Stopping Criteria for Distributed Data Storage in Compressive CrowdSensing Systems

Xingting Liu, Siwang Zhou, Jiaxin Peng, Wei Zhang, Deyan Tang, and Keqin Li, *Fellow, IEEE*

***Abstract*—Distributed data storage (DDS) in mobile crowdsensing (MCS) systems has recently gained popularity. Data should be briefly saved on participants' mobile devices before being gathered once the centralized cloud servers resume normal operations. For MCS systems, the existing DDS strategies briefly considered reconstructing the scene as precisely as possible without thinking about the costs of each step. However, our goal is to obtain a sufficiently accurate approximation of the sensing data from mobile participants with as few costs as possible. We note a crucial observation: when a specified number of participants have been transmitted to a central server, the sensing data has already been well reconstructed, and the accuracy advancement with additional transmitted participants is minimal. In our scheme, two stopping criteria are proposed for DDS in compressive MCS, which aims to enhance recovery performance while reducing the costs of the whole process. In the first stopping criterion, we established a rule to stop the continued recruitment of participants. The algorithm adaptively increases the number of participants until the reconstruction accuracy meets the requirement. Another stopping criterion of the reconstruction algorithm is designed to find a more accurate number of iterations than the original. The experiment results demonstrate that the first stopping criterion can reduce participants' collection while obtaining an approximate value. The second stopping criterion assists the reconstruction algorithm in terminating at a more appropriate number of iterations, saving computing costs while ensuring accuracy.**

***Index Terms*—Adaptive participant, compressive sensing (CS), crowdsensing, distributed storage, stopping rules.**

I. Introduction

In RECENT years, mobile crowdsensing (MCS) has emerged as a promising paradigm to facilitate urban sensing applications [1], [2], including road conditions [3], mobility models [4], noise maps [5], and WiFi maps [6]. Without deploying dedicated sensor networks, the MCS systems assign multiple tasks to move participants to complete the requesters' demands in the platform. It contributes to the completion of the sensing goal by leveraging participants' mobility and sensing various urban data in regions at a low cost in order to realize large-scale distributed data collection.

Doing more with fewer measurements is encouraged by various variables, including the rivalry for limited sensor resources on a single phone or total transmission stress on network infrastructure. To address this issue, a compressive MCS paradigm in which participants perceive only a portion of the city's subareas, with the remaining data being inferred from the sensed data [7], [8], compressive sensing (CS) is a technology that holds promise for sampling and reconstructing data at a lower cost [9], [10], [11]. Crowdsensing systems can benefit from CS for low-cost sampling and highly accurate reconstruction [1].

Nowadays, massive CS-based compressive crowdsensing systems have been introduced, enlisting the assistance of mobile users to collect and submit environmental sensing data to centralized cloud servers [5], [12]. Through WiFi/4G/5G mobile network infrastructure, the corresponding sensing data collected by participants equipped with portable sensing devices are uploaded to centralized cloud servers [13], [14]. Unfortunately, the centralized cloud servers may not regularly receive the sensing data submitted by the participants due to network disruptions in certain conditions, such as an earthquake or other unforeseen events, creating a distributed data storage (DDS) problem [8], [15], [16].

Recently, several DDS strategies for MCS have been researched [15], [16], where participants must temporarily store their sensing data till the network and cloud servers are operational again. Zhou et al. [15] proposed a CS-based DDS scheme for MCS systems called DDS-MCS. By introducing a virtual sensor network abstraction to a target sensing area, the strategy formulated the local trajectories of the participants as the CS encoding processes for the entire data. The DDS-MCS provides the most information about the data field but requires the most participants to get the data, which makes it possible for additional participants to provide less improvement in accuracy. Furthermore, Zhou et al. [16] introduced a decentralized and compressive data storage approach for MCS. In the event of data recovery requirements, the process is carried out on a cloud platform equipped with abundant computational resources. However, these approaches consistently aim to optimize reconstruction accuracy without taking into account the costs of MCS.

We have an important observation that each participant stores data equally in a distributed storage scheme. Due to

Manuscript received 23 December 2022; revised 5 August 2023 and 13 September 2023; accepted 4 November 2023. Date of publication 8 November 2023; date of current version 26 March 2024. This work was supported in part by the National Science Foundation of China under Grant 62172153. *(Corresponding author: Siwang Zhou.)*

Xingting Liu, Siwang Zhou, and Jiaxin Peng are with the College of Computer Science and Electrical Engineering, Hunan University, Changsha 410082, China (e-mail: xingtingliu@hnu.edu.cn; swzhou@hnu.edu.cn; pjx@hnu.edu.cn).

Wei Zhang is with the School of Computer Science and Engineering, Changsha University, Changsha 410022, China (e-mail: zweihnu@hnu.edu.cn).

Deyan Tang is with the School of Computer Science, Hunan First Normal University, Changsha 410205, China (e-mail: deyantang@hnu.edu.cn).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/JIOT.2023.3330983

the characteristics of CS, the actual data can be reconstructed by collecting some of the stored data. By the time a certain number of participants are transmitted to the central server, the sensing data is already well reconstructed, and the accuracy increase for further transmission of participants is relatively small.

In this article, we propose two stopping criteria for DDS in MCS systems, which try to use as little costs as possible to approximate the data field as accurately as feasible. The following is a summary of this work's contributions.

1) We establish a rule to stop the continuing recruitment of participants after considering the enormous cost of data transfer from participants to the cloud. The original data can be reconstructed using the reconstruction algorithm after getting storage details from a few participants. If the accuracy is insufficient, increase the number of participants until it does, then stop recruiting participants.
2) We present the stopping criterion of the reconstruction algorithm. The stop-of-reconstruction algorithm in a distributed storage scheme has limitations, which cannot consistently achieve an excellent factual reconstruction error under different data types. We propose a new stopping criterion that can reach a more accurate error estimation than the existing methods.
3) We perform simulation experiments and contrast the results with those of other methods on two real data sets. The experiments demonstrate that our stopping algorithm may significantly reduce costs while maintaining comparable reconstruction accuracy.

The remainder of this article is organized as follows. Section II reviews the related work of storage approaches in wireless sensor networks (WSNs) and MCS. Section III briefly introduces the fundamentals of CS based on DDS in compressive crowdsensing. Section IV presents the details of our two stopping criteria. The theoretical analysis is shown in Section V, and the experimental results are discussed in Section VI. Finally, the conclusion is presented in Section VII.

II. Related Work

With the advent of the Internet of Things, a novel sensing paradigm termed MCS has emerged. This approach leverages the mobility of mobile users and the sensors integrated into mobile devices, along with the existing wireless infrastructure, to collect environmental data and enhance urban sensing efforts. Organizers of MCS programs have made maintaining data quality a priority while adhering to budgetary constraints. However, achieving this goal can be a huge challenge, as high-quality sensed data is usually costly and there is often a tradeoff between sensing quality and cost, and there usually exists a tradeoff between sensing quality and cost. Fortunately, numerous CS techniques can be employed within MCS to reduce costs without compromising data quality. These techniques take advantage of the strong spatial and temporal correlation between subareas of the sensed data, resulting in superior data collection while reducing costs. This combination of CS and MCS techniques is often referred to as CCS.

All of these research studies in CCS are predicated on the coordination of a central server to achieve comprehensive data collection on a global scale. Balancing the quality of sensing with the associated costs is a serious challenge for CCS systems. As an illustration, Yuan et al. [17] proposed an adaptive compressive data collection scheme based on matrix completion theory, but this depends on prior knowledge of the target region or the availability of historical training data sets. Wang et al. [18] introduced a task allocation framework known as CCS-TA, with the primary objective of reducing the volume of data gathered during each iteration, all the while maintaining the necessary data quality standards stipulated by the organizers. However, these CCS systems are dependent on a central server for the aggregation and processing of data, which may give rise to unforeseen incidents resulting in the temporary storage of data on participants' mobile devices, consequently giving rise to a DDS problem.

It is worth mentioning that recent research has introduced decentralized storage techniques for WSNs [19], [20], [21]. In these approaches, sink nodes deviate from the traditional practice of receiving sensed data for various specific reasons. After a sensor node generates a sensor reading, that reading is propagated throughout the network instead of being immediately uploaded to the central sink node. Each node then receives and stores the readings sent by the other nodes. Talari and Rahnavard [20] introduced a probabilistic broadcasting approach for the distribution of data, where nodes utilize CS techniques to encode the received data, effectively conserving storage space while incurring a reasonable energy cost. Zhou et al. [21] introduced an algorithm referred to as region-based compressive networked storage (CNS), with the primary goal of reducing the decoding ratio, enhancing data accuracy, and minimizing the dissemination cost. Nevertheless, it is important to note that crowdsensing operates differently from WSNs. In MCS systems, the participants equipped with sensing devices are regarded as network nodes, and the distribution of sensing data among these participants is challenging due to their mobility and unpredictable movement patterns. Consequently, the storage strategies designed for WSNs are not suitable for MCS systems.

Inspired by the decentralized storage strategy with WSNs, several DDS strategies for MCS have been researched [15], [16], where participants are required to temporarily retain their sensing data until both the network and cloud servers become operational again. Zhou et al. [15] introduced a DDS scheme for MCS systems, named DDS-MCS, which is based on CS technology. In DDS-MCS, sensing data is distributed across the network, and when the central server is back up and running, many participants send their stored data packets to the cloud to reconstruct the data. The number of participants is calculated based on empirical redundancy transmission, which enables the reconstructed data to achieve high accuracy. Further, Zhou et al. [16] introduced a decentralized and compact data storage approach for MCS. Within this method, various participants autonomously manage data storage without dependency on cloud servers for support. In the event of data recovery requirements, the process is

carried out on a cloud platform equipped with abundant computational resources.

However, these methods are excellent at improving reconstruction accuracy but fail to assign sufficient importance to the cost of MCS. In CS-based distributed storage schemes, a central server enables high-precision data reconstruction by collecting a subset of the participants' measurements. An important characteristic of CS is that, when a certain number of measurements are acquired, the effect of the additional measurements on the improvement of the accuracy significantly declines. Similarly, existing CS reconstruction algorithms exhibit this behavior in terms of the number of reconstruction iterations, which implies that the improvement in accuracy is minimal after the number of iterations exceeds a certain threshold. Obviously, increasing the number of measurements and iterations when the number of measurements and iterations is almost already saturated causes unnecessary overhead.

III. Distributed Data Storage

In distributed compressive crowdsensing systems, participants' mobile devices must temporarily store the sensing data before it can be collected once the central cloud servers are back up and running [15], [22]. Unlike direct sensing data collection, participants randomly walk around the target area to complete measurements, storing the required information on a mobile device. The measurement matrix used in reconstructing the sensed data consists of a predefined Gaussian random matrix and the participants' trajectory matrices. That is $\mathbf{A} = \mathbf{\Phi} \times \mathbf{\Psi}$, where $\mathbf{\Phi}$ is a predefined CS measurement matrix, and $\mathbf{\Psi}$ is the participant's trajectory matrix.

Participant $\boldsymbol{i}$ randomly walks each subarea and carries package $\boldsymbol{dp}_i$ to finish their walks based on the routing. Indeed, the $\boldsymbol{i}$th participant's sensing data and movement trajectory are recorded by $\boldsymbol{dp}_i$. Let $\mathbf{\Psi}_i$ be the sampling matrix that was conducted from the $\mathbf{P}_i$ trajectory. The total data for the sensing area is denoted by $\mathbf{X}$, and $\mathbf{X}_i$ denotes the partial sample result. The sampling process of $\mathbf{P}_i$ is then generalized as

$$\mathbf{X}_i = \mathbf{\Psi}_i \circ \mathbf{X}. \tag{1}$$

Let $\mathbf{\Phi}$, a predetermined CS measurement matrix, contain the $\boldsymbol{i}$th row as $\boldsymbol{\phi}_i$. Participant $\mathbf{P}_i$'s encoding process is demonstrated by using the measuring approach in CS theory as follows:

$$\boldsymbol{y}_i = \boldsymbol{\phi}_i \boldsymbol{x}_i^T \tag{2}$$

where $\boldsymbol{y}_i$ is the encoding result. The encoded measurement $\boldsymbol{y}_i$ is then stored by participant $\mathbf{P}_i$. Participant $\mathbf{P}_i$'s encoding procedure can be expressed as follows:

$$\begin{aligned} \boldsymbol{y}_i &= \boldsymbol{\phi}_i \boldsymbol{x}_i^T \\ &= \boldsymbol{\phi}_i \left(\boldsymbol{\psi}_i \circ \mathbf{X}\right)^T \\ &= \left(\boldsymbol{\phi}_i \circ \boldsymbol{\psi}_i\right) \boldsymbol{x}^T \\ &= \mathbf{A}_i \boldsymbol{x}^T \end{aligned} \tag{3}$$

where $\mathbf{A}_i = \boldsymbol{\phi}_i \circ \boldsymbol{\psi}_i$. The encoding procedure is the same for each participant. The distributed storage procedure carried out by $\boldsymbol{m}$ participants in an MCS system is represented by (3), which reads as

$$\mathbf{Y} = \mathbf{A}\mathbf{X}^T \tag{4}$$

where $\mathbf{A} = \mathbf{\Phi} \circ \mathbf{\Psi}$.

According to conventional wisdom, the main factor that makes it possible to recover signal $\mathbf{X}$ from the CS measurements $\mathbf{Y}$ relies on two objective conditions, i.e., sparsity and incoherence. Regarding a sparse transform, sparsity may exist. The data on temperature, gravity, and particulate matter (PM) air concentration that we focus on in this article are examples of real-world signals that are rarely sparse. However, they are frequently roughly scarce in a transforming domain $\boldsymbol{d}$. Define $\mathbf{S_N}$ as the transform coefficients of $\mathbf{X}$ in terms of $\boldsymbol{s}^T = \boldsymbol{d}\boldsymbol{x}^T$. Assume $\boldsymbol{s}^K$ is a vector made up of partial coefficients taken from by keeping the $\mathbf{K}$ largest coefficients and setting the rest to zero. Let $(\boldsymbol{x}^K)^T = \boldsymbol{d}^{-1}(\boldsymbol{s}^K)^T$, where $\boldsymbol{d}^{-1}$ is the inverse of matrix φ. A nearly sparse signal with power-law distributions can also be somewhat reconstructed using CS if the $\boldsymbol{i}$th most extensive entry of the modified representation fulfills

$$|\boldsymbol{x}_i| \le \boldsymbol{C}_0 \cdot \boldsymbol{i}^{-p} \tag{5}$$

for each $1 \le \boldsymbol{i} \le N$, where $\boldsymbol{C}_0$ is a constant and $\boldsymbol{p} \ge 1$. Additionally, the incoherence between the sensing matrix $\mathbf{A}$ and the transform system $\mathbf{D}$ is essential for CS. The restricted isotropy property (RIP) for an appropriately sparse signal is typically considered a prerequisite for successful CS reconstruction. Regarding the DDS scheme described before, suppose that the signal vector $\mathbf{x}$ composed of $\boldsymbol{n}$ sensor readings is a $\boldsymbol{k}$-sparse signal on representation basis $\mathbf{D}$. The sensing matrix $\mathbf{A} = \mathbf{\Phi} \times \mathbf{\Psi}$ contains two random matrices. As a result of the Metropolis-Hastings random walk, $\mathbf{\Psi}$ is an $\boldsymbol{n} \times \boldsymbol{n}$ Bernoulli random matrix. Furthermore, $\mathbf{\Phi}$ is a predefined Gauss random matrix. It has previously been demonstrated [9], [19] that if the measurement matrix has mutual coherence with the sparsity basis $\mathbf{D}$, then $\boldsymbol{s}$ can be reconstructed by solving the following:

$$\min \|\boldsymbol{s}\|_1, \qquad \text{subject to } \|\boldsymbol{y} = \boldsymbol{F} \cdot \boldsymbol{s}\|_2^2 \le \boldsymbol{\lambda} \tag{6}$$

even if $\boldsymbol{m} \ll \boldsymbol{n}$. Here, $\boldsymbol{\lambda}$ is a predefined small constant, and $\|\cdot\|_1$ and $\|\cdot\|_2$ denote 1-norm and 2-norm, respectively. Several algorithms for CS reconstruction have been investigated, including matching pursuit [23] and the iterative D-AMP algorithm [24].

IV. Stopping Criteria Strategy for MCS Systems

This section first introduces the system model of DDS in compressive crowdsensing systems. Then we elaborate on two stopping criteria, the stopping criterion for participant collection and the stopping criterion for reconstruction algorithm iteration.

A. System Model

Fig. 1 shows the general process of DDS in MCS systems. A cube is used to illustrate all sensing tasks of the target area. The cube's initial layer displays the sensing task for the present period, and the layer below shows the target area's sensing task

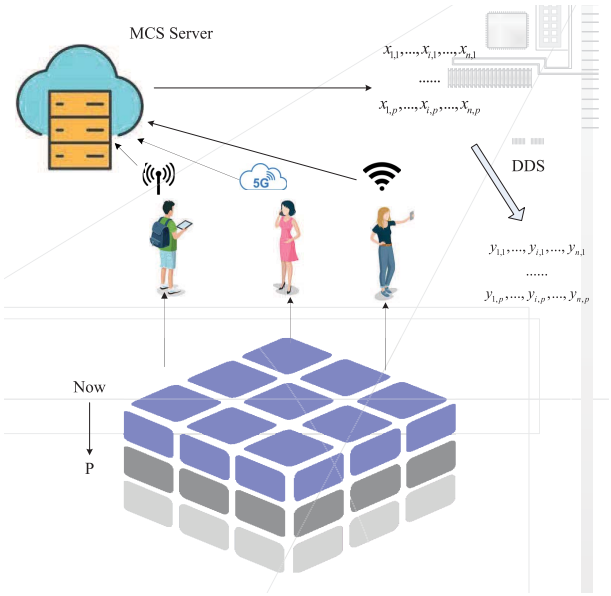


Fig. 1. Overview of the DDS process.

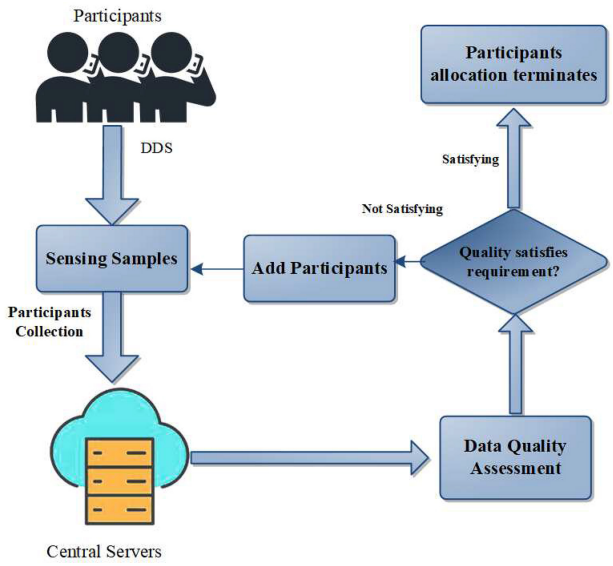


Fig. 2. Stopping criterion for participants collection.

as time passes. The MCS tasks are created by an MCS central server and allocated to participants. Each task corresponds to sensing data at a specific subarea in a particular time cycle. Participants randomly pass through some subregions of the target region at any given time and execute a DDS algorithm for each subregion passed until they exit the target region to complete a single sampling. Participants upload stored data to the cloud when the central server posts a demand. Finally, the server combines all of the sensing data from the participants to get an aggregate sensed result. In this model, we use the same storage approach as DDS-MCS and focus on collecting participant storage data from a central server to reconstruct it in the cloud.

The stopping criterion for participant collection assists the central server in adaptively collecting participants' stored data. When a certain number of participants transmit data, the reconstruction accuracy of the data reaches the requirement and the improvement of reconstruction accuracy becomes minimal by increasing the number of participants. Then the central server stops the collection. Based on this stopping rule, the costs of the participant's side are significantly saved. The stopping criterion for reconstruction algorithm iteration guides the central server side to stop the number of iterations of the reconstruction algorithm. When the algorithm iterates a certain number of times, the results of the last few iterations are analyzed, and the algorithm stops when the stopping condition is reached. With the least amount of computing overhead, the method can retrieve the most accurate reconstructed data by using this stopping rule to iterate to the correct number of stops.

B. Stopping Criterion for Participants Collection

This section describes the stopping criterion for participant collection in detail. The sensing data are stored in a distributed manner on the mobile devices of individual participants. When the central server or other devices start working, the stopping guidelines for participant collection direct the server to collect a determined number of participants.

The common stopping criterion generally considers stopping the transmission of participant data when the error between the reconstructed data and the accurate data is less than a given threshold. However, this scheme is not practical because the actual value of the information is generally not available in natural MCS systems.

MCS's two primary concerns are the participants' cost and the data quality. While DDS-MCS is mainly concerned with the quality of the reconstructed data, this article is more concerned with the overhead of the whole process. To address the cost challenge, we design a stopping criterion for participants' collection framework for DDS in compressive MCS systems, as shown in Fig. 2. When a sensing target starts, the first stage is DDS. Each participant randomly travels to a part of the subregion, and many participants jointly travel to complete the distributed storage of the whole region. In Fig. 2, four participants finish their random sample in MCS. The output of this stage is massive storage samples. Based on these sensing samples, the second stage, participants collection, attempts to collect participants' storage samples to construct a fulfilled sensing map. In the third stage, the central server conducts the data quality assessment on the complete sensing map to see whether the current data quality can meet the predefined quality requirement. If the quality requirement is not satisfied, we go back to the second stage, participants collection, to add more participants for sense; otherwise, the task allocation iteration terminates, and the framework outputs the final full sensing map for the current sensing area. Next, we introduce each step of the stopping criterion for the participant's collection.

After finishing the random walk in the target sensing area, a participant runs the encoding method and stores the encoded measurement. Fig. 2 illustrates the stopping criterion for participant collection in DDS-MCS. Block chart in Fig. 2 functions like forwarding $\boldsymbol{dp}_j$ from a subregion when participant $\mathbf{P}_j$ travels across the target sensing area while

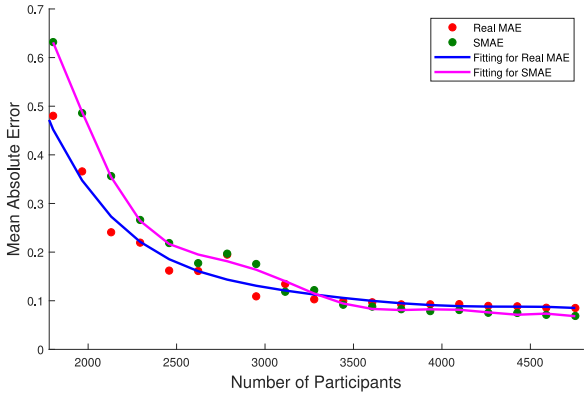


Fig. 3. SMAE and real MAE.

carrying $\boldsymbol{dp}_j$. The values of all sensor readings connected to the subareas that participant $\mathbf{P}_j$ passes through are stored in the data package $\boldsymbol{dp}_j$. In Fig. 2, four participants randomly walk through each subarea and carry package $\boldsymbol{dp}_{1,2,3,4}$ to finish their walks based on the routing which is the black line in the block chart from Fig. 2. Indeed, the $\boldsymbol{j}$th participant's movement trajectory is recorded by $\boldsymbol{dp}_j$.

When the scheme continues here, participant $\mathbf{P}_j$ stores $\boldsymbol{y}^l$ and $\boldsymbol{\psi}$ in their mobile devices. The entire sampling process $\mathbf{Y} = \mathbf{AX}$ has been completed. Then our stopping criteria comes to work. A set of participants are given sensing tasks when the central server is operational. The central server uses the stored data uploaded by these participants to reconstruct the spatiotemporal data set of the target region. As shown in Fig. 2, the central server processes the data quality assessment to see if the error reaches the stopping threshold. If the error is not satisfied, more participants are added for sense; otherwise, the iteration terminates, and the framework outputs the final full sensing map for the current sensing area. The detailed process is described in Algorithm 1.

We adopt the basic idea of stopping criterion for the participant collection algorithm under DDS crowdsensing systems. The central server first collects the storage data $\boldsymbol{y}_m$ and $\mathbf{A}_{m*n}$ of $\boldsymbol{m}$ participants. In each iteration, the stopping criterion for participant collection attempts to add some participants to upload data. Then the central server runs the reconstruction algorithm $\mathbf{R}$ using $\boldsymbol{m} + \boldsymbol{\Delta}$ participants' storage data. If the stopping mean absolute error (MAE) $\mathbf{SMAE}_{\text{times}} = [(\|\hat{\boldsymbol{x}}_{\mathbf{times+1}} - \hat{\boldsymbol{x}}_{\mathbf{times}}\|_2)/\boldsymbol{n}] < \mathbf{T_1}$, then the central server stops participant upload task and the current reconstructed data $\hat{\boldsymbol{x}}_{\text{StopDot}}$ is regarded as meeting the accuracy requirements. Otherwise, the central server randomly selects $\boldsymbol{\Delta}$ new participants to collect storage data.

We utilize a DDS approach to reconstruct actual temperature data [25]. This reconstruction serves as a practical demonstration of our stopping criterion for managing participant collection in compressive crowdsensing systems. In Fig. 3, we present results depicting the real MAE alongside the corresponding Stopping MAE (SMAE). These results are based on a grid of 128 × 128 sensing points across 16 384 observation sites. Notably, we incrementally increased the total participant count from 1638 to 4754 by 164 participants at a time.

Algorithm 1 Stopping Criterion for Participants Collection Algorithm

Input:
y_m,y_Δ: initial random collection of m, Δ participants sampling data;
$A_{m*n}, A_{\Delta*n}$; collected sensing matrix with m, Δ participants and n total sensing data;
T_1: A predefined error threshold to stop participants collect;
R: a distributed data storage scheme(e.g., DDS-MCS);
Max: Maximum number of participants.
Output:
$StopDot$: times of the participants added when the algorithm fails;
$SMAE_{stopdot}$: the Stopping Mean Absolute Error when the algorithm fails.

$y = y_m$
$A = A_{m*n}$
for *times=1 to Max* **do**
  $\hat{x}_{times} = R(y, A)$
  $SMAE_{times} = \frac{\|\hat{x}_{times+1} - \hat{x}_{times}\|_1}{n} \quad || \quad T_1$
  if $SMAE_{times} < T_1$ **then**
    $StopDot = times$
    break
  else
    $y = (y; y_\Delta)$
    $A = (A; A_\Delta)$
return $StopDot, MAE_{StopDot}$

In this context, we employed the DAMP reconstruction algorithm [24]. The figure includes fitted curves that effectively capture the relationship between the number of participants and the recovery MAE, expressed in the form of an exponential function. A key observation from Fig. 3 is the gradual convergence between the stopping error and the real error as the number of participants increases. This convergence becomes particularly evident with a sufficient number of measurements, such as the case with 3114 participants shown in Fig. 3.

Given that we lack access to the real error for the true signal, we leverage the stopping MAE as a surrogate to assess recovery MAE. When the stopping MAE falls below the required accuracy threshold, additional participants are deemed necessary. This requirement can be reasonably estimated using the fitted error function based on the participant count. The findings in Fig. 3 underscore the practical utility of our approach by highlighting how the stopping error closely approximates the real error when an ample number of participants are involved.

C. Stopping Criterion for Reconstruction Iteration

In this section, we introduce the second stopping criterion for the reconstruction algorithm, and here the reconstruction algorithm is DAMP.

Considering the stopping criterion for participant collection, we can find appropriate stopping points to maximize the number of participants uploading data, thus significantly saving the transmission overhead on the user side, which is the

Algorithm 2 Stopping Criterion for Reconstruction Iteration Algorithm

Input:
y, A: sampling signals and sensing matrix currently owned by the central server; T_2: A predefined error threshold to stop reconstruction iteration; $Rec_{oneIter}$: One iter of CS reconstruction algorithm(e.g., DAMP); *Iter*: Maximum number of iterations.
Output:
$\hat{x}$: final reconstruction results after multiple iterations.
$\hat{x}_0 = 0$
for *t=0 to Iter* **do**
  $\hat{x}_{t+1} = Rec_{oneIter}(y, A, \hat{x}_t)$
  $\hat{x}_{t+2} = Rec_{oneIter}(y, A, \hat{x}_{t+1})$
  $\hat{x}_{t+3} = Rec_{oneIter}(y, A, \hat{x}_{t+2})$
  $RSE_t = \frac{\|\hat{x}_{t+1}-\hat{x}_t\|_2}{\|\hat{x}_t\|_2} + \frac{\|\hat{x}_{t+2}-\hat{x}_{t+1}\|_2}{\|\hat{x}_{t+1}\|_2} + \frac{\|\hat{x}_{t+3}-\hat{x}_{t+2}\|_2}{\|\hat{x}_{t+2}\|_2}$
  if $RSE_t < T_2$ **then**
    break
return $\hat{x}$

main overhead for the user side. But we also increase the computational overhead of the central server, which must find the most appropriate stopping point by iterative computation many times. Based on this, we propose an iterative stopping criterion for the reconstruction algorithm, which guides the algorithm to find the most suitable stopping point to stop iteration when the reconstruction accuracy reaches the requirement. The detailed procedure can be seen in Algorithm 2.

We propose a new reconstruction iteration stopping criteria from deciding the stopping criterion for reconstructing sensing data in each target area. This stopping criterion requires that the results of the following three reconstructions all have a slight error with the results of the current reconstruction. Compared with the original stopping criterion using one iteration reconstruction results, the three-iteration reconstruction comparison has higher stability and achieves the accuracy requirements more precisely. Especially in the target region with low correlation, the sparsity of the original data needs to be increased. When the adjacent two-iteration reconstruction results reach the threshold requirement, the variance of the multiple reconstruction results is still significant, and the reconstruction accuracy is reduced. At this time, our stopping criterion can stop when each reconstruction result is stable to ensure the accuracy requirement of the reconstruction result.

The improved three-iteration form of the stop criterion outperforms the original method by enabling more precise termination of iterations. However, it does come with increased computational overhead for the $\mathbf{k}+1$ and $\mathbf{k}+2$ iterations. Thankfully, we can mitigate this burden through cloud reconstruction, utilizing its robust computing capabilities to handle this extra workload. As a result, our stopping process remains highly practical and efficient.

V. Theoretical Analysis

In Section III, we detail two stopping criteria, the stopping criterion for participant collection and the stopping criterion for reconstruction algorithm iteration. This section mathematically demonstrates that the proposed stopping criteria strategies can indicate accurate reconstruction accuracy with a predetermined number of participants in compressive crowdsensing systems.

For an appropriately sparse signal, the RIP is thought to be a prerequisite for successful CS reconstruction. However, it has only been demonstrated that RIP holds up under perfect conditions. Candes and Plan [26] describe a system whereby sensing vectors are independently sampled from a population F and demonstrate that all we need from F is an isotropy property and an incoherence property in order to guarantee successful CS reconstruction.

In the proposed DDS scheme, the measurement matrix $\mathbf{\Phi}$ is a Gaussian random matrix that meets the criteria for a successful CS reconstruction. Moreover, $\mathbf{\Psi}$ is a matrix derived from the packages $\boldsymbol{dp}$ produced by all $\boldsymbol{m}$ participants. Each row of $\boldsymbol{\psi}_j$ represents the moving trajectory of participant $\boldsymbol{P}_j$. Let $\boldsymbol{\delta}$ denote the proportion of 1 in $\boldsymbol{\psi}$, which represents the sparsity of the trajectory matrix $\boldsymbol{\psi}$. Then one has $\mathbf{A} = \mathbf{\Phi} * \mathbf{\Psi}$, and the key to becoming able to use sensing matrix $\mathbf{A}$ as a reconstruction matrix is whether it satisfies the isotropy property and incoherence property. We say that $\mathbf{A}$ holds the isotropy property if $\mathbf{E}(\mathbf{A}^\mathbf{T}\mathbf{A}) = \mathbf{I}_n$, where $\mathbf{E}(\mathbf{X})$ represents the expected value of a random matrix $\mathbf{X}$. In conformity with the isotropy property, the elements of each row vector in the measurement matrix $\mathbf{A}$ have unit variance and are uncorrelated. The coherence coefficient is calculated as the sum of the largest magnitude of the entries in A multiplied by B. The coherence coefficient $\mu(\mathbf{A})$ is defined as the sum of the largest magnitude of the entries in $\mathbf{A}$ multiplied by $\sqrt{n}$, i.e.,

$$\mu(\mathbf{A}) = \sqrt{n} \cdot \max_{i,j} |a_{ij}| \tag{7}$$

where a_{ij} is the element on the ith row and jth column of matrix $\mathbf{A}$. Incoherence implies that a smaller of coherence coefficient μ. From the research of Candes and Plan [26], if $\mu(\mathbf{A})$ is a constant of the defined range, then matrix $\mathbf{A}$ satisfies the irrelevance property. The following section will demonstrate the isotropy and incoherence properties of the measurement matrix $\mathbf{A}$ created in DDS.

Theorem 1: Let measurement matrix $\mathbf{\Phi}$ be a Gaussian random matrix that meets the criteria for a successful CS reconstruction. $\mathbf{\Psi}$ is a matrix derived from the packages $\boldsymbol{dp}$ produced by all $\boldsymbol{m}$ participants based on the Metropolis-Hastings random walk. The product of the two matrices, denoted by $\mathbf{A} = \mathbf{\Phi} * \mathbf{\Psi}$, possesses the isotropy property and incoherence property.

Proof: The matrix $\mathbf{\Phi}$ meets the condition for CS reconstruction, which means it satisfies the isotropy property and incoherence property. So $\mathbf{E}(\mathbf{\Phi}^\mathbf{T}\mathbf{\Phi}) = \mathbf{I}_n$, and $\mu(\mathbf{\Phi}) \leq \boldsymbol{c}_{\boldsymbol{\Phi}}$, where $\boldsymbol{c}_{\boldsymbol{\Phi}}$ is a constant.

In the data storage process of DDS, data dissemination from various participants perform independent random walk. Thus, the column vectors in $\mathbf{\Psi}$ are independent of each other. Each row of $\boldsymbol{\psi}_j$ represents the moving trajectory of participant $\boldsymbol{P}_j$. Let $\boldsymbol{\delta}$ represent the proportion of 1 in $\boldsymbol{\psi}$, which represents the sparsity of the trajectory matrix $\boldsymbol{\psi}$.

The sensing matrix we used for DDS is $\mathbf{A} = \mathbf{\Phi} * \mathbf{\Psi}$, we can derive

$$\begin{aligned}\mathbf{E}\left(\mathbf{A}^{\mathbf{T}}\mathbf{A}\right) &= \mathbf{E}\left((\Phi\Psi)^{\mathbf{T}}\mathbf{\Phi}\mathbf{\Psi}\right)\\ &= E\left(\mathbf{\Psi}^{\mathbf{T}}\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi}\mathbf{\Psi}\right)\\ &= \mathbf{\Psi}^{\mathbf{T}}\mathbf{E}\left(\mathbf{\Phi}^{\mathbf{T}}\mathbf{\Phi}\right)\mathbf{\Psi}\\ &= \mathbf{\Psi}^{\mathbf{T}}\mathbf{I}\mathbf{\Psi} = \mathbf{I}\end{aligned} \tag{8}$$

therefore, the sensing matrix satisfies the isotropy property.

The sensing matrix $\mathbf{A}$ satisfies the incoherence property since the nonzero portion of $\mathbf{A}$ truly originates from the $\mathbf{\Phi}$ matrix. The maximum value of coherence coefficient $\mu(\mathbf{\Phi})$ can be set as $\boldsymbol{q}$. Additionally, $\mathbf{A}$'s nonzero values occur with probability $\boldsymbol{\delta}$, which means that this matrix's range of mutual coherence coefficient is $[0, \boldsymbol{q}]$. and the mean is equal to $\boldsymbol{q\delta} \leq \boldsymbol{q}$, where $\boldsymbol{\delta} \in [0, 1]$. According to the definition of matrix $\mathbf{A}$ and $\mu(\mathbf{A})$ in (8), $\mu(\mathbf{A})$ can be expressed as

$$\begin{aligned}\mu(\mathbf{A}) &= \sqrt{n} \cdot \max_{i,j} |a_{ij}|\\ &= \sqrt{n} \cdot \max_{i,j} |\sum_{k} \phi_{ik}\psi_{kj}|\\ &\leq \sqrt{n} \cdot \max_{i} \sum_{k} |\phi_{ik}| \cdot \max_{k,j} |\psi_{kj}|\\ &\leq \sqrt{n} \cdot \boldsymbol{c}_{\mathbf{\Phi}} \cdot \boldsymbol{q}.\end{aligned} \tag{9}$$

This demonstrates that $\boldsymbol{q}$ scaling linearly with n is a sufficient condition to guarantee that $\mu(\mathbf{A})$ is most likely no more than a constant. Therefore, $\mathbf{A}$ satisfies the incoherence requirement. ■

Theorem 1 proves the sensing matrix $\mathbf{A}$, produced by the suggested storage method, theoretically meets the CS reconstruction requirement. In the following, we derive a theorem comparing the solutions with $\mathbf{M}$ participants and $\mathbf{M} + \boldsymbol{\delta}$ participants, and if they agree, we declare correct recovery. In accordance with the theory of CS, to achieve accurate reconstruction with high probability, the number of participants $\mathbf{M}$ and the signal sparsity $\boldsymbol{k}$ should satisfy $\mathbf{M} \geq c \cdot \boldsymbol{k}$. Here, c is the sampling coefficient, depending on the actual application environment. In most cases, it is impossible to get or properly estimate the sparsity of the sampled signal in practical applications, and the number of participants can only be established empirically. This requires as many participants as possible to ensure that the central server can be accurately reconstructed. The energy consumed by the participants in the transmission process is not low. To ensure accurate reconstruction, the DDS-MCS scheme has to cause too many participants to transmit a large number of redundant samples, which obviously wastes a lot of unnecessary energy. In fact, to save energy on the network, the central server should stop collecting data from participants once the signal recovery results achieve a specific accuracy or a predetermined threshold.

In this article we propose a sampling stopping criteria based on the above-designed storage sensing matrix, which aims to estimate the reconstruction accuracy of the sensing data with high accuracy from the received measurements to decide whether the central server needs to stop receiving data from the participants. In the following theorem, we show theoretically that our stopping criterion guarantees the accuracy of the reconstructed data.

Theorem 2: Suppose the $\boldsymbol{N}$-dimensional signal $\mathbf{S_N} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n]$ is the original signal which represents $\boldsymbol{N}$ sensor subregions. If the central server uses a measurement matrix of $\mathbf{\Phi}$ and receives $\mathbf{M}$ measurements from $\boldsymbol{m}$ participants. The measured value is $\mathbf{Y_M} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m]$ and the reconstructed signal is $\mathbf{S^M}$. And then add $\mathbf{\Delta}$ participants $\mathbf{L}$ times and get $\mathbf{L}$ reconstruction signals $\mathbf{S^{M+\Delta}}$, $\mathbf{S^{M+2\Delta}}$, …, $\mathbf{S^{M+L\Delta}}$. If these reconstructed signal values are equal, the probability that the reconstructed signal $\mathbf{S^M}$ is equal to the original signal satisfies the following condition:

$$\boldsymbol{P}\left(\mathbf{S^M} = \mathbf{S}\right) >= 1 - (1 - \boldsymbol{\delta}/2)^{-L}. \tag{10}$$

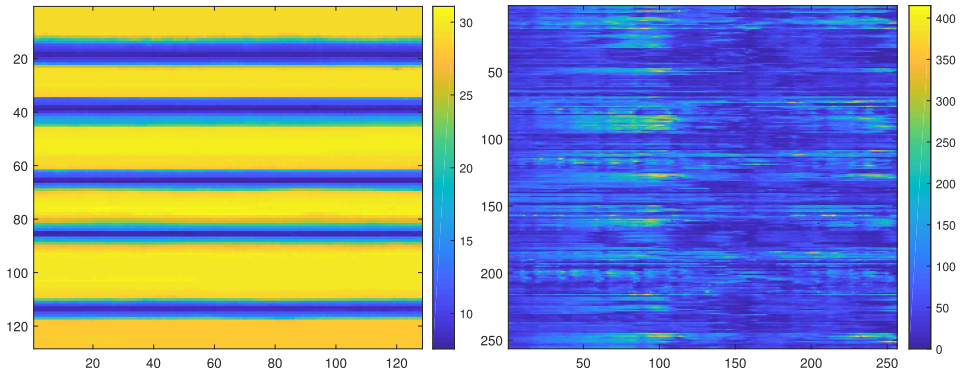
Proof: Assuming that the reconstructed signal $\mathbf{S^M}$ is not equal to the original signal $\mathbf{S}$, there exists a null-space $(\mathbf{S} - \mathbf{S^M})$, so that after the first M measurement, the equation is obtained

$$\boldsymbol{a}_M \cdot \mathbf{S} = \boldsymbol{a}_M \cdot \mathbf{S^M} \tag{11}$$

where $\boldsymbol{a}_M \in \mathbf{W}, \mathbf{W} = \boldsymbol{a}_M|(\mathbf{S}) - \mathbf{S^M}) \cdot \boldsymbol{a}_M = 0, \boldsymbol{a}_M \neq 0$. Obviously, $\mathbf{W}$ is both a null-space and an $\mathbf{N} - \mathbf{1}$ dimensional subspace of an $\mathbf{N}$-dimensional field of real numbers $\boldsymbol{R}^N$. After adding $\mathbf{\Delta}$ participants, the central server can obtain measurement values $\mathbf{Y}_{M+\Delta}$. As $\boldsymbol{a}_{M+\Delta}$ is random and independent of $\mathbf{S^M}$ and of the previous samples if it satisfies $(\mathbf{S} - \mathbf{S^M}) \cdot \boldsymbol{a}_{M+\Delta} = 0$, then the $\mathbf{M}$th reconstructed signal is also applicable as the result of the $(\mathbf{M} + \Delta)$th reconstructed, also $\boldsymbol{P}(\boldsymbol{a}_{M+\Delta} \in \mathbf{W}) \approx \boldsymbol{\delta}/2$. By parity of reasoning, when satisfying $\mathbf{S^M} = \mathbf{S^{M+\Delta}} = \mathbf{S^{M+2\Delta}} = \cdots = \mathbf{S^{M+L\Delta}}$, the lower bound on the probability that $\mathbf{S^M}$ and $\mathbf{S}$ are equal is $1 - (1 - \boldsymbol{\delta}/2)^{-L}$. ■

This theorem allows us to stop collecting participants when a feasible solution has $\mathbf{M}$ nonzero elements or in sparse form has $\mathbf{M}$ nonzero elements. The sensing matrix $\mathbf{A}$, formed by the proposed storage strategy, satisfies the condition of CS reconstruction in theory for the randomness of the movements of the participants. Based on the irrelevance between the sensing matrix and sensing data, the reconstructed value generated by increasing the measurement several times $\mathbf{S^M}, \mathbf{S^{M+\Delta}}, \mathbf{S^{M+2\Delta}}, \ldots, \mathbf{S^{M+L\Delta}}$ varies very little. Thus, one can say $\mathbf{P}(\boldsymbol{a}_{M+\Delta} \in \mathbf{W}) \approx 1$ [27], [28].

Our two stopping criteria are supported by Theorem 2. It is possible to determine whether to continue or stop sampling based on predetermined constraints, which gives the stopping criterion for participants collection theoretical support. Theorem 2 also evaluates the probability that the reconstructed result is equal to the original signal. The reconstructed results obtained by several successive incremental participants are equal, then the higher probability reconstructed results are equal to the original signal. Based on this principle, the stopping criterion for reconstruction iteration can assist us to pick a better number of iterations to make the reconstructed signal closer to the original signal.



(a) (b)

Fig. 4. Visual maps of two test data sets of size 128×128. (a) Temperature. (b) $PM_{2.5}$.

VI. Performance Evaluation

In this section, we first introduce the evaluation scenario and parameters for performance evaluation. Then, the simulation results are given to evaluate the efficiency and effectiveness of stopping criteria. Finally, we compare our algorithms with the DDS-MCS presented in [15]. The simulation establishes a credible experimental setting with a quality experimental design. The Metropolis-Hastings random walk [19], [21] is adopted to simulate the movement pattern of participants, and the movement characteristics of participants can be truly reflected by adjusting the movement probability.

A. Evaluation Scenario

Without loss of generality, we construct a simulated MCS network where the target sensing area contains $\mathbf{S}$ observation sites. Each observation site is uniformly distributed in a unit disc or square, generating a continuous stream of time series data. Massive participants walk around with their mobile sensing devices, gathering and storing sensing data as they pass through these $\mathbf{S}$ sites. The experiment data contain temperature data from the National Data Buoy Center [25], also the PM2.5 data from China National Environmental Monitoring Centre [29]. The two data types are all real-life data sets but have different data correlations. Fig. 4 displays the visual maps of the simulation's temperature and $PM_{2.5}$ concentration data. As can be seen from the figure, the temperature data range is [7.42, 31.08], and the poorly correlated data is $PM_{2.5}$ data, where the data range is [0,415].

Suppose the target sensing area is $\mathbf{R} \times \mathbf{C} \times \mathbf{T}$ size, where $\mathbf{R} \times \mathbf{C}$ denotes two spatial dimensions and $\mathbf{T}$ is the temporal dimension. And $\boldsymbol{x}$ represents the original spatial-temporal data, $\hat{\boldsymbol{x}}$ represents the reconstructed stored data of $\boldsymbol{x}$ in the center server. In this scenario, $\mathbf{T}$ sensing cycles are divided into $\boldsymbol{t}$ storage cycles. Let $\boldsymbol{n} = \mathbf{R} \cdot \mathbf{C} \cdot \mathbf{T}$ and $\boldsymbol{m} = \sum_{l=1}^{t} \boldsymbol{m}_l$, where $\boldsymbol{n}$ represents the length of the data field being recovered and $\boldsymbol{m}_l$ represents the number of CS measurements in the $\boldsymbol{l}$th sensing period of the storage process. Throughout the simulations, we list the key simulation parameters and give some performance indexes to evaluate the experiment's performance. There are 64 observation sites with a spatial 8×8 grid setup. 256 time-series data are generated for each observation site. The decoding ratio is set as

$$\boldsymbol{dr} = \frac{\boldsymbol{m}}{\boldsymbol{n}}. \tag{12}$$

It represents the reconstruction capability of the adopted DDS scheme. Then the decoding ratio can be seen as a performance metric of data storage. Additionally, the mean absolute and relative square errors are used to assess the accuracy of the target area's recovered data field. The MAE $\mathbf{SMAE}(\boldsymbol{x_{t+1}}, \boldsymbol{x_t})$ and relative square error $\mathbf{SRSE}(\boldsymbol{x_{t+1}}, \boldsymbol{x_t})$ are given by

$$\mathbf{SMAE}(\boldsymbol{x_{t+1}}, \boldsymbol{x_t}) = \frac{\|\boldsymbol{x_{t+1}} - \boldsymbol{x_t}\|_1}{\boldsymbol{n}} \tag{13}$$

$$\mathbf{SRSE}(\boldsymbol{x_{t+1}}, \boldsymbol{x_t}) = \frac{\|\boldsymbol{x_{t+1}} - \boldsymbol{x_t}\|_2}{\|\boldsymbol{x_t}\|_2} \tag{14}$$

respectively. Here, $\|\cdot\|_1$ is 1-norm and $\|\cdot\|_2$ denotes 2-norm. Each participant begins to walk from a random observation site $(\boldsymbol{r}, \boldsymbol{c})$ at a random time $\boldsymbol{t}$ and reach another observation site over $\boldsymbol{s}$ steps where $\boldsymbol{s} \in [200, 500]$.

The entire computation is carried out on a server platform equipped with two 3.2-GHz Intel CPUs and 256 GB of memory using the MATLAB R2018b simulator. In order to conduct a fair comparison, the data field stored by both our storage strategy and the rival one is recovered using the identical CS reconstruction algorithm, D-AMP [24] combined with a BM3D denoiser and BIOR 1.5 wavelet transform.

B. Feasibility of the Stopping Criteria

In this part, we show the experimental results of two stopping criteria to prove the feasibility of the stopping algorithm. The primary metrics used to measure the performance of the stopping criteria include the reconstruction accuracy when the stop condition is met, the number of participants at the stopping point, and the performance of the stopping criterion for reconstruction iteration.

1) Reconstruction Accuracy: A series of experiments were conducted to verify the method's efficiency and the performance of two stopping criteria. In this part, we verify the effectiveness of the stopping criteria. An important fact is that we cannot take the actual data as prior knowledge, so the original formula for calculating the error is not practical.

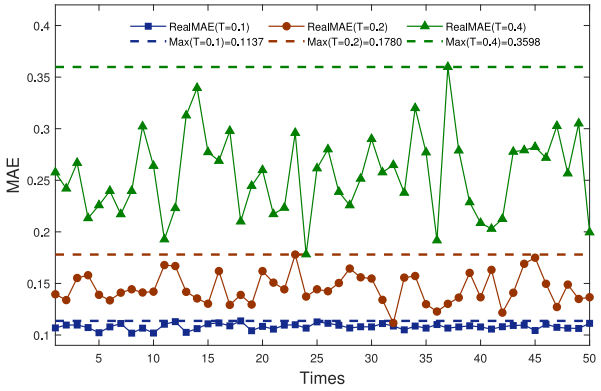


Fig. 5. Real error of reconstructed data under different thresholds for temperature.

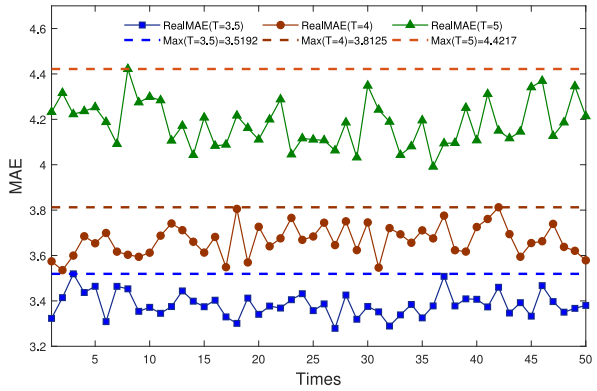


Fig. 6. Real error of reconstructed data under different thresholds for $PM_{2.5}$.

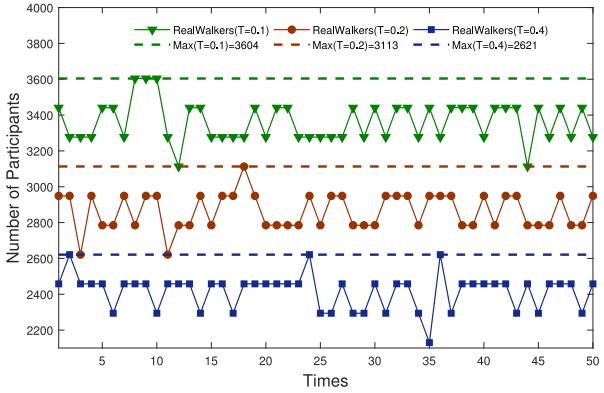


Fig. 7. Number of walkers under different threshold for temperature.

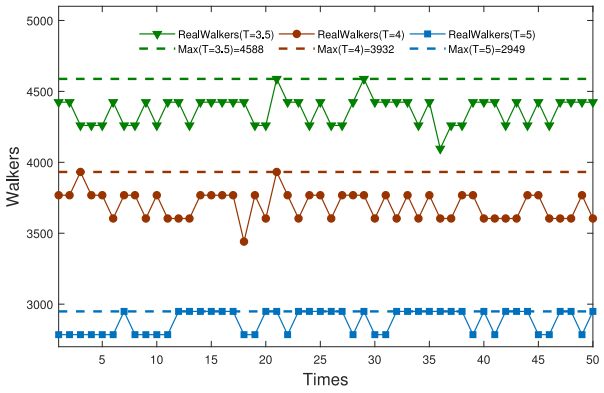


Fig. 8. Number of walkers under different threshold for $PM_{2.5}$.

Based on this, we describe our method for calculating the error as our stopping error. The algorithm stops when the increase of participants is less than a predetermined threshold $\mathbf{T_1}$. **SMAE** and **SRSE** are two utterly different evaluation error indicators. **SMAE** can evaluate the error intuitively, but only for comparing the same unit, while **SRSE** is not intuitive but can compare the error of different units. Without loss of generality, **SMAE** $<$ $\mathbf{T_1}$ is used as the index to measure the algorithm stopping error, but **SRSE** is used to compare the stopping error performance of different types of data.

Figs. 5 and 6 show the factual errors our stopping criteria can achieve at different predetermined thresholds. We choose an initial number of participants to be collected and add a small number of individuals to the process until the stopping MAE **SMAE** is smaller than the predefined threshold $\mathbf{T_1}$. Real sensing data is used for testing. Combining real and reconstructed data, the real reconstructed error can be calculated. The vertical value displays the actual error in reconstructing the data under the current repetition, and the horizontal coordinate indicates how many times the method is performed.

The real **MAE** of reconstructed data under different temperature thresholds is displayed in Fig. 5, where we set the stop conditions to 0.1, 0.2, and 0.4, respectively. Under these conditions, 50 repeated simulations were completed. The max real **MAE** of the 50 repeated were, respectively, 0.1137, 0.1780, and 0.3598. These maximum **MAE** represent the maximum error between the reconstructed data and the actual data at each stop for 50 repetitions. For example, when $\mathbf{T_1} = 0.1$, it means that under our stopping criteria, the difference between the reconstructed and actual temperature data is at most 0.1137 °C.

Fig. 6 shows the real **MAE** for $PM_{2.5}$ data under different thresholds. Also, 50 repeated simulations were performed. For instance, when $\mathbf{T_1} = 3.5$, the difference between the reconstructed $PM_{2.5}$ data and the actual $PM_{2.5}$ data are, at most 3.5192 μ/m^3. From the simulation result, we can find that with each predefined threshold, The result of sensing data reconstruction is consistently below a particular error compared with the real sensing data. And because $PM_{2.5}$ data is less correlated, data reconstruction accuracy is also lower.

2) Number of Participants: The reconstruction accuracy achieved under the stop condition serves as a critical parameter to validate the effectiveness of the stop criterion. As the desired accuracy increases, a larger number of participants are required to ensure reliable results. Additionally, the minimum number of participants needed to achieve the desired accuracy is a key indicator of the stopping algorithm's performance.

Figs. 7 and 8 present the number of walkers under different temperatures and PM2.5 thresholds, respectively. Each simulation was repeated 50 times to ensure robustness. The maximum number of walkers required among the 50 repetitions was 2621, 3113, and 3604 for temperature data at threshold values of 0.1, 0.2, and 0.4, respectively. Similarly, for PM2.5 data, the maximum number of walkers needed was 2939, 3932,

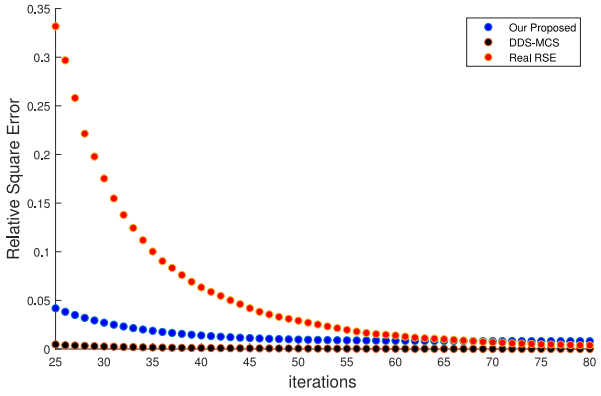


Fig. 9. Comparison of three reconstruction errors of the CS algorithm with different number of iterations.

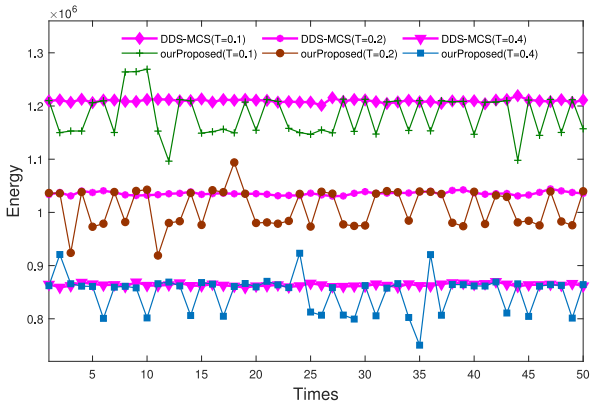


Fig. 10. Energy cost compare for temperature.

and 4588 when the threshold values were 3.5, 4, and 5, respectively. It is worth noting that air concentration data (such as $PM_{2.5}$) exhibit lower correlations compared to temperature data, which leads to a greater need for more participants to achieve the desired reconstruction accuracy.

Without the use of our stop algorithm, only the participant with the highest point could be selected for measurement each time to maintain accuracy below the given threshold T. However, our proposed stop algorithm ensures that the required accuracy is achieved while optimizing the number of participants needed, making the overall process more efficient and effective.

3) Performance of Stopping Criterion for Reconstruction Iteration: Both our scheme and DDS-MCS adopt D-AMP as the CS reconstruction algorithm in light of its superior performance. In DDS-MCS, the stopping criterion for reconstruction iteration is $(\|\hat{x}_{t+1} - \hat{x}_t\|_2/\|\hat{x}_t\|_2)$. This stopping criterion aids the reconstruction algorithm in halting at the appropriate proportions of iterations since in practice the original signal is unavailable. However, as can be observed from Theorem 2, comparing simply the reconstruction outcomes of the tth and $(t+1)$th times does not ensure the true reconstruction error when the created sensing matrix $\mathbf{A}$ is somewhat sparse, i.e., when δ is quite small. Our system takes this into consideration while comparing the reconstructed results from the previous three iterations, where the stopping condition is $\text{RSE}_{\text{stop}} = (\|\hat{x}_{t+1} - \hat{x}_t\|_2/\|\hat{x}_t\|_2) + (\|\hat{x}_{t+2} - \hat{x}_{t+1}\|_2/\|\hat{x}_{t+1}\|_2) + (\|\hat{x}_{t+3} - \hat{x}_{t+2}\|_2/\|\hat{x}_{t+2}\|_2) \leq T_2$. The probability that the reconstructed signal $\mathbf{x_t}$ is equal to the original signal $P(\mathbf{x_t} = \mathbf{x})$ may be extremely near 1 even if δ is quite small after our stopping criterion for reconstruction iteration.

Fig. 9 gives the results of the stopping criterion for reconstruction iteration in terms of the relative square error. The experiment data set is temperature data and the decoding rate dr is set to 0.16 for previous experiments. Three error lines, including the real reconstruction error, DDS-MCS stopping error, and our stopping error are plotted. Fig. 9 also clearly indicates that the difference between two estimation errors and real error decreases as iterations increase. When there are enough iterations, two estimation errors and real error begin to be coherent with each other very well. However, our proposed error line is a better fit to the real error than the DDS-MCS error line, especially when the number of iterations is not enough. Thus, we can use our stopping error estimation to evaluate the recovery error for its excellent performance.

C. Comparison With DDS-MCS

This section contrasts our scheme with the most recent technique, DDS-MCS, while using identical network setup and parameter settings. With the same network configuration as used earlier, the spatial 8×8 grid configuration has 64 observation sites. Each observation site generates 256 time-series data. As a consequence, we can concentrate on evaluating the effect of the stopping criteria algorithm without being disturbed by other factors.

1) Comparison With Stopping Criterion for Participants Collection: We should note that energy cost is a very important index in resource-constrained MCS networks [30]. In this article, we utilized the same storage method as DDS-MCS, but we also took into account the energy required to acquire and rebuild data from storage nodes. So the energy consumption of this step is decided in two parts, participants acquire cost from each participant and data reconstruct cost in the central server. The energy cost in our step is

$$E_{\text{total}} = E_{\text{participants}} + E_{\text{cloud}} \tag{15}$$

where $E_{\text{participants}}$ represents the energy required by the participants and E_{cloud} is the amount of energy required by the central server to reconstruct data. Considering the central server's powerful computing and transmission capacity, we focus our solution on the participants' energy consumption overhead. In DDS scheme, participants' walking and sensing is the most energy-consuming step. The energy consumption of this step is decided by two parameters in data storage, i.e., the number of participants m and the steps s. We define the energy required for a participant to move once and collect data once as the unit e, which is convenient but not unfair. The energy cost in the data storage step on the participants' side is

$$E_{\text{participants}} = m \cdot s \cdot e. \tag{16}$$

Figs. 10 and 11 compare the energy consumption of all participants in the storage phase of the two data sets, including

TABLE I
MEAN OF ENERGY COSTS FROM DIFFERENT STOPPING THRESHOLDS

Method	Temperature($10^6/times$)			$PM_{2.5}$($10^6/times$)		
	T≤ 0.1	T≤ 0.2	T≤ 0.4	T≤ 3.5	T≤ 4	T≤ 5
DDS-MCS	1.2097	1.0357	0.8635	3.5447	3.0188	2.3628
Our Proposed	1.1819	1.0070	0.8482	3.4966	2.9643	2.3070
Improve	2.35%	2.85%	1.80%	1.38%	1.84%	2.42%

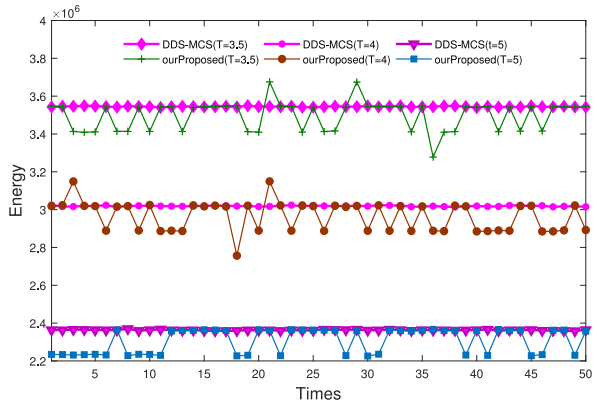


Fig. 11. Energy cost compare for $PM_{2.5}$.

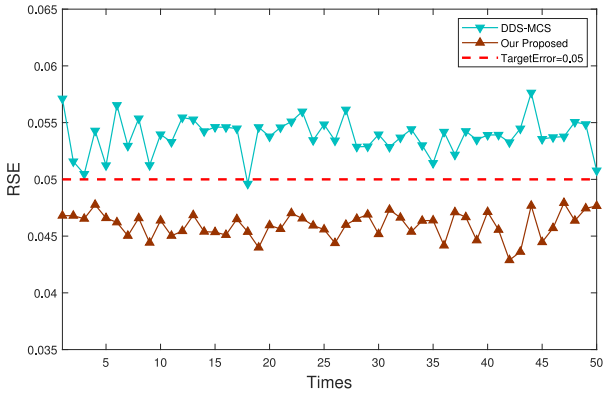


Fig. 12. RSE compare of reconstruction iterations for temperature.

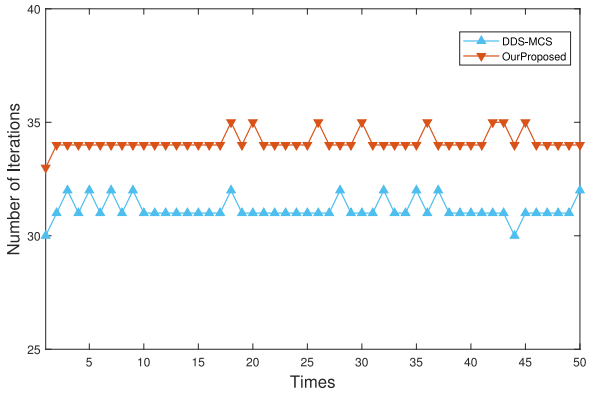


Fig. 13. Iterations compare of reconstruction iterations for temperature.

temperature data and $PM_{2.5}$ data in the distributed storage under the MCS. For each of the two data iterations of 50 times, these two figures depict the energy overhead required by the two stopping criterion algorithms for different accuracy requirements. The horizontal coordinate indicates the number of repetitions and the vertical coordinate indicates the total energy overhead of the participants based on (14), where $\boldsymbol{e}$ is a fixed constant. Regardless of the data set, even though our proposed stopping algorithms require slightly more energy overhead in some of these rounds, generally, our energy costs are significantly lower than DDS-MCS.

We further compute the mean of the energy costs at three stopping thresholds for all 50 simulations and the results are shown in Table I. Just like in Figs. 5 and 6, the stopping thresholds for temperature are set to 0.1, 0.2, and 0.4, and for $PM_{2.5}$, they are set to 3.5, 4, and 5, respectively. Compared to the DDS-MCS method, on average, our system improves energy costs by 2%.

2) Comparison With Stopping Criterion for Reconstruction Iteration: The performance of stopping criteria for participant collection is evaluated in terms of reconstruction accuracy and the number of participants. Then in this section, we assess the performance of stopping criteria for reconstruction iterations. Figs. 12 and 13 list the temperature results compare to our stopping criteria with the original RSE results and a number of iterations. In Fig. 12, we compare our proposed stopping criteria with the DDS-MCS scheme for **RSE** results until **RSE** $\leq$ 0.05. From Fig. 12, although the actual **RSE** error of our proposed and DDS-MCS are all close to 0.05. Our algorithm gets a minor error closer to 0.05 when the stop condition is satisfied. Fig. 13 lists the number of iterations the algorithm requires when the stop condition is reached, which costs the central server's computing resources. The central server has relatively strong computing power. Although our stopping criteria require three more iterations on average, the added computing overhead for the central server is negligible. Still, the stopping point is more accurate than the original stopping point.

VII. CONCLUSION

This article proposes two stopping criteria to keep the efficiency and effectiveness of compressive crowdsensing systems for DDS. The stopping criterion for participant collection can implement high-precision data reconstruction while reducing the number of participants when the original signal is unknown to a central server. The stopping criterion for reconstruction iteration is then used to conduct the reconstruction algorithm to stop iteration after a rational number of iterations. Compared with the previous stopping criteria, it can stop closer to the actual error. The experiment results show that the first stopping criterion effectively reduces participant collection while still

obtaining an approximate value. The second stopping criterion assists the reconstruction algorithm in stopping at a more appropriate number of iterations, saving computing costs and ensuring accuracy.

REFERENCES

[1] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 161–167, Jul. 2016.

[2] L. Xiao, D. Jiang, D. Xu, W. Su, N. An, and D. Wang, "Secure mobile crowdsensing based on deep learning," *China Commun.*, vol. 15, no. 10, pp. 1–11, Oct. 2018.

[3] D. Santani et al., "CommuniSense: Crowdsourcing road hazards in Nairobi," in *Proc. 17th Int. Conf. Human-Comput. Interact. Mobile Devices Services*, 2015, pp. 445–456.

[4] Z. Zhu, B. Chen, W. Liu, Y. Zhao, Z. Liu, and Z. Zhao, "A cost-quality beneficial cell selection approach for sparse mobile crowdsensing with diverse sensing costs," *IEEE Internet Things J.*, vol. 8, no. 5, pp. 3831–3850, Mar. 2021.

[5] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw.*, 2010, pp. 105–116.

[6] R. Becker et al., "Human mobility characterization from cellular network data," *Commun. ACM*, vol. 56, no. 1, pp. 74–82, 2013.

[7] X. Hao, L. Xu, N. D. Lane, X. Liu, and T. Moscibroda, "Density-aware compressive crowdsensing," in *Proc. 16th ACM/IEEE Int. Conf. Inf. Process. Sens. Netw. (IPSN)*, 2017, pp. 29–40.

[8] P. Sun, Z. Wang, L. Wu, H. Shao, H. Qi, and Z. Wang, "Trustworthy and cost-effective cell selection for sparse mobile crowdsensing systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 6108–6121, Jun. 2021.

[9] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[10] S. Zhou, Y. He, Y. Liu, C. Li, and J. Zhang, "Multi-channel deep networks for block-based image compressive sensing," *IEEE Trans. Multimedia*, vol. 23, pp. 2627–2640, 2020.

[11] S. Zhou, X. Deng, C. Li, Y. Liu, and H. Jiang, "Recognition-oriented image compressive sensing with deep learning," *IEEE Trans. Multimedia*, vol. 25, pp. 2022–2032, 2022.

[12] L. Han, Z. Yu, L. Wang, Z. Yu, and B. Guo, "Keeping cell selection model up-to-date to adapt to time-dependent environment in sparse mobile crowdsensing," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 13914–13925, Sep. 2021.

[13] E. Wang et al., "Outlier-concerned data completion exploiting intra-and inter-data correlations in sparse crowdsensing," *IEEE/ACM Trans. Netw.*, vol. 31, no. 2, pp. 648–663, Apr. 2023.

[14] L. Chettri and R. Bera, "A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems," *IEEE Internet Things J.*, vol. 7, no. 1, pp. 16–32, Jan. 2020.

[15] S. Zhou, Y. Lian, D. Liu, H. Jiang, Y. Liu, and K. Li, "Compressive sensing based distributed data storage for mobile crowdsensing," *ACM Trans. Sens. Netw.*, vol. 18, no. 2, pp. 1–21, 2022.

[16] S. Zhou, X. Zhang, Y. Liu, H. Jiang, and K. Li, "Decentralized and compressed data storage for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, early access, Jul. 13, 2023, doi: 10.1109/TMC.2023.3294969.

[17] Q. Yuan, Z. Liu, J. Li, S. Yang, and F. Yang, "An adaptive and compressive data gathering scheme in vehicular sensor networks," in *Proc. IEEE Int. Conf. Parallel Distrib. Syst.*, 2015, pp. 207–215.

[18] L. Wang, D. Zhang, A. Pathak, C. Chen, and Y. Wang, "CCS-TA: Quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput.*, 2015, pp. 683–694.

[19] F. Liu, M. Lin, Y. Hu, C. Luo, and F. Wu, "Design and analysis of compressive data persistence in large-scale wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 10, pp. 2685–2698, Oct. 2015.

[20] A. Talari and N. Rahnavard, "CStorage: Decentralized compressive data storage in wireless sensor networks," *Ad Hoc Netw.*, vol. 37, pp. 475–485, Feb. 2016.

[21] S. Zhou, Y. He, S. Xiang, K. Li, and Y. Liu, "Region-based compressive networked storage with lazy encoding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1390–1402, Jun. 2019.

[22] B. Gong, P. Cheng, Z. Chen, N. Liu, L. Gui, and F. De Hoog, "Spatiotemporal compressive network coding for energy-efficient distributed data storage in wireless sensor networks," *IEEE Commun. Lett.*, vol. 19, no. 5, pp. 803–806, May 2015.

[23] M. Yang and F. De Hoog, "Orthogonal matching pursuit with thresholding and its application in compressive sensing," *IEEE Trans. Signal Process.*, vol. 63, no. 20, pp. 5479–5486, Oct. 2015.

[24] C. A. Metzler, A. Maleki, and R. G. Baraniuk, "From denoising to compressed sensing," *IEEE Trans. Inf. Theory*, vol. 62, no. 9, pp. 5117–5144, Sep. 2016.

[25] X. Xu, R. Ansari, A. Khokhar, and A. V. Vasilakos, "Hierarchical data aggregation using compressive sensing (HDACS) in WSNs," *ACM Trans. Sens. Netw.*, vol. 11, no. 3, pp. 1–25, 2015.

[26] E. J. Candes and Y. Plan, "A probabilistic and RIPless theory of compressed sensing," *IEEE Trans. Inf. Theory*, vol. 57, no. 11, pp. 7235–7254, Nov. 2011.

[27] M. Leinonen, M. Codreanu, and M. Juntti, "Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 14, no. 3, pp. 1622–1635, Mar. 2015.

[28] D. M. Malioutov, S. R. Sanghavi, and A. S. Willsky, "Sequential compressed sensing," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 2, pp. 435–444, Apr. 2010.

[29] Y. Zheng et al., "Forecasting fine-grained air quality based on big data," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min.*, 2015, pp. 2267–2276.

[30] X. Chen and G. Liu, "Energy-efficient task offloading and resource allocation via deep reinforcement learning for augmented reality in mobile edge networks," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10843–10856, Jul. 2021.