrix, which may impair the performance of the CS reconstruction algorithm.

7.3 Comparison with Existing Methods

This section compares our scheme with two state-of-the-art methods with dedicated WSNs, including the spatial CNS-WSNs and the spatial-temporal ST-CNC-WSNs. For WSNs, 900 sensor nodes are supposed to deploy to the observation sites in the target sensing area, and each node encodes and stores 700 sensing data. As in Section 7.2, we only recover the middle 256×300 data with six sensing periods from the central subarea with 256 nodes. In this way, we can focus on evaluating the storage algorithms themselves without being impacted by the network edges when disseminating the sensing data.

Figure 7 gives the results of performance evaluation in terms of the relative square error. It is observed from Figure 7 that for both temperature and $PM_{2.5}$ data, our scheme has far higher

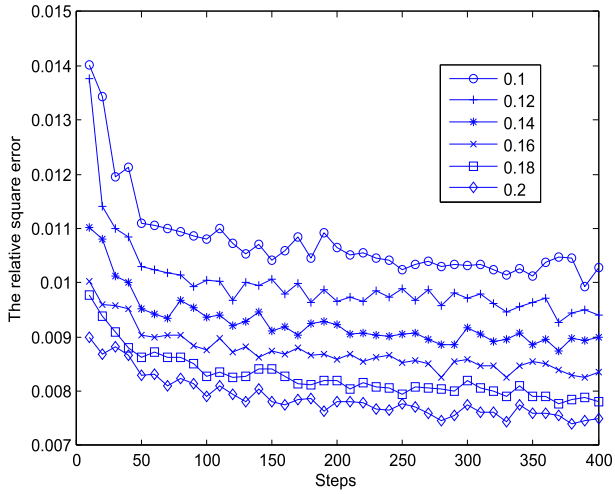
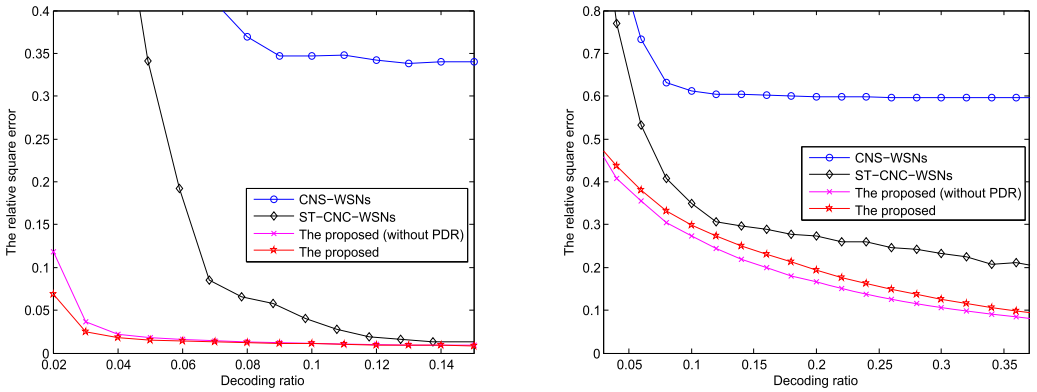


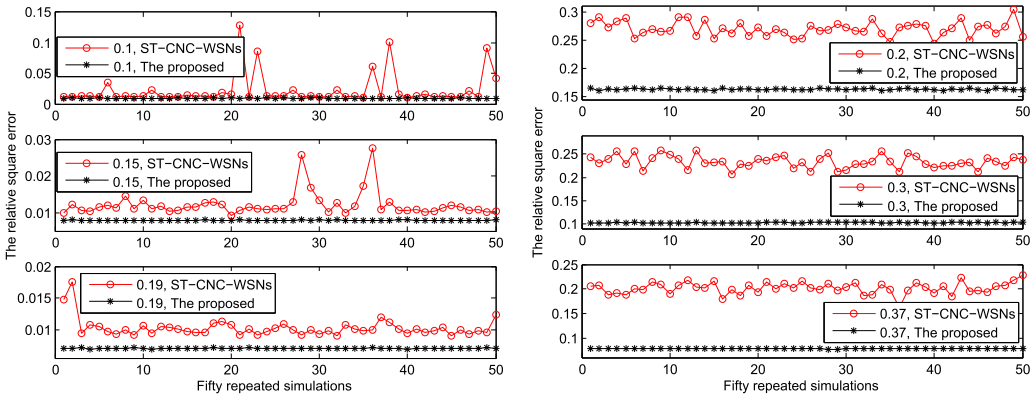
Fig. 6. The relative square error with various numbers of steps at the decoding ratio of 0.1, 0.12, 0.14, 0.16, 0.18, and 0.2.



(a) Temperature data (b) $PM_{2.5}$ concentration data

Fig. 7. Recovery performance comparison at various decoding ratios.

recovery accuracy at various decoding ratios. Our scheme also outperforms that without using the PDR algorithm. CNC-WSNs is our proposed region-based networked storage method for WSNs where we consider 256 sensor nodes as one region. Unfortunately, the CNC-WSNs method only exploits the spatial data correlation, and the temporal correlation among the time-series data is ignored. By contrast, ST-CNC-WSNs takes the spatial-temporal correlation into consideration and has lower error. However, it uses the Kronecker product framework to implement spatial-temporal data recovery in order to not affect the data dissemination in WSNs, resulting in degraded data recovery accuracy. In the proposed scheme, we design a united CS measurement matrix to handle the spatial and temporal correlation simultaneously. Moreover, our scheme further exploits the inter-period correlation by designing a period-based recovery algorithm and thus achieve better data recovery performance. As shown in Figure 7(b), $PM_{2.5}$ concentration data have bigger error and need a higher decoding ratio to do data recovery. This is because, in comparison to temperature,



(a) Temperature data (b) $PM_{2.5}$ concentration data

Fig. 8. Stability comparison of data recovery at decoding ratios of 0.1, 0.15, and 0.19 for temperature data and 0.2, 0.3, and 0.37 for $PM_{2.5}$ concentration data, respectively.

Table 1. Variances of Mean Absolute Error at Three Decoding Ratios

Temperature Data			
	0.1	0.15	0.19
ST-CNC-WSNs	0.2313	0.0254	0.0151
The Proposed	0.0017	0.0010	0.0008
$PM_{2.5}$ Concentration			
	0.2	0.3	0.37
ST-CNC-WSNs	0.7388	0.6961	0.7663
The Proposed	0.0772	0.0373	0.0278

For temperature data, the decoding ratios are set to 0.1, 0.15, and 0.19, whereas for $PM_{2.5}$ concentration data, the decoding ratios are 0.2, 0.3, and 0.37, respectively.

$PM_{2.5}$ concentration data are less correlated, as can be seen from the visual data map shown in Figure 3. But our scheme always outperforms the competing ones at all decoding ratios.

In terms of relative square error, Figure 8 illustrates the accuracy and stability of data recovery with our scheme and the ST-CNC-WSNs method through 50 repeated simulations. Here we ignore the CNS-WSNs method for simplicity since it does not consider the temporal correlation and has a worse recovery performance with regard to spatio-temporal datasets. It is easy to see from Figure 8 that in terms of both temperature and $PM_{2.5}$ concentration data, our scheme has a smaller relative square error and is much more stable than the competing ST-CNC-WSNs method at all three decoding ratios. In ST-CNC-WSNs, Kronecker structure is employed to construct a CS measurement matrix. The Kronecker-based measurement matrix can exploit spatio-temporal correlation of sensing data, but the spatial and temporal dimensions of the sensing data are in essence handled separately, degrading CS reconstruction performance. Our measurement matrix, unlike the Kronecker-based one, can directly exploit the spatio-temporal correlation of sensing data and thus has better CS reconstruction performance.

We further compute the variance of the mean absolute error at three decoding ratios for all 50 simulations, and the results are shown in Table 1. Here the maximum decoding ratios are set to

0.19 and 0.37 for temperature and air concentration data, respectively, since these two data can be 100% recovered at the corresponding decoding ratios, as illustrated in Figure 4. Compared to the ST-CNC-WSNs method, the variance values of our scheme decrease by a factor of more than 18 (0.1151/0.008) for temperature data and more than 9 (0.7388/0.0772) for $PM_{2.5}$ concentration. This also indicates that our proposed scheme has significantly higher stability than the ST-CNC-WSNs method. We believe the performance improvement should be ascribable to our proposed CS measurement matrix, since it can directly exploit both spatial and temporal correlation of sensing data. In this work, we only experimentally validate the performance of the proposed matrix but do not analyze how the matrix improves CS reconstruction mathematically. This can be our future work, which is worthy of investigation.

8 CONCLUSION

Considering the scenarios where the network is temporarily interrupted and the cloud servers cannot receive the sensing data as usual, this article presented a compressive and distributed data storage scheme for MCS systems. By introducing a virtual sensor network abstraction to a target sensing area, we formulated the local trajectories of the participants as the CS encoding processes for the entire data. Each participant stored an encoded measurement roughly reflecting the data corresponding to the whole sensing area, and the entire data field could then be stored by any m participants. We further proposed to perform encoding operations according to sensing periods but considered entire-data reconstruction across all periods, improving the recovered data. In the simulations with real temperature data and $PM_{2.5}$ concentration, we validated that our compressive storage strategy can attain 100% data recovery. The experimental results also showed that our scheme achieves better recovery quality than existing state-of-the-art schemes with traditional WSNs, without having to deploy dedicated sensors.

REFERENCES

[1] Emmanuel Candes and Yaniv Plan. 2009. Near-ideal model selection by $l1$ minimization. *Annals of Statistics* 37, 5A (2009), 2145–2177.

[2] Emmanuel Candes and Justin Romberg. 2007. Sparsity and incoherence in compressive sampling. *Inverse Problems* 23, 3 (2007), 969.

[3] Emmanuel J. Candes and Justin K. Romberg. 2005. Signal recovery from random projections. In *Proceedings of SPIE 5674, Computational Imaging III*. SPIE, San Jose, CA, 76–87.

[4] Emmanuel J. Candes and Michael B. Wakin. 2008. An introduction to compressive sampling. *IEEE Signal Processing Magazine* 25, 2 (2008), 21–30.

[5] Andrea Capponi, Claudio Fiandrino, Burak Kantarci, Luca Foschini, Dzmitry Kliazovich, and Pascal Bouvry. 2019. A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. *IEEE Communications Surveys and Tutorials* 21, 3 (2019), 2419–2465.

[6] Khanh Quoc Dinh, Hiuk Jae Shim, and Byeungwoo Jeon. 2017. Small-block sensing and larger-block recovery in block-based compressive sensing of images. *Signal Processing Image Communication* 55 (2017), 10–22.

[7] David L. Donoho. 2006. Compressed sensing. *IEEE Transactions on Information Theory* 52, 4 (2006), 1289–1306.

[8] J. E. Fowler, Sungkwang Mun, and E. W. Tramel. 2010. Block-based compressed sensing of images and video. *Foundations and Trends in Signal Processing* 4, 4 (2010), 297–416. https://doi.org/10.1561/2000000033

[9] Lu Gan. 2007. Block compressed sensing of natural images. In *Proceedings of the International Conference on Digital Signal Processing*. IEEE, Los Alamitos, CA, 403–406.

[10] Hong Gao, Xiaolin Fang, Jianzhong Li, and Yingshu Li. 2015. Data collection in multi-application sharing wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 26, 2 (2015), 403–412.

[11] Bo Gong, Peng Cheng, Zhuo Chen, Ning Liu, Lin Gui, and Frank De Hoog. 2015. Spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks. *IEEE Communications Letters* 19, 5 (2015), 803–806.

[12] Ma Huadong, Zhao Dong, and Yuan Peiyan. 2014. Opportunities in mobile crowd sensing. *IEEE Communications Magazine* 52, 8 (2014), 29–35.

[13] Saeed Karimi-Bidhendi, Jun Guo, and Hamid Jafarkhani. 2020. Energy-efficient node deployment in heterogeneous two-tier wireless sensor networks with limited communication range. *IEEE Transactions on Wireless Communications* 20, 1 (2020), 40–55.

[14] Feng Liu, Mu Lin, Yusuo Hu, Chong Luo, and Feng Wu. 2015. Design and analysis of compressive data persistence in large-scale wireless sensor Networks. *IEEE Transactions on Parallel and Distributed Systems* 26, 10 (2015), 2685–2698.

[15] Jinwei Liu, Haiying Shen, and Husnu Saner Narman. 2018. A survey of mobile crowdsensing techniques: A critical component for the Internet of Things. *ACM Transactions on Cyber-Physical Systems* 2, 3 (2018), Article 18, 26 pages.

[16] Xiao Yang Liu, Yanmin Zhu, Linghe Kong, Cong Liu, Yu Gu, Athanasios V. Vasilakos, and Min You Wu. 2015. CDC: Compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems* 26, 8 (2015), 2188–2197.

[17] Christopher A. Metzler, Arian Maleki, and Richard G. Baraniuk. 2016. From denoising to compressed sensing. *IEEE Transactions on Information Theory* 62, 9 (2016), 5117–5144.

[18] Jianbing Ni, Kuan Zhang, Qi Xia, Xiaodong Lin, and Xuemin Sherman Shen. 2020. Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing. *IEEE Transactions on Mobile Computing* 19, 6 (2020), 1317–1331.

[19] Tomer Peleg, Yonina C. Eldar, and Michael Elad. 2012. Exploiting statistical dependencies in sparse representations for signal recovery. *IEEE Transactions on Signal Processing* 60, 5 (2012), 2286–2303.

[20] Richard G. Baraniuk, Volkan Cevher, and Michael B. Wakin. 2010. Low-dimensional models for dimensionality reduction and signal recovery: A geometric perspective. *Proceedings of the IEEE* 98, 6 (2010), 959–971.
STCNC,Cstorage

[21] Ali Talari and Nazanin Rahnavard. 2016. CStorage: Decentralized compressive data storage in wireless sensor networks. *AdHoc Networks* 37, 2 (2016), 475–485.

[22] Feng Wang and Jiangchuan Liu. 2011. Networked wireless sensor data collection: Issues, challenges, and approaches. *IEEE Communications Surveys and Tutorials* 13, 4 (2011), 673–687.

[23] Leye Wang, Wenbin Liu, Daqing Zhang, Yasha Wang, En Wang, and Yongjian Yang. 2018. Cell selection with deep reinforcement learning in sparse mobile crowdsensing. In *Proceedings of the IEEE 38th International Conference on Distributed Computing Systems (ICDCS'18)*. IEEE, Los Alamitos, CA.

[24] Leye Wang, Daqing Zhang, Animesh Pathak, Chao Chen, Haoyi Xiong, Dingqi Yang, and Yasha Wang. 2015. CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp'15)*. ACM, New York, NY, 683–694.

[25] Leye Wang, Daqing Zhang, Yasha Wang, Chao Chen, Xiao Han, and Abdallah Mhamed. 2016. Sparse mobile crowdsensing: Challenges and opportunities. *IEEE Communications Magazine* 54, 7 (2016), 161–167.

[26] Leye Wang, Daqing Zhang, Dingqi Yang, Animesh Pathak, Chao Chen, Xiao Han, Haoyi Xiong, and Yasha Wang. 2018. SPACE-TA cost-effective task allocation exploiting intradata and interdata correlations in sparse crowdsensing. *ACM Transactions on Intelligent Systems and Technology* 9, 2 (2018), 20.

[27] Thakshila Wimalajeewa and Pramod K. Varshney. 2019. Application of compressive sensing techniques in distributed sensor networks: A survey. *arXiv:1709.10401* (2019).

[28] Chaocan Xiang, Zhao Zhang, Yuben Qu, Dongyu Lu, Xiaochen Fan, Panglong Yang, and Fan Wu. 2020. Edge computing-empowered large-scale traffic data recovery leveraging low-rank theory. *IEEE Transactions on Network Science and Engineering* 7, 4 (2020), 2205–2218.

[29] Chaocan Xiang, Yanlin Zhou, Haipeng Dai, Yuben Qu, Suining He, Chao Chen, and Panlong Yang. 2021. Reusing delivery drones for urban crowdsensing. *IEEE Transactions on Mobile Computing*. Early access, November 13, 2021.

[30] Xi Xu, Rashid Ansari, Ashfaq Khokhar, and Athanasios V. Vasilakos. 2015. Hierarchical data aggregation using compressive sensing (HDACS) in WSNs. *ACM Transactions on Sensor Networks* 11, 3 (2015), 45.

[31] Mingrui Yang and Frank de Hoog. 2015. Orthogonal matching pursuit with thresholding and its application in compressive sensing. *IEEE Transactions on Signal Processing* 63, 20 (2015), 5479–5486.

[32] Xianjun Yang, Xiaofeng Tao, Eryk Dutkiewicz, Xiaojing Huang, Y. Jay Guo, and Qimei Cui. 2013. Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding. *IEEE Transactions on Wireless Communications* 12, 10 (2013), 5087–5099.

[33] Quan Yuan, Haibo Zhou, Zhihan Liu, Jinglin Li, Fangchun Yang, and Xuemin Shen. 2019. CESense: Cost-effective urban environment sensing in vehicular sensor networks. *IEEE Transactions on Intelligent Transportation Systems* 20, 9 (2019), 3235–3246.

[34] Jian Zhang, Debin Zhao, and Wen Gao. 2014. Group-based sparse representation for image restoration. *IEEE Transactions on Image Processing* 23, 8 (2014), 3336–3351.

[35] Yu Zheng, Xiuwen Yi, Ming Li, Ruiyuan Li, Zhangqing Shan, Eric Chang, and Tianrui Li. 2015. Forecasting fine-grained air quality based on big data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, New York, NY, 2267–2276.

[36] Siwang Zhou, Yan He, Yonghe Liu, Chengqing Li, and Jianming Zhang. 2021. Multi-channel deep networks for block-based image compressive sensing. *IEEE Transactions on Multimedia* 23 (2021), 2627–2640.
[37] Siwang Zhou, Yan He, Shuzhen Xiang, Keqin Li, and Yonghe Liu. 2019. Region-based compressive networked storage with lazy encodings. *IEEE Transactions on Parallel and Distributed Systems* 30, 6 (2019), 1390–1402.
[38] Siwang Zhou, Shuzhen Xiang, Xingting Liu, and Yonghe Liu. 2018. Compressive networked storage with lazy-encoding. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'18).* IEEE, Los Alamitos, CA.
[39] Siwang Zhou, Qian Zhong, Bo Ou, and Yonghe Liu. 2019. Data ferries based compressive data gathering for wireless sensor networks. *Wireless Networks* 25 (2019), 675–687.

Received November 2020; revised August 2021; accepted November 2021