Energy-efficient Edge Computing Framework for Decentralized Sensing in WSN-assisted IoT

Vini Gupta and Swades De

Abstract

This paper addresses the problem of decentralized sensor selection in an energy-constrained wireless sensor network-based Internet-of-Things, for monitoring a spatio-temporally varying process. To do so, an adaptive edge computing framework and its variants are proposed which distributedly optimize a critical trade-off between sensing quality and remaining energy of the sensor nodes (SNs). Unlike the existing distributed sensing approaches, the proposed one aims to maintain energy balance among the SNs. The original sensor selection problem is decoupled into multiple sub-problems, each solvable at an edge node elected as head of a coverage region containing a set of SNs. The sub-problem in each coverage region is adapted to variations of the underlying process. In each region, the process is estimated using PCA-SBL (principal component analysis-sparse Bayesian learning) on noisy signal measured by the respective active SNs. Further, to correctly adapt to the process and estimate the signal, a novel logic is designed that indicates requirement of network retraining in the next measurement cycle. The results from extensive simulation studies illustrate improved energy efficiency and network energy balance of the proposed framework over the existing closest competitive centralized and decentralized approaches. The proposed framework is tested on synthetic as well as real data-sets of a sensor network.

Index Terms

Wireless sensor networks, adaptive and decentralized sensor selection, sparse Bayesian learning, principal component analysis, network residual energy

I. Introduction

Rapid growth of smart wireless sensors deployment for Internet-of-Things (IoT) applications is expected to exponentially increase the energy consumption and delay [1]. In this context, intelligent sensing and transmission at the field sensor nodes (SNs) can prolong their life span [2], [3]. In a densely-deployed wireless sensor network (WSN) for monitoring a spatio-temporal

V. Gupta and S. De are with the Department of Electrical Engineering and Bharti School of Telecommunication, Indian Institute of Technology Delhi, New Delhi 110016, India (e-mail: {vini.gupta, swadesd}@ee.iitd.ac.in).

process, inherent redundancy in the signal can be exploited to perform parsimonious sensing without sacrificing the sensing quality [4]. Further, decentralized processing at the edge nodes can overcome the scalability issues.

A. Prior art and motivation

The existing works on sensing mainly focus on centralized decisioning wherein the fusion center (FC) receives information (noisy measurements) from the SNs, takes sensing decision, and conveys it back to the SNs. The approaches in [5], [6] randomly activate a fixed number of SNs without paying attention to energy efficiency. A greedy selection approach in [7] aimed at improved energy efficiency. The aspect of sensing quality has gained recent attention. For instance, in [8] a sensor selection scheme was presented for a linear measurement model using a well known D-optimal performance criterion. In [9] the Cramér-Rao lower bound performance measure was employed as sensor selection criterion for a non-linear measurement model. For monitoring a temporally-correlated source, in [10] a sensor selection scheme was developed that optimizes the sensing quality subject to a prescribed power budget for the energy harvesting (EH) system. Subsequently, in [11] a sensor selection scheme was proposed that minimizes the estimation error at the FC subject to energy and spectral budgets of an EH IoT network. The work in [12] optimizes both sensing quality and energy efficiency while selecting a fixed number of SNs. To address the shortcoming of sensing a dynamic process by activating a fixed number of SNs, centralized sensing schemes in [13]–[16] adapted the number of active sensors.

The centralized sensor selection approaches involve high communication and energy overheads and processing complexity. To overcome these limitations in large-scale WSNs, decentralized strategies have been developed in recent studies. An offline iterative distributed sensor selection scheme was proposed in [17], [18], employing dual subgradient method. Likewise, the study in [19] suggested an optimization problem for decentralized sensor selection and solved using a fast empirically converging iterative alternating direction method of multipliers. Distributed implementation of the approach in [8] was formulated in [20], in a network with two leader nodes. Further, considering a Bayesian framework a low-complexity greedy decentralized algorithm for heterogeneous sensing environment was developed in [21].

Iterative nature of the existing distributed sensor selection approaches [17]–[21] requires local information exchange among the SNs in each iteration, which significantly increases the energy and processing overheads. These works assume regression coefficients to be known a priori,

which may not be realistic. Moreover, these approaches did not account for the critical aspect of unequal remaining energy of the SNs. Consequently, a few SNs may be repeatedly selected over the others, which may result in a network coverage outage. Also, importantly, these approaches are non-adaptive to the dynamics of the sensed process; they fix one of the two performance measures (sensing quality and the number of active SNs) and optimize the other. Intuitively, optimizing both the measures as per the process dynamics and minimum required sensing quality may yield a better performance.

B. Novelty and Contribution

To overcome the above-mentioned lacuna in the literature, this work proposes an adaptive edge computing framework for decentralized sensor selection that jointly optimizes the sensing quality of a slowly-varying process and WSN energy efficiency. To the best of our knowledge this framework is first to provide energy balance among the SNs by accounting for their unequal remaining energy in the sensor selection problem. The existing works considered distributed sensor selection and remaining energy in isolation, without factoring that the SNs can be selected for sensing only if they have sufficient remaining energy. In contrast to the prior art, the proposed framework accounts for energy consumption at each step and does not require iterative local information exchange in each cycle. Another important aspect is that the proposed framework learns the regression matrix online using previously estimated signals to better adapt to spatio-temporally evolving process, unlike a random Gaussian matrix considered in the literature. To keep a check on accumulated estimation error, a retraining logic is also developed. In an edge computing framework, entire processing is done by the edge nodes in contrast to the cloud computing where the central entity performs all such computations. In network's context, the centralized sensor selection is analogous to the cloud computing. Similarly, for decentralized sensor selection, this paper considers edge computing framework that is realized by dividing the WSN field in coverage regions/clusters with an elected cluster head (CH) in each. The CHs perform the role of edge nodes by taking sensing decisions and sharing them with the cloud (FC) and the field SNs. Thus, the novel ideas are decoupling the sensing quality and energy efficiency-based sensor selection problem into various sub-problems, integration of energy consumption, non-iterative decentralized sensing decisions, and data-driven retraining of the network/coverage regions(s). The contributions of this work are as follows.

1) A decentralized sensor selection framework and its four variants are proposed to jointly optimize the sensing quality and energy efficiency.
2) A closed-loop adaptive mechanism is developed which runs on the edge nodes to predict the number of active SNs in the next measurement cycle in the respective regions.
3) The sensor selection and estimation processes rely on sparsification matrices constructed using previously estimated signals. To limit the accumulation of estimation errors over the measurement cycles, a retraining logic is designed to indicate the next retraining instants.
4) For monitoring a spatio-temporally dynamic process, a sparsification matrix is updated in each measurement cycle using principal component analysis (PCA) which utilizes w past instances of signal samples estimated via sparse Bayesian learning (SBL). The window size w is chosen by addressing a trade-off between retraining overhead and cluster head-to-cluster head (CH-to-CH) communication overhead.
5) Energy consumption in sensing, processing, and transmission is inherently integrated into the proposed framework. It is also integrated with the closest existing approaches [15], [18], and [21] for fair performance comparison.
6) Relative performance of the proposed framework with the closest approaches in [18], [21] demonstrates the achieved gain in energy efficiency (up to $\approx 84\%$) and network energy balance. Performance of the proposed framework is verified on real WSN data as well.

C. Organization

The WSN data acquisition model, sparse signal representation, and recovery are presented in Section II. Section III describes the proposed decentralized sensing problem. Energy cost and coverage regions are discussed in Section IV. Sections V, VI, and VII respectively presents the proposed decentralized sensing framework and its variants, simulation results, and concluding remarks. The main symbols and abbreviations used are listed in Table I.

II. System Model, Sparsification, and Recovery

A. WSN system model and its distributed counterpart

A homogeneous WSN is considered where energy-constrained SNs are deployed for monitoring a slowly-varying spatio-temporal process (e.g., air pollution shown in Fig. 1). Due to dense deployment, there exists spatial correlation among signals across the SNs. This can be used for energy-efficient sensing by activating a subset of SNs while the remaining ones can sleep.

TABLE I: Symbols and abbreviations

Notation	Description
$R, \mathcal{R}_r$, $N_r = \lvert\mathcal{R}_r\rvert$	Number of coverage regions, set containing SNs of region r, total number of SNs in region r
$\mathcal{A}_r^{(k)}$, $M_r^{(k)} = \lvert\mathcal{A}_r^{(k)}\rvert$	Set containing active SNs of region r in k^{th} cycle, number of active SNs of region r in k^{th} cycle
$\mathbf{z}_r^{(k)}, \widetilde{\mathbf{y}}_r^{(k)}$	Spatial signal $\left(\in \mathbb{R}^{N_r \times 1}\right)$ across SNs of region r in k^{th} cycle, signal measured by active SNs in set $\mathcal{A}_r^{(k)}$ $\left(\in \mathbb{R}^{M_r^{(k)} \times 1}\right)$
$\mathbf{A}_r^{(k)}, \mathbf{B}_r^{(k)}$	Sensing matrix $\left(\in \mathbb{R}^{M_r^{(k)} \times N_r}\right)$ and sparsification matrix $\left(\in \mathbb{R}^{N_r \times N_r}\right)$ of region r in k^{th} cycle
$E_c^{(k)}(n)$, E_s, $E_{rem}^{(k)}(n)$	Energy consumed by n^{th} SN in transmission and sensing (in J), remaining energy of n^{th} SN in k^{th} cycle (in J)
$E_{proc}^{(k)}(\cdot)$, $E_{tx}(d_{t-r})$	Energy consumed by SN in processing tasks and transmission of b bits of information at a distance d_{t-r} (in J)
E_{rx}, E_{elec}, η	Energy spent by SN to receive b information bits (J), consumed by radio electronics (J/bit), amplifier energy (J/bit/m^2)
$[\alpha, \beta]$	Lower and upper limits of BCRB
$M_r^{(k+1\lvert k)}$	Number of active SNs of region r predicted for $(k+1)^{th}$ cycle based on signal estimated during current cycle k
$\mathcal{A}_r^{k\lvert M_r^{(k)}-i}$	Pruned active set obtained by removing i SNs with the lowest signal amplitude $\lvert z_r^{(k)}(\cdot)\rvert$ from $\mathcal{A}_r^{(k)}$
$\widehat{\mathbf{x}}_r^{k\lvert M_r^{(k)}-i} \in \mathbb{R}^{N_r \times 1}$	Sparse signal estimate obtained using measurements from SNs $\in \mathcal{A}_r^{k\lvert M_r^{(k)}-i}$
$\text{BCRB}_r^{k\lvert M_r^{(k)}-i}, \widehat{\epsilon}_r^{(k)}$	BCRB corresponding to SNs $\in \mathcal{A}_r^{k\lvert M_r^{(k)}-i}$, euclidean distance between $\widehat{\mathbf{x}}_r^{(k)}$ and $\widehat{\mathbf{x}}_r^{k\lvert M_r^{(k)}-i}$
$\widehat{\delta}_r^{(k)}$	Heuristic giving an idea of process variability with respect to $(k-1)^{th}$ cycle
t_1, t_2	Time required to compute the matrix $\mathbf{B}_r^{(k)}$ using the PCA scheme and run update step of the adaptive mechanism
t_3, t_4	Time required in SBL-based recovery of $\mathbf{x}_r^{(k)}$ and prediction step of adaptive mechanism and to run retraining logic
BCRB_r, SS_r, CH_r, FC	Bayesian Cramér-Rao bound, sensor selection function, cluster head (or edge node) for region r, fusion center
$\text{Tr}\{\mathbf{A}\}$, $\lVert\mathbf{a}\rVert$, $(\cdot)^T$, $\text{diag}(\cdot)$	Trace of matrix $\mathbf{A}$, l_2-norm of vector $\mathbf{a}$, transpose of the vector/matrix, diagonalization operation on a vector

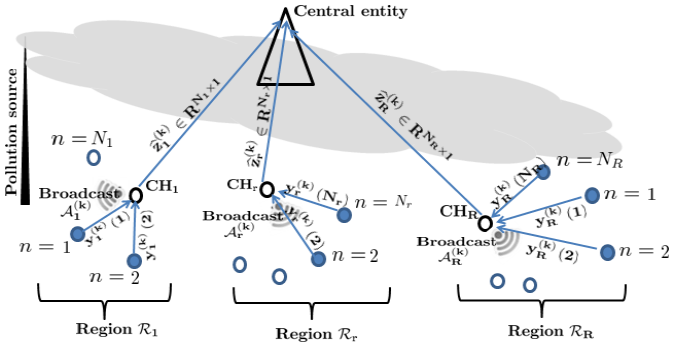


Fig. 1: Decentralized WSN system.

Let the signal corresponding to the random process observed across N SNs in the k^{th} measurement cycle be denoted by the spatial signal vector $\mathbf{z}^{(k)} = \left[z^{(k)}(1), \cdots, z^{(k)}(N)\right]^T \in \mathbb{R}^{N \times 1}$. Let $\mathcal{A}^{(k)} \subseteq \{1, \cdots, N\}$ represent the set of SNs activated for sensing in the k^{th} cycle (called active set); the total number of active SNs is given by $M^{(k)} = \left\lvert\mathcal{A}^{(k)}\right\rvert \leq N$. The active SNs are selected using a sensor selection problem (Section III-A) solved by the FC in a centralized setting. Let the activity status of the SNs be captured in a binary sensing matrix $\mathbf{A}^{(k)} \in \mathbb{R}^{M^{(k)} \times N}$. Each row of $\mathbf{A}^{(k)}$ corresponds to a distinct active SN. Hence, there will be $M^{(k)}$ unique rows in the matrix $\mathbf{A}^{(k)}$. If the m^{th} row represents the i^{th} active SN, then its entries are given by,

$$\mathbf{A}^{(k)}(m,n) = \begin{cases} 1, & n = i \quad \text{s.t. } i \in \mathcal{A}^{(k)} \\ 0, & n \in \{1, \cdots, N\} \setminus \{i\}. \end{cases} \tag{1}$$

Likewise, if another row (say l) represents a different active SN (say j), then $\mathbf{A}^{(k)}(l,j)=1$ and $\mathbf{A}^{(k)}(l,n)=0,\ (n\neq j)$. The measurement vector $\widetilde{\mathbf{y}}^{(k)}\in\mathbb{R}^{M^{(k)}\times 1}$ contains signals measured by the SNs $\in\mathcal{A}^{(k)}$. Similar to [15], [21], the system model in the centralized setting is given by

$$\widetilde{\mathbf{y}}^{(k)}=\mathbf{A}^{(k)}\mathbf{z}^{(k)}+\mathbf{n}^{(k)}, \tag{2}$$

where $\mathbf{n}^{(k)}\in\mathbb{R}^{M^{(k)}\times 1}$ is the additive white Gaussian noise vector with independent and identically distributed (IID) components of zero mean and variance σ^2. It comprises both measurement and channel noises, distributed as $\mathbf{n}_m^{(k)}\sim\mathcal{N}\left(\mathbf{0}_{M^{(k)}},\sigma_m^2\mathbf{I}_{M^{(k)}}\right)$ and $\mathbf{n}_{ch}^{(k)}\sim\mathcal{N}\left(\mathbf{0}_{M^{(k)}},\sigma_{ch}^2\mathbf{I}_{M^{(k)}}\right)$, respectively. These noises are independent of each other, as one is due to sensing/measurement impairment while the other is due to signal reception over the wireless channel. Thus, the overall noise $\mathbf{n}^{(k)}=\mathbf{n}_{ch}^{(k)}+\mathbf{n}_m^{(k)}$ with variance $\sigma^2=\sigma_{ch}^2+\sigma_m^2$, i.e., $\mathbf{n}^{(k)}\sim\mathcal{N}\left(\mathbf{0}_{M^{(k)}},\sigma^2\mathbf{I}_{M^{(k)}}\right)$. Note that the spatial correlation among the process signal across different SNs due to their dense deployment is captured by the spatial signal vector $\mathbf{z}^{(k)}$. It is not related to the noise. In general, the scenario where ambient noise corrupts the SNs' measurements may result in colored measurement noise, distributed as $\mathbf{n}_m\sim\mathcal{N}\left(\mathbf{0}_{M^{(k)}},\mathbf{Q}\right)$, where $\mathbf{Q}\in\mathbb{R}^{M^{(k)}\times M^{(k)}}$ is the colored covariance matrix. Then, $\mathbf{n}^{(k)}\sim\mathcal{N}\left(\mathbf{0}_{M^{(k)}},\mathbf{\Sigma}_n\right)$, where $\mathbf{\Sigma}_n=\left(\sigma_{ch}^2\mathbf{I}_{M^{(k)}}+\mathbf{Q}\right)\in\mathbb{R}^{M^{(k)}\times M^{(k)}}$. For simplicity, the case with white noise is considered throughout the paper.

In a centralized setting [15], the FC receives components of the vector $\widetilde{\mathbf{y}}^{(k)}$ from the currently active SNs, processes it to estimate the unknown spatial vector $\mathbf{z}^{(k)}$ (i.e., the signal across the entire WSN field), computes $\mathcal{A}^{(k+1)}$ (active set for next cycle), and broadcasts $\mathcal{A}^{(k+1)}$ to the SNs. In a decentralized setting, adaptive sensor selection (proposed in Sections III-B, III-C) and signal estimation are carried out at the node level. The WSN field is divided into R coverage regions/clusters, each containing N_r SNs given by the set $\mathcal{R}_r=\{1,2,\ldots,N_r\},\ \forall\ 1\leq r\leq R$, such that $N_r=|\mathcal{R}_r|$. Note that $\mathcal{R}_r,\forall r$, are the sets of non-overlapping SNs and $\sum_{r=1}^{R}N_r=N$. One SN from each region is elected as the cluster head (CH_r), called *edge node*, which receives the measurement signal vector $\widetilde{\mathbf{y}}_r^{(k)}\in\mathbb{R}^{M_r^{(k)}\times 1}$ from the set of SNs $\mathcal{A}_r^{(k)}$ that are activated from the respective region r. The total number of active SNs in region r is $M_r^{(k)}=\left|\mathcal{A}_r^{(k)}\right|\leq N_r$, and $\bigcup_{r=1}^{R}\mathcal{A}_r^{(k)}=\mathcal{A}^{(k)}$. The CH_r processes $\widetilde{\mathbf{y}}_r^{(k)}$ to estimate the spatial signal across all the SNs of that region denoted by $\mathbf{z}_r^{(k)}\in\mathbb{R}^{N_r\times 1}$, which is followed by computation of the region's active set $\mathcal{A}_r^{(k+1)}$ for the next measurement cycle. Subsequently, each CH_r sends the estimate of the signal vector $\mathbf{z}_r^{(k)}$ to the FC for archival purpose and broadcasts $\mathcal{A}_r^{(k+1)}$ in their respective regions. To formulate the distributed sensor selection problem, the system model in (2) is represented as in

(3), by placing the rows of the sensing matrix $\mathbf{A}^{(k)}$ corresponding to the active SNs of each region $\mathcal{R}_r$ one after the other and denoting the submatrix as $\check{\mathbf{A}}_r^{(k)} \in \mathbb{R}^{M_r^{(k)} \times N}$.

$$\begin{bmatrix} \widetilde{\mathbf{y}}_1^{(k)} \\ \vdots \\ \widetilde{\mathbf{y}}_R^{(k)} \end{bmatrix} = \begin{bmatrix} \check{\mathbf{A}}_1^{(k)} \\ \vdots \\ \check{\mathbf{A}}_R^{(k)} \end{bmatrix} \begin{bmatrix} \mathbf{z}_1^{(k)} \\ \vdots \\ \mathbf{z}_R^{(k)} \end{bmatrix} + \begin{bmatrix} \mathbf{n}_1^{(k)} \\ \vdots \\ \mathbf{n}_R^{(k)} \end{bmatrix}, \tag{3}$$

with $M^{(k)} = \sum_{r=1}^{R} M_r^{(k)}$ and $\mathbf{n}_r^{(k)} \in \mathbb{R}^{M_r^{(k)} \times 1}, \forall r$, denoting the noise across the SNs belonging to $\mathcal{A}_r^{(k)}$. The measurement vector of the r^{th} region $\widetilde{\mathbf{y}}_r^{(k)}$ has no contribution from the signal across the SNs of the other regions, i.e., $\mathbf{z}_j^{(k)},\ \forall\ 1 \leq j \leq R,\ j \neq r$. Thus, the sensing submatrix $\check{\mathbf{A}}_r^{(k)}$ can be truncated by considering the columns corresponding to the SNs $\in \mathcal{R}_r$ only. Denoting the truncated matrix as $\mathbf{A}_r^{(k)} \in \mathbb{R}^{M_r^{(k)} \times N_r}$, the system model for $\mathcal{R}_r,\ \forall r$, is given by

$$\widetilde{\mathbf{y}}_r^{(k)} = \mathbf{A}_r^{(k)} \mathbf{z}_r^{(k)} + \mathbf{n}_r^{(k)}. \tag{4}$$

Also, the correspondence between $\mathcal{A}_r^{(k)}$ and $\mathbf{A}_r^{(k)}$ is given by,

$$\mathbf{A}_r^{(k)}(m,n) = \begin{cases} 1, & n = i \ \text{ s.t. } i \in \mathcal{A}_r^{(k)} \\ 0, & n \in \{1, \cdots, N_r\} \setminus \{i\}. \end{cases} \tag{5}$$

The system model in (4) is used in the decentralized sensor selection problem proposed in Section III-B. A detailed discussion on finding the number of regions R and their edge nodes is provided in Section IV-B. These edge nodes are re-elected, as mentioned in the proposed variants of the decentralized framework (Section V). The considered homogeneous network setting enables any SN to serve as a candidate edge node for the region in which it is located.

B. PCA-SBL based signal sparsification and recovery

The unknown signals from all SNs (vectors $\mathbf{z}^{(k)}$ in (2) and $\mathbf{z}_r^{(k)}, \forall r$, in (4)) are required to be reconstructed using the measurements from a few active SNs (vectors $\widetilde{\mathbf{y}}^{(k)}$ in (2) and $\widetilde{\mathbf{y}}_r^{(k)}, \forall r$, in (4)) in ill-posed estimation scenarios. This is possible using sparse signal recovery frameworks, such as SBL and compressed sensing [4], that enable estimation of sparse signals using the under-sampled measurement vectors within certain accuracy level. However, this requires a priori information of the sparsification matrices that can make the observed signals $(\mathbf{z}^{(k)}, \mathbf{z}_r^{(k)})$ sparse.

Since there exists spatial correlation among the signals across the SNs, the vectors $\mathbf{z}^{(k)}$ and $\mathbf{z}_r^{(k)}, \forall r$, can be represented in sparse form using PCA-based sparsification matrices similar to [13], [15]. PCA-based sparsification is known to work well with real signals compared to other existing transformation matrices [22]. Thereafter, using the SBL framework, the sparse vectors

can be estimated respectively from the under-sampled measurement vectors $\mathbf{y}^{(k)}$ and $\mathbf{y}_r^{(k)}, \forall r$, of the sparse system model, that is discussed next.

PCA-based sparsification: For the system model (2), let $\mathbf{B}^{(k)} \in \mathbb{R}^{N\times N}$ be an orthogonal sparsification matrix obtained using PCA for representing the vector $\mathbf{z}^{(k)}$ as a sparse vector $\mathbf{x}^{(k)} \in \mathbb{R}^{N\times 1}$. Then, the system model (2) becomes

$$\mathbf{y}^{(k)} = \mathbf{\Theta}^{(k)}\mathbf{x}^{(k)} + \mathbf{n}^{(k)}, \tag{6}$$

where $\mathbf{\Theta}^{(k)} = \mathbf{A}^{(k)}\mathbf{B}^{(k)} \in \mathbb{R}^{M^{(k)}\times N}$ denotes the measurement matrix and $\mathbf{y}^{(k)} \in \mathbb{R}^{M^{(k)}\times 1}$ is the equivalent measurement vector. The details on PCA-based sparse representation are presented in Appendix A. Similarly, sparse representation of the regional system model (4) is given by (7), with $\mathbf{\Theta}_r^{(k)} = \mathbf{A}_r^{(k)}\mathbf{B}_r^{(k)} \in \mathbb{R}^{M_r^{(k)}\times N_r}$ and $\mathbf{B}_r^{(k)} \in \mathbb{R}^{N_r\times N_r}$ being regional measurement and sparsification matrices, respectively.

$$\mathbf{y}_r^{(k)} = \mathbf{\Theta}_r^{(k)}\mathbf{x}_r^{(k)} + \mathbf{n}_r^{(k)}, \ \forall r, \tag{7}$$

In the cycle immediately after training/retraining, the matrix $\mathbf{B}^{(k)}$ is computed using K_{tr} instances of the vector $\mathbf{z}^{(\cdot)}$. In other cycles, this matrix cannot be computed as presented in Appendix A due to the unknown vector $\mathbf{z}^{(k)}$. Exploiting the fact that the considered densely-deployed WSN monitors a slowly-varying process, the matrix $\mathbf{B}^{(k)}$ is anticipated to change slowly. This fact along with the availability of previous signal estimates allow online computation of $\mathbf{B}^{(k)}$ using w most recent instances of the estimated signal vector in the learning set, i.e., $\mathcal{T} = \{\widehat{\mathbf{z}}^{(k-w)}, \cdots, \widehat{\mathbf{z}}^{(k-2)}, \widehat{\mathbf{z}}^{(k-1)}\}$, such that $|\mathcal{T}| = w$. This adapts the sparsification matrix to the varying process better and avoids ensuing loss in sensing quality due to the usage of a stale estimate of $\mathbf{B}^{(k)}$ and hence $\mathbf{\Theta}^{(k)}$ in recovering the unknown vector $\mathbf{x}^{(k)}$ $\left(\text{or } \mathbf{z}^{(k)}\right)$. A similar argument goes for regional sparsification matrices $\mathbf{B}_r^{(k)}, \forall r$, where the region-wise sets $\mathcal{T}_r = \{\widehat{\mathbf{z}}_r^{(k-w)}, \cdots, \widehat{\mathbf{z}}_r^{(k-1)}\}, \forall r$, are used. The parameter w, referred to as the window size, is chosen by suitably addressing a trade-off observed in variations of retraining overheads and CH-to-CH transmission overheads, as detailed in Section VI-B. A larger value of w compared to K_{tr} is considered in simulations (Section VI) because error due to the use of previously estimated signals is absent in the latter case which uses the exact signal.

SBL-based recovery: A basic requirement of the adaptive sensing framework is a reliable estimate of the sensed signal to update the sparsification matrix and accurately provide feedback on the process dynamics for sensing and recovery purposes. SBL is known to give improved sparse signal estimate in ill-posed estimation scenarios [23]. It assigns a parameterized Gaussian

distribution to the unknown sparse vector and uses evidence-maximization criterion to estimate those hyperparameters. This work uses SBL to obtain estimate of the unknown vector $\mathbf{x}_r^{(k)}$ denoted by $\widehat{\mathbf{x}}_r^{(k)} \in \mathbb{R}^{N_r \times 1}$ from which the spatial signal vector estimate $\widehat{\mathbf{z}}_r^{(k)}$ is obtained. It employs a dictionary matrix $\mathbf{\Theta}_r^{(k)} = \mathbf{A}_r^{(k)} \mathbf{B}_r^{(k)}$ and an equivalent measurement vector $\mathbf{y}_r^{(k)}$ during each k^{th} cycle in the r^{th} region, $\forall 1 \leq r \leq R$, considering the regional system model (7). SBL-based signal recovery is explained in Appendix B. In presence of colored noise, CoNo-SBL scheme [24] is used for signal estimation.

The sparsification and recovery techniques presented here are used in the formulation of the decentralized sensing problem in the next section.

III. Proposed Decentralized Sensing Problem and Adaptive Solution

A. Sensor selection problem formulation

A practical sensing scheme needs to optimize a critical trade-off between sensing quality and energy efficiency. Sensing quality is measured by the mean-squared error (MSE) in the estimation of the unknown vector $\mathbf{x}^{(k)}$: MSE $= \mathrm{E}_{\mathbf{x}^{(k)}} \left\{ \left\| \mathbf{x}^{(k)} - \widehat{\mathbf{x}}^{(k)} \right\|^2 \right\}$. However, the MSE in the underlying ill-posed scenario (6) may not have a tractable closed-form expression [12]. Also, its dependence on the actual estimator $\widehat{\mathbf{x}}^{(k)}$ and $\mathbf{x}^{(k)}$ makes it practically unsuitable since the sensor selection task is followed by the estimation of $\mathbf{x}^{(k)}$. Therefore, similar to [15], [21], the current work exploits a theoretical lower bound, called Bayesian Cramér-Rao bound (BCRB), to measure sensing quality. It enables sensor selection before data acquisition and does not mandate availability of either $\mathbf{x}^{(k)}$ or $\mathbf{y}^{(k)}$ (measurement vector). The BCRB in the k^{th} cycle is expressed in (8); for the derivation and convexity proof, see Appendix C and [15] respectively.

$$\mathrm{BCRB}^{(k)} = \mathrm{Tr}\left\{ \left(\frac{1}{\sigma^2} \left(\mathbf{B}^{(k)} \right)^T \mathrm{diag}\left(\widetilde{\mathbf{a}}^{(k)} \right) \mathbf{B}^{(k)} + \left(\mathbf{\Gamma}^{(k)} \right)^{-1} \right)^{-1} \right\}, \tag{8}$$

where $\widetilde{\mathbf{a}}^{(k)} \in \mathbb{R}^{N \times 1}$ contains binary diagonal elements of the diagonal matrix $\widetilde{\mathbf{A}}^{(k)}$, i.e., $\mathrm{diag}\left(\widetilde{\mathbf{a}}^{(k)} \right) = \widetilde{\mathbf{A}}^{(k)} = \left(\mathbf{A}^{(k)} \right)^T \mathbf{A}^{(k)} \in \mathbb{R}^{N \times N}$. These diagonal elements have a one-to-one correspondence to the SNs; $\widetilde{a}^{(k)}(n) = 1$ implies an active state of the n^{th} SN, i.e., $\widetilde{a}^{(k)}(n) = \widetilde{\mathbf{A}}^{(k)}(n,n) = \begin{cases} 1, & n \in \mathcal{A}^{(k)} \\ 0, & n \notin \mathcal{A}^{(k)} \end{cases}$. The parameter $\mathbf{\Gamma}^{(k)} \in \mathbb{R}^{N \times N}$ represents a diagonal matrix with variance of the components of sparse vector $\mathbf{x}^{(k)}$ as its diagonal elements. These elements are set as the eigen values of the matrix $\widehat{\mathbf{\Sigma}}_{\mathcal{T}}$, computed using PCA in Appendix A, as done in [15]. For the energy efficiency of the WSN, consider a weighted sensor selection (SS) function:

$$\mathrm{SS}^{(k)} = \sum_{n=1}^{N} \frac{\left(E_c^{(k)}(n) + E_s \right)}{E_{rem}^{(k)}(n)} \widetilde{a}^{(k)}(n), \tag{9}$$

where $E_c^{(k)}(n), E_s$, and $E_{rem}^{(k)}(n)$ respectively denote the energy consumed by the n^{th} SN in signal transmission, sensing and its remaining energy in the k^{th} cycle detailed in Section IV-A.

The sensor selection problem needs to simultaneously minimize the BCRB (expression (8)) and SS function (expression (9)). Minimizing the BCRB minimizes the MSE [9], [12]. Minimizing the SS function gives more weight to the activation of the SNs with a low energy consumption $\left(E_c^{(k)} + E_s\right)$ and high remaining energy $\left(E_{rem}^{(k)}\right)$. This ensures efficient energy utilization and aids in energy load balancing among SNs. Basic constraints that are required to be handled by a practical sensing problem of a WSN are- (1) active/sleep state of the SNs $\left(\widetilde{a}^{(k)}(n) = 1/0, \forall n\right)$, (2) SNs with zero remaining energy $\left(E_{rem}^{(k)}(n) = 0\right)$ should not participate in sensing, and (3) coverage constraint (activates one SN from each designated regions) to enable better monitoring, ensure network energy balance, and prevent the creation of coverage holes. The binary constraint (1) makes the sensing problem non-convex; solving it using exhaustive search is infeasible for large number of nodes (N). Hence, similarly as in [8], [25], a convex sensing problem is realized by considering relaxed convex box constraint $\left(\widetilde{a}^{(k)}(n) \in [0,1], \forall n\right)$. Thus, the relaxed convex multi-objective sensor selection problem [15] is given by (10). Its decentralized formulation is proposed in the next section.

$$\underset{\widetilde{a}^{(k)}(n)\ \forall n}{\text{minimize}} \quad \left[\text{BCRB}^{(k)}, \text{SS}^{(k)}\right] \tag{10a}$$

$$\text{s.t.} \qquad \widetilde{a}^{(k)}(n) \in [0,1],\ n = 1, \cdots, N, \tag{10b}$$

$$\widetilde{a}^{(k)}(n) = 0, n \in \left\{i \,|\, E_{rem}^{(k)}(i) = 0, 1 \le i \le N\right\}, \tag{10c}$$

$$\sum_{j \in \mathcal{R}_r} \widetilde{a}^{(k)}(j) \ge 1,\ r = 1, \cdots, R. \tag{10d}$$

B. Decentralized sensor selection: Edge computing paradigm

In decentralized sensing, (10) is decoupled into R sub-problems, solvable at the edge nodes of the clusters. The BCRB (8) is decoupled as follows. Using the orthogonality property of the PCA-based matrix $\mathbf{B}^{(k)}$, i.e. $\left(\mathbf{B}^{(k)}\right)^T \mathbf{B}^{(k)} = \mathbf{B}^{(k)} \left(\mathbf{B}^{(k)}\right)^T = \mathbf{I}_N$, cyclic property of trace operator, i.e. $\text{Tr}\{\mathbf{PQR}\} = \text{Tr}\{\mathbf{RPQ}\}$ [26], and dropping the subscript (k) for brevity, expression (8) becomes $\text{BCRB} = \text{Tr}\left\{\left(\mathbf{B}\boldsymbol{\Gamma}^{-1}\mathbf{B}^T + \mathbf{A}^T \frac{1}{\sigma^2}\mathbf{I}_M \mathbf{A}\right)^{-1}\right\}$. Further, employing the matrix inversion lemma [26] and property $\text{Tr}\{\mathbf{C} + \mathbf{D}\} = \text{Tr}\{\mathbf{C}\} + \text{Tr}\{\mathbf{D}\}$ [26], the BCRB is given in (11). Denoting $\mathbf{P} = \mathbf{AB}$ and $\mathbf{Q} = \boldsymbol{\Gamma}\mathbf{B}^T\mathbf{A}^T$ in (11) and using identity $(\mathbf{I} + \mathbf{PQ})^{-1}\mathbf{P} = \mathbf{P}(\mathbf{I} + \mathbf{QP})^{-1}$ [26], the BCRB simplifies to (12). Using the notation $\mathbf{A}^T\mathbf{A} = \widetilde{\mathbf{A}}$ (mentioned along with (8)), (12) is rewritten as (13). If all the SNs are active, i.e. $\widetilde{\mathbf{A}} = \mathbf{I}_N$, the second term in (13) becomes $\text{Tr}\left\{\mathbf{B}^T\mathbf{B}\left(\sigma^2\mathbf{I}_N + \boldsymbol{\Gamma}\mathbf{B}^T\mathbf{B}\right)^{-1}\boldsymbol{\Gamma}^2\right\}$. Further, using orthogonality property of $\mathbf{B}$, it reduces to

$$\text{BCRB}^{(k)} = \text{Tr}\left\{\mathbf{B}\boldsymbol{\Gamma}\mathbf{B}^T\right\} - \text{Tr}\left\{\left(\sigma^2\mathbf{I}_M + \mathbf{A}\mathbf{B}\boldsymbol{\Gamma}\mathbf{B}^T\mathbf{A}^T\right)^{-1}\mathbf{A}\mathbf{B}\boldsymbol{\Gamma}^2\mathbf{B}^T\mathbf{A}^T\right\} \tag{11}$$

$$= \text{Tr}\left\{\mathbf{B}\boldsymbol{\Gamma}\mathbf{B}^T\right\} - \text{Tr}\left\{\mathbf{A}\mathbf{B}\left(\sigma^2\mathbf{I}_N + \boldsymbol{\Gamma}\mathbf{B}^T\mathbf{A}^T\mathbf{A}\mathbf{B}\right)^{-1}\boldsymbol{\Gamma}^2\mathbf{B}^T\mathbf{A}^T\right\} \tag{12}$$

$$= \text{Tr}\left\{\boldsymbol{\Gamma}\right\} - \text{Tr}\left\{\mathbf{B}^T\widetilde{\mathbf{A}}\mathbf{B}\left(\sigma^2\mathbf{I}_N + \boldsymbol{\Gamma}\mathbf{B}^T\widetilde{\mathbf{A}}\mathbf{B}\right)^{-1}\boldsymbol{\Gamma}^2\right\} \tag{13}$$

$$= \sum_{r=1}^{R}\left(\text{Tr}\left\{\boldsymbol{\Gamma}_r\right\} - \sum_{n\in\mathcal{R}_r}\frac{\boldsymbol{\Gamma}_r\left(n,n\right)^2}{\sigma^2 + \boldsymbol{\Gamma}_r\left(n,n\right)}\right)$$

$$= \sum_{r=1}^{R}\left(\text{Tr}\left\{\boldsymbol{\Gamma}_r\right\} - \text{Tr}\left\{\left(\sigma^2\mathbf{I}_{N_r} + \boldsymbol{\Gamma}_r\right)^{-1}\boldsymbol{\Gamma}_r^2\right\}\right) \tag{14}$$

$$= \sum_{r=1}^{R}\left(\text{Tr}\left\{\boldsymbol{\Gamma}_r\right\} - \text{Tr}\left\{\mathbf{B}_r^T\widetilde{\mathbf{A}}_r\mathbf{B}_r\left(\sigma^2\mathbf{I}_{N_r} + \boldsymbol{\Gamma}_r\mathbf{B}_r^T\widetilde{\mathbf{A}}_r\mathbf{B}_r\right)^{-1}\boldsymbol{\Gamma}_r^2\right\}\right). \tag{15}$$

$\text{Tr}\left\{\left(\sigma^2\mathbf{I}_N + \boldsymbol{\Gamma}\right)^{-1}\boldsymbol{\Gamma}^2\right\} = \sum_{n=1}^{N}\frac{\Gamma(n,n)^2}{\Gamma(n,n)+\sigma^2}$. Thus, the contribution of the n^{th} SN to the BCRB is $\left(-\frac{\Gamma(n,n)^2}{\Gamma(n,n)+\sigma^2}\right)$. Segregating the SNs' contributions region-wise, (13) can be rewritten as in (14) with the diagonal elements of $\boldsymbol{\Gamma}_r^{(k)} \in \mathbb{R}^{N_r\times N_r}$ equal to $\boldsymbol{\Gamma}^{(k)}\left(n,n\right),\ n\in\mathcal{R}_r$. Further, employing the orthogonality property of $\mathbf{B}_r$ and the assumption of all active SNs ($\widetilde{\mathbf{A}}_r = \mathbf{I}_{N_r}$), the BCRB expression (14) can be generalized as in (15). Following the intermediate steps (11) to (12) in reverse order and applying the dropped subscript (k) of the measurement cycle in (15), one obtains $\text{BCRB}^{(k)}$ as the sum of regional $\text{BCRB}_r^{(k)}$ functions:

$$\text{BCRB}^{(k)} = \sum_{r=1}^{R}\underbrace{\text{Tr}\left\{\left(\frac{1}{\sigma^2}\left(\mathbf{B}_r^{(k)}\right)^T\text{diag}\left(\widetilde{\mathbf{a}}_r^{(k)}\right)\mathbf{B}_r^{(k)} + \left(\boldsymbol{\Gamma}_r^{(k)}\right)^{-1}\right)^{-1}\right\}}_{\text{BCRB}_r^{(k)}}, \tag{16}$$

where $\widetilde{\mathbf{a}}_r^{(k)} \in \mathbb{R}^{N_r\times 1}$ contains diagonal elements of the diagonal matrix $\widetilde{\mathbf{A}}_r^{(k)}$ which have a one-to-one correspondence with SNs $\in \mathcal{R}_r$, i.e., $\text{diag}\left(\widetilde{\mathbf{a}}_r^{(k)}\right) = \widetilde{\mathbf{A}}_r^{(k)}$. Note that, (16) holds true when all the SNs are active; while for the other cases it is approximated as $\text{BCRB}^{(k)} < \sum_{r=1}^{R}\text{BCRB}_r^{(k)}$. Validity of this approximation is verified via simulations, in Section VI-D. Further, the SS function can be rewritten as a sum of regional $\text{SS}_r^{(k)}$ functions,

$$\text{SS}^{(k)} = \sum_{r=1}^{R}\underbrace{\sum_{n\in\mathcal{R}_r}\frac{\left(E_c^{(k)}\left(n\right) + E_s\right)}{E_{rem}^{(k)}\left(n\right)}\widetilde{a}_r^{(k)}\left(n\right)}_{\text{SS}_r^{(k)}}. \tag{17}$$

Employing (16), (17) in (10) and interchanging minimization and summation, the relaxed sensor selection problem (10) is decoupled into R sub-problems as:

$$\underbrace{\begin{aligned}\underset{\widetilde{a}_r^{(k)}(n),\ n\in\mathcal{R}_r}{\text{minimize}}\ & \left[\text{BCRB}_r^{(k)}, \text{SS}_r^{(k)}\right] \\ \text{s.t.}\quad & \widetilde{a}_r^{(k)}(n) \in [0,1],\ n\in\mathcal{R}_r, \\ & \widetilde{a}_r^{(k)}(n) = 0, n\in\left\{i \,|\, E_{rem}^{(k)}(i) = 0, i\in\mathcal{R}_r\right\}, \\ & \textstyle\sum_{n\in\mathcal{R}_r}\widetilde{a}_r^{(k)}(n) \geq 1\end{aligned}}_{R \text{ sub-problems}} \tag{18}$$

The number of active SNs $M_r^{(k)}$ and the active set $\mathcal{A}_r^{(k)}$ in $\mathcal{R}_r$ are obtained from the solution of (18) as:

$$M_r^{(k)} = \text{round}\left(\sum_{n=1}^{N_r} \widetilde{a}_r^{(k)}(n)\right), \tag{19}$$

$$\mathcal{A}_r^{(k)} = \text{Indices of } M_r^{(k)} \text{ largest elements of ordered set: } \left\{\widetilde{a}_r^{(k)}(n)\,|\widetilde{a}_r^{(k)}(n) \geq \widetilde{a}_r^{(k)}(m)\,, \forall n, m \in \{1,\ldots,N_r\}\right\} \tag{20}$$

The solution can also be obtained by first rounding $\widetilde{a}_r^{(k)}(n)$ to the nearest integer and then computing $M_r^{(k)}$ and $\mathcal{A}_r^{(k)}$, or using randomized rounding [9]. Note that the aim of current work is not to compare the decentralized and centralized sensing problems, but to propose and solve a decentralized sensing problem with components similar to the centralized one (Section III-A).

C. Proposed adaptive solution for decentralized sensing

The R multi-objective optimization problems for regional sensor selection (18) are solved at the respective edge nodes by using the scalarization method [27], with the difference that, here a mechanism is proposed to adapt the scalar weights $\lambda_r^{(k)}, \forall r$, in the k^{th} cycle. The scalarized sub-problem in (21) is solved using the CVX solver [28] due to its convex nature (similar to the sensor selection problem in [15]).

$$\underset{\widetilde{a}_r^{(k)}(n),\ n\in\mathcal{R}_r}{\text{minimize}} \ \left(1-\lambda_r^{(k)}\right)\text{BCRB}_r^{(k)} + \left(\lambda_r^{(k)}\right)\text{SS}_r^{(k)} \tag{21a}$$

$$\text{s.t.} \ \ \widetilde{a}_r^{(k)}(n) \in [0,1]\,,\ n \in \mathcal{R}_r, \tag{21b}$$

$$\widetilde{a}_r^{(k)}(n) = 0, n \in \left\{i \,|\, E_{rem}^{(k)}(i) = 0, i \in \mathcal{R}_r\right\}, \tag{21c}$$

$$\sum_{n\in\mathcal{R}_r} \widetilde{a}_r^{(k)}(n) \geq 1. \tag{21d}$$

The motivation behind the adaptation method is two-fold. First, it is intuitive that a higher number of active SNs is required to sense a process with a faster dynamics, so as to obtain the same sensing accuracy [29]. Second, it is observed that associating a fixed scalar weight λ_r gives more importance to the decreasing energy based function $\text{SS}_r^{(\cdot)}$ over the $\text{BCRB}_r^{(\cdot)}$ as the measurement cycle progresses. This may keep increasing the value of $\text{BCRB}_r^{(\cdot)}$ and the corresponding MSE.

A feedback-based adaptation mechanism (run by each CH_r) is proposed that predicts the number of active SNs $M_r^{(k+1|k)}$ for the $(k+1)^{th}$ cycle based on signal variations estimated using current estimates of $\widehat{\mathbf{x}}_r^{(k)}$ and $\widehat{\mathbf{z}}_r^{(k)}$. This value is further updated considering the SNs' updated energies (i.e., $E_{rem}^{(k+1)}(n), E_c^{(k+1)}(n) + E_s, \forall n \in \mathcal{R}_r$) while keeping the estimation error within a tolerable range. For this, it is ensured that the $\text{BCRB}_r^{(k)} \in [\alpha, \beta]$ during each cycle, with α and β

respectively being lower and upper BCRB limits required by the sensing application. The energy information is sent by the SNs to the respective CH_rs along with the measured signal $y_r^{(k)}(\cdot)$.

Prediction step: In k^{th} cycle, a heuristic $\widehat{\delta}_r^{(k)}$ (defined in (22)) is computed to estimate the process variability with respect to the $(k-1)^{th}$ cycle. If the value of $\widehat{\delta}_r^{(k)}$ is greater than a suitably chosen threshold δ_r^{th}, then the predicted value is increased as $M_r^{(k+1|k)} = \left(M_r^{(k)} + 1\right)$.

$$\widehat{\delta}_r^{(k)} = \frac{\left\| \widehat{\mathbf{z}}_r^{(k)} - \widehat{\mathbf{z}}_r^{(k-1)} \right\|}{\sqrt{N_r}}. \tag{22}$$

When $\widehat{\delta}_r^{(k)} \leq \delta_r^{th}$, the value of $M_r^{(k+1|k)}$ is decreased as follows. Consider the pruned active set $\mathcal{A}_r^{\left(k|M_r^{(k)}-i\right)}$ by removing i SNs (starting $i=1$) with the lowest signal amplitude $|\widehat{z}_r^{(k)}(\cdot)|$ from the set $\mathcal{A}_r^{(k)}$. Employ $\mathcal{A}_r^{\left(k|M_r^{(k)}-i\right)}$-based signal estimate $\widehat{\mathbf{x}}_r^{\left(k|M_r^{(k)}-i\right)}$ to compute another heuristic $\widehat{\epsilon}_r^{(k)}$ defined in (23). If $\widehat{\epsilon}_r^{(k)} \leq \epsilon_r^{th}$ (an appropriate threshold) and $\text{BCRB}_r^{(k|M_r^{(k)}-i)} \in [\alpha, \beta]$, set $M_r^{(k+1|k)} = \left(M_r^{(k)} - i\right)$; repeat the process with unit increment of i until $\text{BCRB}_r^{(k|M_r^{(k)}-i)} \notin [\alpha, \beta]$ or non-selection of any SN from the region i.e. $\left(M_r^{(k)} - i\right) < 1$. The reason is to explore the possibility of maximum achievable decrease in $M_r^{(k)}$ for the $(k+1)^{th}$ cycle such that the estimates $\widehat{\mathbf{x}}_r^{\left(k|M_r^{(k)}-i\right)}$ and $\widehat{\mathbf{x}}_r^{(k)}$ (current one) are similar and $\text{BCRB}_r^{(k|M_r^{(k)}-i)} \in [\alpha, \beta]$.

$$\widehat{\epsilon}_r^{(k)} = \left\| \widehat{\mathbf{x}}_r^{(k)} - \widehat{\mathbf{x}}_r^{\left(k|M_r^{(k)}-i\right)} \right\|. \tag{23}$$

Update step: The scalar $\lambda_r^{(k+1)}$ and active set for the next cycle $\mathcal{A}_r^{(k+1)}$ with $M_r^{(k+1|k)}$ active nodes, are computed by considering energy resources in the network, the updated $\mathbf{B}_r^{(k+1)}$, and employing modified binary search (MBS) (Alg. 1) algorithm [15]. The matrix $\mathbf{B}_r^{(k+1)}$ is updated using the updated learning set $\mathcal{T}_r = \left\{ \mathcal{T}_r \setminus \widehat{\mathbf{z}}_r^{(k-w+1)} \right\} \cup \left\{ \widehat{\mathbf{z}}_r^{(k)} \right\}$ as discussed in Section II-B. The MBS minimizes the sub-problem (21) on a search space $\left[\lambda_r^L, \lambda_r^U\right] \in [0,1]$ with $\lambda_r^U - \lambda_r^L < \Delta$ or $\sum_{n \in \mathcal{R}_r} \widetilde{a}_r^{(k+1)}(n) = M_r^{(k+1|k)}$ as the stopping criteria and Δ being a small positive number. The active set is extracted from the output $\widetilde{\mathbf{a}}_r^{(k+1)}$. If the corresponding $\text{BCRB}_r^{(k+1)}$ belongs to $[\alpha, \beta]$, then the SNs $\in \mathcal{A}_r^{(k+1)}$ are used for data acquisition. Otherwise $M_r^{(k+1|k)}$ is increased or decreased $\left(\text{i.e. } \min\left\{M_r^{(k+1|k)} + 1, N_r\right\} \text{ or } \max\left\{M_r^{(k+1|k)} - 1, 1\right\}\right)$ for the cases with $\text{BCRB}_r^{(k+1)} > \beta$ or $< \alpha$. The update step reiterates until $\text{BCRB}_r^{(k+1)} \in [\alpha, \beta]$ is achieved.

The adaptation mechanism stops when the energy of all SNs of a region is exhausted, i.e., $\sum_{n \in \mathcal{R}_r} E_{rem}^{(k+1)}(n) = 0$ or when it is not possible to satisfy the BCRB requirement with all alive SNs i.e. $\lambda_r^U - \lambda_r^L < \Delta$. At this juncture, it is worth noting that despite the unavailability of the true MSE, the adaptation mechanism ensures from the beginning that $\text{BCRB}_r^{(k+1)} \in [\alpha, \beta], \forall k$. Thus, the proposed approach is practically useful. A stepwise description of the proposed adaptive

Algorithm 1 Modified Binary Search (MBS)

Input: $E_s, E_c^{(k+1)}(n), E_{rem}^{(k+1)}(n), \forall n \in \mathcal{R}_r, \mathbf{B}_r^{(k+1)}, M_r^{(k+1|k)}$; **Initialization:** $\left[\lambda_r^L, \lambda_r^U\right] \leftarrow [0,1]$.
while $\lambda_r^U - \lambda_r^L \geq \Delta$ **do**
Set $\lambda_r^{(k+1)} = \left(\frac{\lambda_r^L + \lambda_r^U}{2}\right)$ and obtain $\widetilde{\mathbf{a}}_r^{(k+1)}$ by solving (21) using $\lambda_r^{(k+1)}, \mathbf{B}_r^{(k+1)}, E_s, E_c^{(k+1)}(n), E_{rem}^{(k+1)}(n), \forall\, n \in \mathcal{R}_r$.
if $\sum_{n\in\mathcal{R}_r} \widetilde{a}_r^{(k+1)}(n) > M_r^{(k+1|k)}$ **then** $\lambda_r^L \leftarrow \lambda_r^{(k+1)}$.
else if $\sum_{n\in\mathcal{R}_r} \widetilde{a}_r^{(k+1)}(n) < M_r^{(k+1|k)}$ **then** $\lambda_r^U \leftarrow \lambda_r^{(k+1)}$.
else $break$.
end if
end while
Output: $\widetilde{\mathbf{a}}_r^{(k+1)}, \lambda_r^{(k+1)}$.

mechanism that runs at each edge node is given under block 2 head of Algorithm 2. The energy update equations and the other blocks included in Algorithm 2 are explained in Section V.

Note that, the temporal correlation (or variations) of the process is used to adapt the number of active SNs ($M_r^{(k)}$) in a region in each cycle, while the spatial correlation is used in the MOP (18) to select these $M_r^{(k)}$ SNs out of total N_r SNs. This MoP ensures that the same SNs may not be selected in each cycle. Thus, this mechanism implicitly adjusts the inter-sample time of each SN (i.e., gap between cycles in which a SN is activated for sensing). The spatio-temporal modeling of synthetically generated process signal is discussed in simulation section VI.

IV. Energy Cost and Coverage Regions

A. Operational energy costs in WSN

The energy stored in a SN is used for transmission, reception, sensing, sleeping, and processing. Unlike the existing literature on distributed sensor selection, this work incorporates the energy information to improve energy efficiency.

Sensing cost: The energy consumed per sensing operation depends on the process to be sensed and the type of sensors that are used. The measurement time varies with the sensors, so does their sensing energy E_s [30]. It can be obtained from the data-sheet of the used sensor.

Processing cost: In the proposed framework, an edge node (i.e., CH in that cycle) performs computation for taking sensing decision of the SNs in the cluster. The processing energy consumption of the CH in the k^{th} cycle is computed as $E_{proc}^{(k)}\left(t_f^{(k)}\right) = V_f I_f t_f^{(k)}$ [31], where V_f is the supply voltage of the processor at operating frequency f, I_f is its average current, and $t_f^{(k)}$ is the computation time required in the k^{th} cycle for making sensing decision for the next cycle. $t_f^{(k)}$ is expressed in terms of number of clock cycles $N_{clk}^{(k)}$ needed to execute the task in k^{th} cycle as $t_f^{(k)} = \left(\frac{N_{clk}^{(k)}}{f}\right)$. Thus, $E_{proc}^{(k)}\left(t_f^{(k)}\right) = V_f I_f \left(\frac{N_{clk}^{(k)}}{f}\right)$. To approximately obtain $N_{clk}^{(k)}$ of the sensor board's processor, the task is run on a computer's processor (e.g., Intel i7-6700 CPU). The time $t_{proc}^{(k)}$ required to run current cycle's sub-problems on the computer is measured and $N_{clk}^{(k)}$ is then

set to $\frac{t_{proc}^{(k)}}{1/\widetilde{f}}$ where $\widetilde{f}$ is the processor's operating frequency. Thus, the energy consumed by the on-board processor is approximated as:

$$E_{proc}^{(k)}\left(t_{proc}^{(k)}\right) = V_f I_f \left(\frac{t_{proc}^{(k)}}{1/\tilde{f}}\right) \cdot \left(\frac{1}{f}\right). \tag{24}$$

Since a non-CH does not carry out any computation, its processing energy is $E_{proc}^{(k)}\left(t_{proc}^{(k)}\right) = 0$ J.

Radio energy cost: To account for the radio energy consumption, the model in [32] is used. The energy spent by a SN to transmit and receive a b bits long packet at a distance d_{t-r}, denoted respectively by $E_{tx}\left(d_{t-r}\right)$ and E_{rx}, are given by

$$E_{tx}\left(d_{t-r}\right) = bE_{elec} + b\eta\left(d_{t-r}\right)^{\nu} \text{ and } E_{rx} = bE_{elec}, \tag{25}$$

where E_{elec} (J/bit) is the energy consumed by the radio electronics, η (J/bit/m$^{\nu}$) characterizes the amplifier energy, and ν is the path loss exponent considered as 2. It is assumed that each SN knows its distance to the other SNs in its cluster. In the proposed framework, communication takes place between the SNs and their respective CHs, CHs and FC, and among CHs; thus, d_{t-r} and E_{tx} vary accordingly. The communication among CHs happens when a new CH is elected because the existing CH's remaining energy is on the verge of exhaustion. The old CH transfers control information to enable the new CH to execute the decentralized sensor selection task. This is explained in block 1 (control operations) of the proposed framework in Section V.

B. Number of coverage regions and their formation

In the proposed framework, the number of coverage regions R is determined during the initial stage (before the start of any sensing operation). It is calculated by considering the energy consumed in communication per measurement cycle, as noted in [32]. However, the energy consumed in processing and CH-to-CH communication are not considered because they are unknown at the initial stage. The frequency of occurrence of CH-to-CH communication depends on the remaining energies of the SNs and CHs left (based on the outcome of the decentralized sensor selection problem) and to-be-chosen new CHs. E_{proc} is tracked during the execution of the decentralized framework (block 2, 3 of Section V). Consider a SN deployment density in the i^{th} region of radius r_i as ρ_i with probability distribution f_ρ, the average number of SNs in the region is then given by $\overline{N}_i = \int_0^\infty \rho_i r_i^2 f_\rho\left(x\right) dx$. For simplicity, we consider the same radius for each region, i.e. $r_i = r,\ \forall i$. Accordingly, $\overline{N}_i =$ constant $= \frac{N}{R}$, where R is the number of equal sized coverage regions. Thus, with $\frac{N}{R}$ nodes per region on average, energy dissipated per cycle of a non-CH node (E_{non-CH}) and a CH node (E_{CH}) are given by,

$$E_{non-CH} = \underbrace{bE_{elec} + b\eta\left(d_{SN-CH}\right)^2}_{E_{tx}\text{: SN to CH comm.}} + \underbrace{bE_{elec}}_{E_{rx}\text{: CH to SN comm.}}, \tag{26}$$

$$E_{CH} = \underbrace{(N/R-1)\,bE_{elec}}_{E_{rx}\text{: SN to CH comm.}} + \underbrace{bE_{elec} + b\eta\left(d_{CH-SN}\right)^2}_{E_{tx}\text{: CH to SN comm.}} + \underbrace{bE_{elec} + b\eta\left(d_{CH-FC}\right)^2}_{E_{tx}\text{: CH to FC comm.}}. \quad (27)$$

It is assumed that every SN packs information in payload part of the b bits long packet, with zero padding if the information is less. Since the number of active SNs in a cycle is not known at the initial stage, consider all the SNs to be active in a cycle; the total communication energy consumption per cycle is $E_{total} \approx R\left[E_{CH} + (N/R-1)\,E_{non-CH}\right]$. The number of coverage regions R in a $F \times F$ m^2 field that minimizes E_{total} is obtained by setting its derivative with respect to R to zero [32]:

$$R \approx \sqrt{\frac{NF^2}{2\pi\left(\left(d_{CH-FC}\right)^2 - \frac{E_{elec}}{\eta}\right)}}. \quad (28)$$

Out of N SNs, R SNs with the highest remaining energy are chosen as edge nodes (designated as $CH_r,\ \forall\ r$) whenever coverage regions are formed/reformed. However, in simulations (Section VI), initially (when sensing starts, i.e. with $k = 0$) all the SNs are considered to have the same remaining (initial) energy. To avoid skewed selection of positions of the CHs in the network, κ-medoid [33] algorithm is used. It ensures a good distribution of the CHs over the WSN field. In the κ-medoid algorithm, spatial distance-based cost function is used. Since communication energy E_{tx} depends on the spatial distance, using κ-medoid energy load is evenly distributed among all the SNs. This avoids any overly-utilized SN that may run out of energy before the others. Each remaining SN chooses to connect with one CH_r that requires minimum communication energy.

V. Proposed Decentralized Sensing Framework

This section presents the proposed edge computing framework for decentralized sensor selection and its variants. The variants are proposed to get insights on the network performance provided by them which could help in deciding their suitability for different application scenarios. The computational complexity of the framework has also been discussed. In the schematic shown in Fig. 2, block 1 is customized according to the variant, while blocks 2 and 3 are generic for all the variants, and thus explained only once in Section V-A.

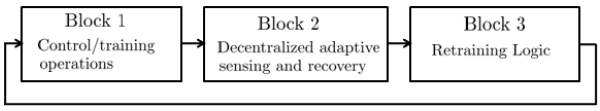


Fig. 2: Edge computing framework for decentralized sensing.

A. *Variant 1*

In this variant, the coverage regions remain fixed.

Block 1: This block carries out control/training operations in all the regions. When the remaining energy of the current edge node of a region is on the verge of depletion i.e., $E_{rem}^{(k)}\left(CH_r\right) \approx 0$,

another SN from the same region with the highest remaining energy is elected as the new CH_r. If the region does not require retraining, i.e., if retrain flag $= 0$, this new CH election process incurs CH-to-CH communication (hence energy) overhead due to transmission of w instances of recently estimated signal vectors $\widehat{\mathbf{z}}_r^{(\cdot)}$ from the previous CH to the new one so that the required matrix $\mathbf{B}_r^{(k)}$ can be computed at the new CH. The remaining energies are updated as follows:

$$E_{rem}^{(k)}\big(CH_{r(old)}\big)=E_{rem}^{(k-1)}\big(CH_{r(old)}\big)-wE_{tx}\big(\mathrm{d}_{CH_{r(old)}-CH_{r(new)}}\big), \tag{29a}$$

$$E_{rem}^{(k)}\big(CH_{r(new)}\big)=E_{rem}^{(k-1)}\left(CH_{r(new)}\right)-wE_{rx}. \tag{29b}$$

If the region requires retraining as well (retrain flag $= 1$), the energy is consumed only due to retraining (using K_{tr} instances of the signal $\mathbf{z}_r^{(\cdot)}$) and not CH-to-CH communication. Thus, the remaining energy is computed as:

$$E_{rem}^{(k)}\left(n\right)=E_{rem}^{(k-1)}\left(n\right)-K_{tr}E_s,\ 1\leq k\leq K,\ n\in\mathcal{R}_r. \tag{30}$$

For the case $E_{rem}^{(k)}\left(CH_r\right)>0$, the current CH_r continues to remain the edge node. Further, if retraining is not needed, the region's operations continue without any overheads. Otherwise, the region is retrained without changing CH and the corresponding energy update is given by (30).

Block 2: This block comprises the proposed adaptive solution for decentralized sensing and SBL-based recovery (Section III-C) that is run in parallel by the CHs in all regions. The SNs $\in \mathcal{R}_r$ are informed by CH_r about their active/sleep status by running the sub-problem (21). Thereafter, the measured data $\mathbf{y}_r^{(k)}$ is sent by the active SNs to the respective CH_r. The remaining energies of CH_r and the active/sleeping SNs are updated as,

$$E_{rem}^{(k)}(CH_r) = E_{rem}^{(k-1)}(CH_r)-E_{rx}M_r^{(k)}-E_{tx}\big(\bar{d}_{CH_r-SN}\big)-E_{proc}^{(k)}\left(t_1+t_2\right), \tag{31a}$$

$$E_{rem}^{(k)}(n) = E_{rem}^{(k-1)}(n)-E_{rx}-E_{tx}(d_{n-CH_r})-E_s \quad [\text{for active SNs, } n\in\mathcal{A}_r^{(k)}], \tag{31b}$$

$$E_{rem}^{(k)}(m) = E_{rem}^{(k-1)}\left(m\right)-E_{rx}-E_{sl} \quad [\text{for sleeping SNs, } m\in\mathcal{R}_r-\mathcal{A}_r^{(k)}], \tag{31c}$$

where t_1 and t_2 are respectively the time required to compute the matrix $\mathbf{B}_r^{(k)}$ and to run the update step of the adaptive solution (Section III-C) for decentralized sensor selection, $\bar{d}_{CH_r-SN}$ is the average distance of the CH_r to any SN $\in \mathcal{R}_r$, and E_{sl} denotes the energy needed by a SN to sustain network operations without collecting data in sleep mode. The vector $\widehat{\mathbf{x}}_r^{(k)}$ is estimated at CH_r using the SBL approach (Section II-B) and then the estimated spatial signal vector $\widehat{\mathbf{z}}_r^{(k)}$ is transmitted to the FC. Further, the value $M_r^{(k+1|k)}$ and hence $\lambda_r^{(k+1)}$ are predicted using the mechanism described in Section III-C. The remaining energy of CH_r is then updated as in (32) with t_3 being the processing time required in the SBL-based recovery of $\mathbf{x}_r^{(k)}$ and prediction step of the adaptive mechanism.

$$E_{rem}^{(k)}(CH_r)=E_{rem}^{(k-1)}(CH_r)-E_{tx}(d_{CH_r-FC})-E_{proc}^{(k)}(t_3)\,. \tag{32}$$

Block 3: This block runs the retraining logic at CH_r at the end of every measurement cycle. The remaining energy is updated as in (33) with t_4 being processing time to run the logic.

Algorithm 2 Adaptive decentralized sensing framework

Input: $\mathcal{T}_r = \left\{\mathbf{z}_r^{(-K_{tr}+1)}, \ldots, \mathbf{z}_r^{(-1)}, \mathbf{z}_r^{(0)}\right\}, \mathcal{R}_r, \alpha, \beta, \delta_r^{th}, \epsilon_r^{th}$.
Initialization: $k=1, E_c^{(1)}(\cdot), E_{rem}^{(0)}(\cdot), \lambda_r^{(1)}, exitflag=0$.
-Compute $\bar{\mathbf{z}}_{\mathcal{T}_r}, \widehat{\mathbf{\Sigma}}_{\mathcal{T}_r}$, & $\mathbf{B}_r^{(0)}$ using PCA (Section II-B); set $\mathbf{B}_r^{(k)} \approx \mathbf{B}_r^{(k-1)}$; calculate t_1.
-Solve (21) and construct $M_r^{(1)}, \mathcal{A}_r^{(1)}$ using (19), (20).
Obtain $\mathbf{A}_r^{(1)}$ using (5); calculate t_2.
while $exitflag == 0$ **do**
  Block 1: Execute operations; update energy (29,30).
  Block 2: Broadcast $\mathcal{A}_r^{(k)}$, collect $\mathbf{y}_r^{(k)}, E_{rem}^{(k+1)}(\cdot)$ from active SNs.
  Update remaining energy using (31).
  Start clock to measure time t_3. Set $\mathbf{\Theta}_r^{(k)} = \mathbf{A}_r^{(k)}\mathbf{B}_r^{(k)}$.
  Compute $\widehat{\mathbf{x}}_r^{(k)}$ using SBL scheme (Section II-B).
  Set $\widehat{\mathbf{z}}_r^{(k)} = \bar{\mathbf{z}}_{\mathcal{T}_r} + \left(\mathbf{B}_r^{(k)}\right)^T \widehat{\mathbf{x}}_r^{(k)}$ & transmit to the FC.
  if $\sum_{n\in\mathcal{R}_r} E_{rem}^{(k+1)}(n) > 0$ **then** **(Prediction step)**
    Initialize $M_r^{(k+1|k)} \leftarrow M_r^{(k)}$; Calc. $\widehat{\delta}_r^{(k)}$ using (22).
    if $\widehat{\delta}_r^{(k)} > \delta_r^{th}$ **then** $M_r^{(k+1|k)} \leftarrow M_r^{(k)} + 1$.
    else
      for $i = 1, \ldots, M_r^{(k)} - 1$ **do**
        Construct $\mathcal{A}_r^{(k|M_r^{(k)}-i)}$, $\mathbf{A}_r^{(k|M_r^{(k)}-i)}$ (eq. 5).
        Set $\mathbf{\Theta}_r^{(k|M_r^{(k)}-i)} = \mathbf{A}_r^{(k|M_r^{(k)}-i)}\mathbf{B}_r^{(k)}$.
        Obtain $\widehat{\mathbf{x}}_r^{(k|M_r^{(k)}-i)}$ using SBL (Section II-B).
        Calc. $\text{BCRB}_r^{(k|M_r^{(k)}-i)}, \widehat{\epsilon}_r^{(k)}$ using (16, 23).
        if $\widehat{\epsilon}_r^{(k)} \leq \epsilon_r^{th}$ and $\text{BCRB}_r^{(k|M_r^{(k)}-i)} \in [\alpha, \beta]$ **then**
          $M_r^{(k+1|k)} \leftarrow M_r^{(k)} - i$.
        else if $\text{BCRB}_r^{(k|M_r^{(k)}-i)} \notin [\alpha, \beta]$ **then** $break$.
        end if
      end for
    end if
  end if
  Stop clock & calc. t_3; update energy using (32).
  $\mathcal{T}_r = \left\{\mathcal{T}_r \setminus \widehat{\mathbf{z}}_r^{(k-w+1)}\right\} \bigcup \left\{\widehat{\mathbf{z}}_r^{(k)}\right\}$ **(Update step)**.
  Compute $\bar{\mathbf{z}}_{\mathcal{T}_r}, \widehat{\mathbf{\Sigma}}_{\mathcal{T}_r}, \mathbf{B}_r^{(k+1)}$ (Section II-B); calc. t_1.
  do
    Start clock to measure t_2. Call Alg. 1- MBS.
    if $\sum_{n\in\mathbf{R}_r} \widetilde{\mathbf{A}}_r^{(k+1)}(n,n) == M_r^{(k+1|k)}$ **then**
      Obtain $\mathcal{A}_r^{(k+1)}, \mathbf{A}_r^{(k+1)}$ using (20), (5).
      Calc. $\text{BCRB}_r^{(k+1)}$ using $\mathbf{A}_r^{(k+1)}, \mathbf{B}_r^{(k+1)}$ in (16).
      if $\text{BCRB}_r^{(k+1)} < \alpha$ **then**
        $M_r^{(k+1|k)} \leftarrow \max\left\{M_r^{(k+1|k)} - 1, 1\right\}$.
      else if $\text{BCRB}_r^{(k+1)} > \beta$ **then**
        $M_r^{(k+1|k)} \leftarrow \min\left\{M_r^{(k+1|k)} + 1, N_r\right\}$.
      end if
    else $exitflag \leftarrow 1$; $break$.
    end if
  while $\text{BCRB}_r^{(k+1)} \notin [\alpha, \beta]$.
  Set $M_r^{(k+1)} \leftarrow M_r^{(k+1|k)}$. Stop clock; calc. t_2.
else $exitflag \leftarrow 1$.
end if
Block 3: Run retraining logic; calc. t_4.
Update remaining energy using (33). Set $k \leftarrow k+1$.
end while

$$E_{rem}^{(k)}(CH_r) = E_{rem}^{(k-1)}(CH_r) - E_{proc}^{(k)}(t_4). \tag{33}$$

Proposed retraining logic: As the measurement cycles progress, the recovery process of $\widehat{\mathbf{x}}_r^{(k)}$ can become increasingly erroneous due to the usage of overcomplete dictionary $\mathbf{\Theta}_r^{(k)}$ which is based on $\mathbf{B}_r^{(k)}$. Since a few past instances of the estimated signal $\widehat{\mathbf{z}}_r^{(\cdot)}$ are used to estimate the matrix $\mathbf{B}_r^{(k)}$, the estimation error accumulates over cycles. This in turn affects sensor selection and signal recovery accuracy, producing non-sparse estimate $\widehat{\mathbf{x}}_r^{(k)}$ of otherwise-unknown sparse signal $\mathbf{x}_r^{(k)}$. It calls for an error detection logic and retraining in the region r.

As the true signal vector $\mathbf{x}_r^{(k)}$ is unknown, the retraining logic is proposed based on the current reconstructed signal vector $\widehat{\mathbf{x}}_r^{(k)}$ itself. The idea is to approximately find the number of non-sparse components $\widehat{NC}_r^{(k)}$ in the reconstructed signal $\widehat{\mathbf{x}}_r^{(k)}$ and compare it against a suitable threshold value NC_r^{th}. $\widehat{NC}_r^{(k)} > NC_r^{th}$ hints the need of retraining the region r. However, keeping in mind the uncertainty that the logic is based on the reconstructed signal $\widehat{\mathbf{x}}_r^{(k)}$, which usually has some error, the threshold violation is checked for a few consecutive measurement cycles (3 cycles

considered for simulation results in Section VI). If for all cycles $\widehat{NC}_r^{(\cdot)} > NC_r^{th}$, the network is retrained. Note that a component $\widehat{x}_r^{(k)}(\cdot)$ is considered as non-sparse if it constitutes more than 0.1% of total energy of signal vector $\widehat{\mathbf{x}}_r^{(k)}$, i.e., $\frac{\left(\widehat{x}_r^{(k)}(\cdot)\right)^2}{\sum_{n\in\mathcal{R}_r}\left(\widehat{x}_r^{(k)}(n)\right)^2} > \frac{0.1}{100}$. The 0.1% energy criteria ensures that no non-sparse component of $\widehat{\mathbf{x}}_r^{(k)}$ is wrongly considered as sparse. Also, the knowledge of NC_r^{th} can be roughly obtained from instances of the training signal. The logic runs in parallel at all the R edge nodes during each cycle.

B. Variant 2 (Block 1)

This variant keeps the coverage regions fixed while changes the CH_rs in every measurement cycle. The SN in a region with the highest remaining energy is elected as the new CH. If the region does not require retraining (retrain flag $= 0$), the remaining energies left after the transmission of overheads are given by (29). Else (retrain flag $= 1$), the region requires retraining; the energy is consumed in retraining overheads, and the remaining energy is given by (30).

C. Variant 3 (Block 1)

In this variant, the coverage regions and their CHs are re-formed (using the procedure outlined in Section IV-B) when the remaining energy of any CH is exhausted, i.e. $E_{rem}^{(k)}(CH_r) \approx 0,\ 1 \leq r \leq R$, and the entire network (all regions) is retrained (irrespective of retrain flag status). Otherwise, the regions and their CHs remain fixed and region-wise retraining is carried out only when respective retrain flag is set 1. In both cases, the updated remaining energy of SNs is given by (30). When the regions are re-formed, their NC_r^{th}, δ_r^{th} are changed and $M_r^{(k+1|k)}$ are also updated such that $\text{BCRB}_r \in [\alpha, \beta], \forall r$, for the $(k+1)^{th}$ cycle.

D. Variant 4 (Block 1)

This variant re-forms the coverage regions and the CHs whenever any region requires retraining (i.e., retrain flag $= 1$). The updated remaining energy is given by (30). Like variant 3, $M_r^{(k+1|k)}$, NC_r^{th}, and δ_r^{th} are changed for all the R regions. However, if retrain flag $= 0$ for all the regions and $E_{rem}^{(k)}(CH_r) \approx 0$ for any region r, then a SN from that region with the highest remaining energy is elected as the new CH and the updated remaining energies are given by (29). Otherwise, the regions continue their operations when $E_{rem}^{(k)}(CH_r) > 0, \forall r$.

E. Complexity analysis

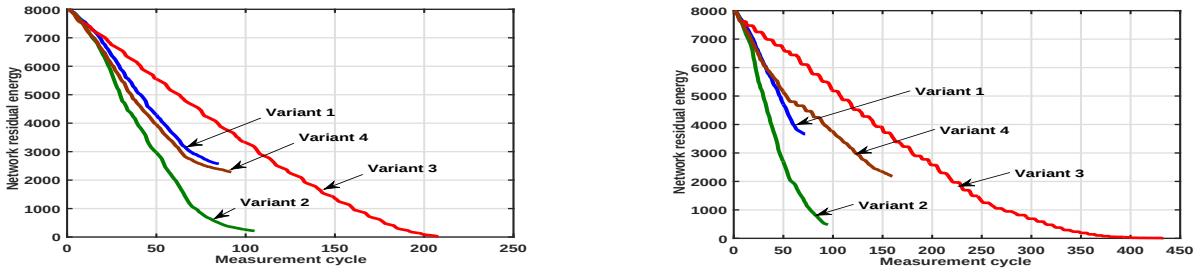
The CHs of R regions run the proposed framework and the complexity is computed as noted in [15]. Let the complexity of the recovery scheme executed by the CHs be denoted by $\mathcal{O}_{Rec}$.

Complexity of block 1 and 3 are respectively $\approx \mathcal{O}(1)$ and $\mathcal{O}(N_r)$. The overall complexity of the prediction step of block 2 run by each CH_r is $\approx \left(\mathcal{O}(N_r) + \mathcal{O}\left((M_r^{(k)})^3\right) + \mathcal{O}_{Rec}\right)$. The complexity of update step in block 2 is given by $\approx \left(\mathcal{O}\left(\left(M_r^{(k+1|k)}\right)^3\right) + \left(M_r^{(k+1|k)}\right)\log_2\left(\frac{1}{\Delta}\right)\mathcal{O}\left(\widetilde{N}_r \log\left(\frac{1}{\nu}\right)\right)\right)$, where $\widetilde{N}_r$ represents the number of variables after converting problem (21) to a standard form by the CVX and ν is the precision accuracy (10^{-8} by default). Note that $\widetilde{N}_r$ is a function of N_r. The overall complexity of the proposed framework is $\approx \left(\mathcal{O}\left(max\left(N_r, \left(M_r^{(k)}\right)^3\right)\right) + \mathcal{O}_{Rec} + \mathcal{O}\left((M_r^{(k+1|k)})^3\right) + \left(M_r^{(k+1|k)}\right)\log_2\left(\frac{1}{\Delta}\right)\mathcal{O}\left(\widetilde{N}_r \log\left(\frac{1}{\nu}\right)\right)\right)$.

VI. Simulation Results

Energy efficiency of the proposed adaptive decentralized sensing framework is studied over the competitive decentralized approaches in [21], [18] and the centralized one in [15]. The results show comparable sensing accuracy achieved due to adaptive sensor selection and retraining logic.

Consider $N = 80$ sensors equipped with non-rechargeable batteries and deployed in a 2-D WSN field of size $F \times F = 100 \times 100$ m^2, which is divided into $\lceil\sqrt{N}\rceil \times \lceil\sqrt{N}\rceil = 9 \times 9$ square areas of size $\approx 11 \times 11$ m^2 each roughly containing one randomly deployed SN [34]. The FC is located at a distance 110 m above the center of the field. Consider a total $S = 1$ pollution source located at the origin of the field. The pollution signal at the source $s, 1 \leq s \leq S$, during $k = 1$ cycle ($f^{(k)}(s)$) is generated as an independent real Gaussian [23], [35] with zero mean and variance $\sigma_f^2(s) = 1$. Modeling the temporal correlation using an AR(1) process [15], the temporal samples for ($k > 1$) are generated as $f^{(k)}(s) = \phi f^{(k-1)}(s) + \sqrt{1-\phi^2} w^{(k)}(s)$, where $w^{(k)}(s) \sim \mathcal{N}\left(0, \sigma_f^2(s)\right)$ and ϕ is temporal correlation coefficient between two consecutive samples. Further, the spatial correlation is modeled using power exponential model. During the k^{th} cycle, the pollution signal across SN n is obtained as $z^{(k)}(n) = \sum_{s=1}^{S} e^{\frac{-d_{n\text{-}s}}{\theta}} f^{(k)}(s)$ [34], where $d_{n\text{-}s}$ is the distance between SN n and the source s and θ is the spatial correlation parameter of the source. Spatio-temporal dynamics of the pollution process are set as $(\theta, \phi) = (100, 0.99)$. To guarantee a reasonable sensing performance in severely ill-posed estimation scenarios $\left(N_r/M_r^{(k)} \gg 2\right)$, the noise variance is fixed as $\sigma^2 = 10^{-5}$. $K_{tr} = 5$ cycles are used for training/retraining purpose. The existing works [12], [21] generate the sparse signal $\mathbf{x}^{(k)}$ with known sparsity and Gaussian distributed matrix $\mathbf{B}^{(k)}$ separately. In contrast, herein synthetic data, that considers the effect of process dynamics, is used to generate the matrix $\mathbf{B}_r^{(k)}$ and obtain the sparse signal $\mathbf{x}_r^{(k)}$ without assuming any prior knowledge of sparsity. Thus, the sensing framework evaluated on



(a) $E_s = 0.48$ J (b) $E_s = 0.0048$ J

Fig. 3: Performance of network residual energy of the proposed variants of decentralized sensing framework.

this synthetic data is readily applicable to real spatio-temporal process, as illustrated in simulation Section VI-G. The diagonal elements of $\mathbf{\Gamma}_r^{(k)}$ are approximately set as the eigenvalues of $\mathbf{\Sigma}_{\mathcal{T}_r}^{(k)}$ [15]. The sparsification matrix $\mathbf{B}_r^{(k)}$ is set as eigen vectors of $\mathbf{\Sigma}_{\mathcal{T}_r}^{(k)}$. The initial energy of SNs is set to 100 J $\left(\text{i.e. } E_{rem}^{(0)}(\cdot) = 100\right)$ and the other energy parameters are: $E_{elec} = 50$ nJ/bit, $\eta = 10^{-10}$ J/bit/m^2 [32], $b = 50, \nu = 2$, $V_f = 0.8$ V, $I_f = 0.05$ A, $f = 60$ MHz [36], and $\tilde{f} = 3.4$ GHz as mentioned in Section IV-A. Using (28), the value of R is obtained as 4. Initial value of the hyperparameter estimates and convergence accuracy of the SBL scheme are respectively set as $\left(\gamma_r^{(k)}(n)\right)^{(0)} = 1, \forall n, \forall k$ and $\left\| \left(\boldsymbol{\gamma}_r^{(k)}\right)^{(l+1)} - \left(\boldsymbol{\gamma}_r^{(k)}\right)^{(l)} \right\| \leq 10^{-7}$. Performance measures are, sensing error $(\text{MSE}) = \left\|\mathbf{x}^{(k)} - \widehat{\mathbf{x}}^{(k)}\right\|^2 = \left\|\mathbf{z}^{(k)} - \widehat{\mathbf{z}}^{(k)}\right\|^2$ and network residual energy $= \sum_{n=1}^{N} E_{rem}^{(k)}(n)$. SBL-based signal estimation is averaged over 300 Monte-Carlo iterations in each measurement cycle. The objective functions $\text{BCRB}_r^{(k)}$ and $\text{SS}_r^{(k)}$ of (21) are normalized using their respective smallest values as, $\frac{\text{Tr}\left\{\left(\frac{1}{\sigma^2}\left(\mathbf{B}_r^{(k)}\right)^T \text{diag}\left(\tilde{\mathbf{a}}_r^{(k)}\right)\mathbf{B}_r^{(k)} + \left(\mathbf{\Gamma}_r^{(k)}\right)^{-1}\right)^{-1}\right\}}{\text{Tr}\left\{\left(\frac{1}{\sigma^2}\left(\mathbf{B}_r^{(k)}\right)^T \mathbf{I}_{N_r}\mathbf{B}_r^{(k)} + \left(\mathbf{\Gamma}_r^{(k)}\right)^{-1}\right)^{-1}\right\}}$ and $\frac{\left\{\sum_{n\in\mathcal{R}_r} \frac{\left(E_c^{(k)}(n)+E_s\right)}{E_{rem}^{(k)}(n)} \tilde{a}_r^{(k)}(n)\right\}}{\min\left\{\frac{\left(E_c^{(k)}(n)+E_s\right)}{E_{rem}^{(k)}(n)} \tilde{a}_r^{(k)}(n), n\in\mathcal{R}_r\right\}}$. CVX [28] is used to solve the decoupled sensing problem (21). The parameter Δ in the MBS is set as 10^{-4} and the BCRB window is $[\alpha, \beta] = [10^{-5}, 10^{-4}]$. To effectively capture the process dynamics, the threshold ϵ_r^{th} is set as 10^{-5} and δ_r^{th} is the average of threshold value of signal variation observed at SNs of that region, i.e. $\frac{1}{N_r}\sum_{n\in\mathcal{R}_r}(e^{\frac{-d_{n\text{-}s}}{\theta}} \times 0.11)$, where 0.11 is the threshold value of signal variation. The default window size for $\mathbf{B}_r^{(k)}$ estimate is set as $w = 6$ as explained in Section VI-B. Though the competitive approaches do not account for SN's energy, for comparison energy consumption of the SNs in these approaches are considered similar to the proposed one.

A. Performance of the proposed CH selection variants

Network residual energy of all the variants versus measurement cycles are plotted for two different values of the sensing energy E_s in Figs. 3(a)-(b). Variant 3 offer the highest network lifetime. A higher network lifetime of variant 3 compared to variant 1 is attributed to the fact

that it re-forms coverage regions and elects new CHs when the energy of the CH of any region depletes. This ensures that the SNs always communicate with the CHs that require the least energy, which is not the case with variant 1. Lesser network lifetime of the variant 2 is due to the CH-to-CH transmission overheads incurred in each cycle.

TABLE II: Network lifetime (in cycles) and normalized average node lifetime of the proposed variants

	Variant	1	2	3	4
$E_s = 0.48$ J	Network lifetime	85	105	**213**	91
	Node lifetime	0.7065	**0.8725**	0.6806	0.6506
$E_s = 0.0048$ J	Network lifetime	71	95	**438**	159
	Node lifetime	0.7375	**0.9144**	0.5739	0.6164

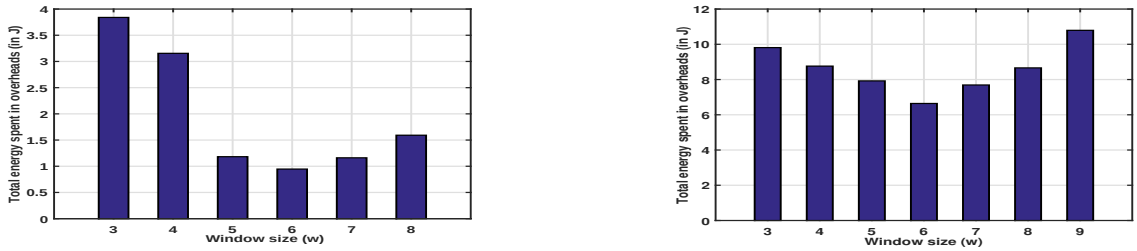
Table II shows the network lifetime (total number of measurement cycles executed (K)) and normalized average node lifetime (normalized by K) in different CH selection variants. Note that, the terms normalized average node lifetime and node lifetime are used interchangeably in this work. It can be observed that, despite being less energy-efficient than the other variants, variant 2 has a higher node lifetime which can help in preventing network coverage outage.

Remark 1. *Variant 3 offers a higher network lifetime, variant 2 offers a higher node lifetime.*

B. Choice of window size w

Using a higher number of instances w of the estimated signal $\widehat{\mathbf{z}}_r^{(\cdot)}$ increases the accuracy of $\mathbf{B}_r^{(k)}$ computation. It decreases the overhead of frequent retraining, but increases the communication overhead, as w instances of $\widehat{\mathbf{z}}_r^{(\cdot)}$ are transmitted from a CH to the chosen new CH. Thus, the chosen w needs to balance between the retraining and CH-to-CH communication overheads. Retraining overhead is calculated as the sum of sensing energy of all the SNs in all retraining phases; CH-to-CH communication overhead is the sum of transmission and reception energy consumed in sending w instances of $\widehat{\mathbf{z}}_r^{(\cdot)}$ from one CH to another. From Figs. 4(a)-(b) it is observed that, with the considered network settings the total energy spent in the overheads is minimum when $w = 6$ in both the variants 1 and 2. A similar behavior is seen for variant 4 as well. In variant 3, there is only retraining overhead. However, for consistency in performance comparison among all the variants in Section VI-A, w is set as 6.

Since the sensor selection and process estimation tasks are performed region-wise to monitor the same spatio-temporal process, the parameter w is kept the same for all regions to maintain the same computation accuracy of $\mathbf{B}_r^{(\cdot)}, \forall r$, across the regions. Different choices of w would be desirable for multi-sensing scenario [37] where each region monitors a different process.



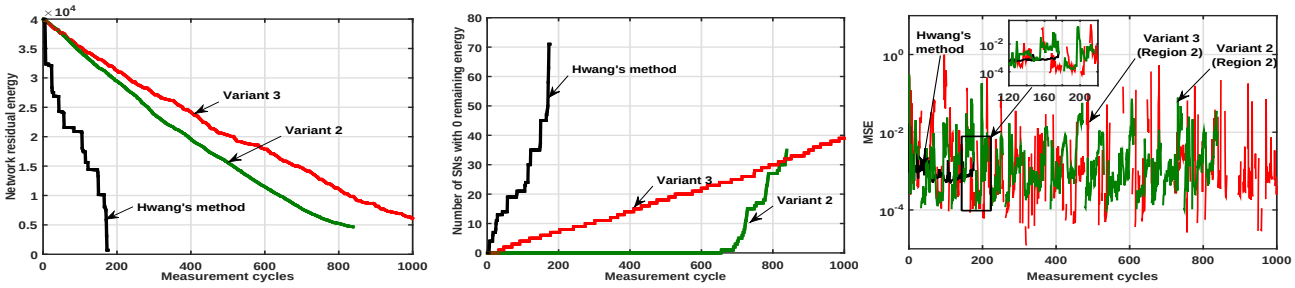
(a) Variant 1 (b) Variant 2

Fig. 4: Variation of total overhead with window size. $\eta = 10^{-7}$ J/bit/m^2 and $E_s = 0.0048$ J.

Note that w instances of the estimated signal $\widehat{\mathbf{z}}_r^{(\cdot)}$ are used in computing the matrix $\mathbf{B}_r^{(\cdot)}$ in every measurement cycle except for the one that follows the training/retraining phase. For the cycle immediately after training/retraining, K_{tr} instances of the actual signal $\mathbf{z}_r^{(\cdot)}$ are used. To mitigate the effect of error due to the use of previously estimated signal vectors in the $\mathbf{B}_r^{(\cdot)}$ matrix computation, it is also ensured that $w > K_{tr}$.

Remark 2. *The choice of w is required to limit the retraining overheads without significantly increasing the CH-to-CH communication overheads.*

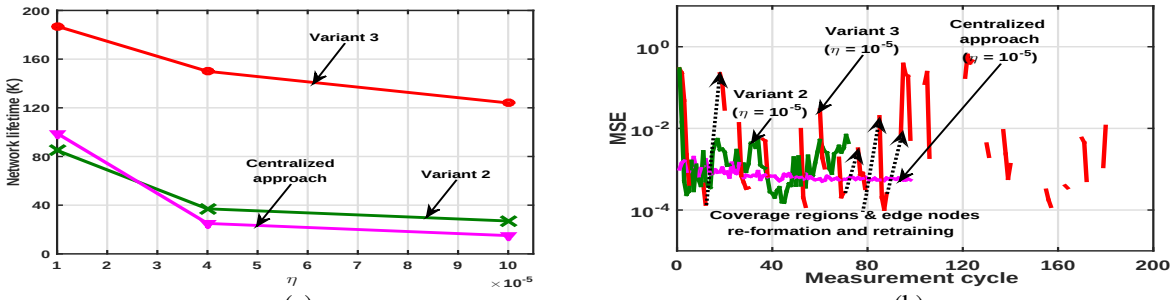
C. Performance comparison with state-of-the-art



(a) (b) (c)

Fig. 5: Comparison of (a) network residual energy, (b) number of SNs with 0 remaining energy, and (c) MSE of the proposed framework with Hwang's method [21]. $E_s = 0.0048$ J, $\eta = 10^{-7}$ J/bit/m^2, and $E_{rem}^{(0)}(\cdot) = 500$ J. The best performing variants are considered: variant 3 (maximum network life), variant 2 (maximum node life).

Performance of the competitive variants of the proposed decentralized sensing framework are now compared with the closest decentralized approach developed by Hwang *et al.* [21] and centralized framework [15] in Figs. 5 and 6 respectively. It can be observed from Figs. 5(a)-(c) that the proposed variants with standout performance (variant 3 with maximum network lifetime, variant 2 with maximum node lifetime) offer significant improvement in energy efficiency and node lifetime (or lower outage of network coverage) without compromising the sensing quality. Energy consumption per cycle in Hwang's method and that with variants 2 and 3 are respectively $223.274, 42.12,$ and 33.909 J/cycle. Thus, the variants 2 and 3 are respectively 81.135% and 84.813% more energy efficient compared to the Hwang's method. The gain in energy efficiency is



(a) (b)

Fig. 6: Comparison of (a) network lifetime versus power amplification parameter η and (b) MSE of the best performing variants: variant 3 (maximum network lifetime), variant 2 (maximum node lifetime) of the proposed decentralized framework and that of centralized framework [15]. Parameters values: E_s=0.48 J and $E_{rem}^{(0)}(\cdot)$=100 J.

due to non-iterative information exchange by the SNs in a measurement cycle and the adaptation mechanism in decentralized sensor selection. On the other hand, node-level energy saving is due to the consideration of unequal energy consumption and remaining energy information in the proposed framework. The discontinuities in Fig. 5(c) show that, whenever the sensing quality of a region becomes poor, the proposed retraining logic detects it, and retraining is conducted in that region. Due to space constraint, MSE plot of the estimated signal of only one region is shown. Further, these discontinuities occur during different measurement cycles in different regions; thus, plotting the MSE summed over all the regions is not possible at these points.

From Fig. 6(a) it can be observed that the proposed variants 2 and 3 outperform the centralized approach in [15] when the power amplification component $\eta \geq 2.5 \times 10^{-5}$. Compared to the centralized approach, these variants have reduced energy consumed in direct SN-FC communication. For the case $\eta < 2.5 \times 10^{-5}$, only the variant 3 performs better than the centralized one. The reason for the variant 2 not performing better in this case is that, the energy consumed in CH-to-CH communication overheads incurred in each cycle outweigh the energy gain achieved via reduced SN-FC communication overhead. However, the component of energy consumed in the SN-FC communication overhead in the proposed decentralized case are still lesser than that in the centralized case [15]. Note that, loss in sensing accuracy in some measurement cycles in Fig. 6(b) is addressed by the proposed retraining logic and is shown as discontinuities in the MSE curve. It can also be observed that the frequency of retraining is high in the later cycles. Due to low network residual energy in these cycles, there is a lesser flexibility in sensor selection from all the SNs in $\mathcal{R}_r$, as some of them may not participate owing to insufficient remaining energy. Although the proposed framework may select more number of SNs to satisfy $\text{BCRB}_r \in [\alpha, \beta]$, the resulting sensing quality may still be poor which results in frequent retraining of the region.

TABLE III: Performance gap: centralized versus decentralized BCRB ($R = 4$)

	$M^{(k)}$	12	16	20	24	28	32	36	40
Centralized	avg BCRB$^{(k)}$ ($\times 10^{-5}$)	3.287	2.671	2.326	2.157	2.096	2.062	2.0370	2.012
	avg MSE$^{(k)}$ ($\times 10^{-4}$)	11.019	9.258	7.440	6.291	5.653	5.270	4.978	4.786
	avg $\frac{E_{cons}}{E_{total}}$ (%)	46.277	48.105	50.234	52.654	55.355	58.362	61.656	65.243
Decentralized	avg $\sum_{r=1}^{R}$BCRB$_r^{(k)}$ ($\times 10^{-5}$)	15.961	12.424	10.223	8.977	8.233	7.572	7.177	6.834
	avg $\sum_{r=1}^{R}$MSE$_r^{(k)}$ ($\times 10^{-4}$)	196	190	8.2	6.747	6.007	5.547	5.167	4.937
	avg $\frac{E_{cons}}{E_{total}}$ (%)	46.177	47.392	48.405	49.205	49.856	50.576	51.352	52.181

An acceptable range of sensing quality (see [38]) achieved in the proposed framework validates suitability of the used BCRB. The BCRB window $[\alpha, \beta]$ is set as per the minimum and maximum BCRB values achieved in the respective comparative approaches. $\lambda_r^{(1)}, \forall r$, in (21) are set such that the minimum number of SNs are activated and the achieved BCRB$_r^{(1)} \in [\alpha, \beta], \forall r$. The usage of sparsification matrix, computed using w instances of previously estimated signal, in current cycle's signal estimation in the decentralized framework results in MSE variations (seen in Figs. 5(c) and 6(b)) owing to ensuing error which gets accumulated over cycles. These variations are absent in the Hwang's method and the centralized scheme because of the considered strong assumption of known sparsification matrix and the use of matrix computed using all the estimated signal instances available from the beginning of network operation respectively. To address the degradation of sensing quality, a retraining logic is proposed. Further, these variations can also be reduced by tightening the BCRB window at the expense of reduced network lifetime.

Remark 3. *Thus, the proposed edge computing framework offers energy-efficient decentralized sensing of a slowly varying process, thereby extending the network lifetime.*

D. BCRB$^{(k)}$ versus $\sum_{r=1}^{R}$BCRB$_r^{(k)}$

Assuming all the SNs active, the BCRB is decoupled as BCRB$^{(k)} = \sum_{r=1}^{R}$BCRB$_r^{(k)}$ (16). For the cases when only a few SNs are active, it is approximated as BCRB$^{(k)} < \sum_{r=1}^{R}$BCRB$_r^{(k)}$. The performance degradation caused by this approximation is tabulated in Table III. It can be observed by comparing respective BCRBs and MSEs that the performance loss decreases with increasing number of active SNs $\left(\text{i.e., } M^{(k)}\right)$. Further, the energy consumption percentage $(\frac{E_{cons}}{E_{total}}\%)$ of the centralized approach increases at a faster pace compared to the decentralized approach. Trade-off between the sensing quality and energy consumption can also be seen as the number of active SNs changes. The weight $\lambda^{(k)}$ of centralized $M^{(k)}$ is used to set the weights of decentralized problem as $\lambda_r^{(k)} = \lambda^{(k)},\ \forall r$. Further, it was observed that $M^{(k)}$ need not necessarily be equal to $\sum_{r=1}^{R} M_r^{(k)}$. The average BCRB values are computed after running centralized and

decentralized problems for 80 different sets of remaining energy of the SNs in $k = 1$ cycle.

Table IV summarizes the effect of number of clusters (R) on centralized and decentralized sensing performance, keeping $M^{(k)}$ same. It can be observed that as R increases (or cluster size decreases), the MSE of the centralized approach (10) decreases (because of improved feature selection) with no significant change in energy consumption. In the decentralized approach (18), $\sum_{r=1}^{R}$ BCRB$_r$ and $\sum_{r=1}^{R}$ MSE$_r$ increase with increase in R. The gap between BCRB of the centralized approach and $\sum_{r=1}^{R}$ BCRB$_r$ of the decentralized approach decreases with decreasing R. Similarly, the gap between MSE and $\sum_{r=1}^{R}$ MSE$_r$ decreases. This is because the approximation BCRB $\leq \sum_{r=1}^{R}$ BCRB$_r$ becomes tight as R decreases. Further, though the decentralized problem consumes lesser energy $(\frac{E_{cons}}{E_{total}}\%)$ compared to the centralized approach, the gap between two decreases with increasing R. This is because, the number of CHs increases with increased R, which results in an increased number of CH-FC communication. These communications consume significant energy due to long distances of the CHs from the FC.

TABLE IV: Impact of R on performance of centralized and decentralized selection

R	Centralized				Decentralized		
	$M^{(k)}$	avg BCRB$^{(k)}$	avg MSE$^{(k)}$	avg $\frac{E_{cons}}{E_{total}}$ (%)	avg $\sum_{r=1}^{R}$BCRB$_r^{(k)}$	$\sum_{r=1}^{R}$MSE$_r^{(k)}$	avg $\frac{E_{cons}}{E_{total}}$ (%)
2	12	3.203×10^{-5}	0.00586	46.2209	13.525×10^{-5}	0.0991	45.9833
4	12	3.287×10^{-5}	0.0011	46.2773	15.961×10^{-5}	0.0196	46.1773
6	12	3.820×10^{-5}	0.00066	46.9012	27.309×10^{-5}	0.0573	46.6825

E. Factors governing retraining in the network

Table V captures the number of times retraining occurs, using CH selection variant 4 for different values of R and w. By comparing the case $\{R, w\} = \{4, 6\}$ with the cases $\{2, 8\}$ and $\{6, 4\}$ it is observed that the retraining count decreases with the increase in window size w and/or decrease in the number of coverage regions R, and vice-versa. Further, by simultaneously increasing R and w (e.g., $\{2, 4\}$ to $\{6, 8\}$) or decreasing them (e.g., $\{6, 8\}$ to $\{3, 5\}$) no significant change in the retraining count is observed. The same trend is seen with the other variants.

Remark 4. *In the proposed framework, retraining count decreases as the reconstruction accuracy/window size w increases and degree of decentralization of WSN decreases.*

TABLE V: Retraining count for various $\{R, w\}$ values in variant 4. $E_{rem}^{(0)}(\cdot) = 500$ J, $E_s = 4.8$ J

$\{R, w\}$	$\{4, 6\}$	$\{2, 8\}$	$\{6, 4\}$	$\{2, 4\}$	$\{6, 8\}$	$\{3, 5\}$
Retraining count	15	1	41	7	8	10

F. Impact of various parameters on WSN's performance

An increase in value of β may increase network lifetime and decrease sensing quality because the constraint $\text{BCRB}_r \in [\alpha, \beta]$ will be satisfied with a lower number of active SNs (i.e., lower M_r). This effect was demonstrated for the centralized sensor selection proposed in a prior study [15, Figs. 2(c) and 3(c)]. Similar effect can be seen for α as well.

The parameters δ^{th} and ϵ^{th} quantify increase/decrease in the number of active SNs by effectively capturing the process variations. For a coverage region r, decreasing δ_r^{th} may result in frequent increase in the value of M_r which leads to decrease in sensing error and network lifetime. Intuitively, a process with high temporal correlation (or slower variations) could be effectively monitored using a lesser M_r. A higher value of $\delta_{(\cdot)}^{th}$ would be suitable for this. Further, the parameter $\epsilon_{(\cdot)}^{th}$ aid in quantifying the extent up to which the number of active SNs can be decreased for sensing in subsequent measurement cycle such that the resulting signal estimate will not be much different from the prior available signal estimates. It also gives an approximate idea of spatial correlation of the process as for high spatial correlation case, the heuristic $\delta_r^{(k)}$ (which is compared with δ_r^{th}) evaluates to lower values as mentioned in the work [15] as well for centralized sensing case. Increasing $\epsilon_{(\cdot)}^{th}$ would allow larger decrease in value of M_r which may result in increased network lifetime and MSE.

TABLE VI: Impact of number of coverage regions

R	2	3	4	5	6	7
Network Lifetime	173	188	194	192	179	175
Energy consumption per cycle (in J)	229.1995	218.7119	205.3688	208.0227	223.4185	228.1437
Retraining Count	4	13	15	24	27	38

On increasing the number of coverage regions (or the degree of decentralization), the sensing quality deteriorates faster which invokes frequent retraining. This results in an increase in number of retraining phases as illustrated in Table VI for variant- 2. However, as R increases, the energy consumption per cycle (network lifetime) first decreases (increases) and then increases (decreases). Due to this, a criterion is developed in Section IV-B to find appropriate R such that minimum energy is consumed per cycle. It may be noted that comment on the MSE is made based on retraining count because different regions undergo retraining during different measurement cycles due to which computing overall MSE was not possible.

G. Performance study using real WSN data-set

This section presents comparative performance results of the proposed variants and Hwang's method [21] and DiSparSense [18] using humidity data collected by a WSN deployed in Intel

TABLE VII: Performance comparison with state-of-the-art approaches

	Hwang's method: Gaussian random $\mathbf{B}$ (PCA-based $\mathbf{B}$)				Variant 2	Variant 3	DiSparSense	Exhaustive
M	12	17	25	30	–	–	–	32
Average NMSE	1.8039 (2.7813×10^{-4})	1.7697	0.3963	0.3695	2.4886×10^{-4}	2.0254×10^{-4}	0.5405	1.9295×10^{-5}
E_{cons}	71.3274 (110.1692)	92.2521	164.2230	213.7741	87.7705	79.1789	315.7181	132.6000
Network lifetime	18 (13)	10	8	7	18	19	5	12
Node lifetime	10 (9)	10	8	7	12	9	4	11

Berkeley research lab [39]. Signals sensed by a set of 32 Mica2Dot sensors spanning over 25×32 m^2 lab area are considered. The FC is considered to be located at 40 m above the center of the considered field span. Parameters $[\alpha, \beta], \delta_r^{th}, \epsilon_r^{th} \forall r, R, w, E_s, \eta, E_{elec},$ and $E_{rem}^{(0)}(\cdot)$ are respectively set as $[2.1628 \times 10^{-5}, 0.0522], [0.0709, 0.4892, \ \ 0.8305, 0.0532], 0.5 \ \forall r, 4, 6, 4.8$ J, 10^{-10} J/bit/m^2, 140 nJ/bit, 50 J. Performance metric used in the simulation plots is NMSE $= \frac{\left\|\mathbf{z}_r^{(k)} - \widehat{\mathbf{z}}_r^{(k)}\right\|^2}{\left\|\mathbf{z}_r^{(k)}\right\|^2}$. It gauges the error's energy against the signal energy. Note that it has not been directly used in the framework, as the WSN signal $\mathbf{z}_r^{(k)}$ is unknown. It can be observed from Table VII that the variants 2 (3) of the proposed framework provide 58.9424% (62.9614%) and 72.1997% (74.9210%) energy efficiency compared to Hwang's method (Gaussian $\mathbf{B}, M = 30$) and DiSparSense, respectively. Sensing quality of the proposed variants is significantly better than the competitive schemes. Practical significance of the data-driven PCA-based sparsification matrix $\mathbf{B}$ of the proposed framework can be seen from the improvement achieved in the sensing quality when this $\mathbf{B}$ is used in Hwang's method instead of Gaussian $\mathbf{B}$ matrix. Further, it can be verified that the variant 3 (respectively, 2) offers better network (respectively, node) lifetime. Unlike the competitive schemes (Hwang's and DiSparSense), a higher gain of decentralization (i.e., energy efficiency) is achieved by paying lower price (i.e., sensing quality) when the proposed framework is compared with the centralized exhaustive scheme (with all active SNs).

VII. Conclusion

In this paper, a novel edge computing framework for decentralized sensing of a spatio-temporal process has been proposed. For this, a critical trade-off between the sensing quality and energy efficiency has been distributedly optimized at the dynamically selected edge nodes. Residual energy of the SNs has been incorporated in the formulation which was otherwise ignored in the literature. The framework adapts the number of active SNs in a coverage region according to the process dynamics and keeps the sensing error within a given limit. To limit the accumulation of estimation error over the measurement cycles, a logic to detect the need for retraining the network

coverage regions has also been proposed. Extensive simulation results have demonstrated that, compared to the closest competitive decentralized approach the proposed framework provides up to 84% higher energy efficiency (network lifetime) and improves energy balance among the SNs without impacting the sensing quality. The energy efficiency of the proposed decentralized framework has been verified by executing it on a real WSN data-set as well.

APPENDIX A
SIGNAL SPARSIFICATION: A PCA-BASED APPROACH

At the onset, K_{tr} training instances of signal vector $\mathbf{z}^{(\cdot)}$, forming a learning set $\mathcal{T} = \left\{\mathbf{z}^{(k-K_{tr}+1)}, \ldots, \mathbf{z}^{(k-1)}, \mathbf{z}^{(k)}\right\}$, are considered to compute the sample mean vector $\bar{\mathbf{z}}_{\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{\mathbf{z}^{(\cdot)}\in\mathcal{T}} \mathbf{z}^{(\cdot)} \in \mathbb{R}^{N\times 1}$ and the sample covariance matrix $\widehat{\mathbf{\Sigma}}_{\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{\mathbf{z}^{(\cdot)}\in\mathcal{T}} \left(\mathbf{z}^{(\cdot)} - \bar{\mathbf{z}}_{\mathcal{T}}\right)\left(\mathbf{z}^{(\cdot)} - \bar{\mathbf{z}}_{\mathcal{T}}\right)^T \in \mathbb{R}^{N\times N}$. The sparsification matrix $\mathbf{B}^{(k)} \in \mathbb{R}^{N\times N}$ is formed using orthogonal eigen vectors of the matrix $\widehat{\mathbf{\Sigma}}_{\mathcal{T}}$, while its eigenvalues form the diagonal elements of a diagonal matrix $\mathbf{\Gamma}^{(k)} \in \mathbb{R}^{N\times N}$ (used later), i.e., $\widehat{\mathbf{\Sigma}}_{\mathcal{T}} = \mathbf{B}^{(k)}\mathbf{\Gamma}^{(k)}\left(\mathbf{B}^{(k)}\right)^T$. Owing to the correlation present in elements of the set $\mathcal{T}$, projection of $\mathbf{z}^{(k)}$ onto vector space $\mathcal{R}\left(\mathbf{B}^{(k)}\right)$ gives the corresponding sparse vector $\mathbf{x}^{(k)}$ as $\mathbf{x}^{(k)} = \left(\mathbf{B}^{(k)}\right)^T\left(\mathbf{z}^{(k)} - \bar{\mathbf{z}}_{\mathcal{T}}\right)$. Substituting the sparse representation $\mathbf{z}^{(k)} = \bar{\mathbf{z}}_{\mathcal{T}} + \mathbf{B}^{(k)}\mathbf{x}^{(k)}$ in (2) and denoting $\mathbf{y}^{(k)} = \widetilde{\mathbf{y}}^{(k)} - \mathbf{A}^{(k)}\bar{\mathbf{z}}_{\mathcal{T}} \in \mathbb{R}^{M_k\times 1}$, the system model becomes $\mathbf{y}^{(k)} = \mathbf{A}^{(k)}\mathbf{B}^{(k)}\mathbf{x}^{(k)} + \mathbf{n}^{(k)}$.

Likewise, for the regional system model (4), compute $\bar{\mathbf{z}}_{\mathcal{T}_r}$ and $\widehat{\mathbf{\Sigma}}_{\mathcal{T}_r}$ using the learning set $\mathcal{T}_r = \{\mathbf{z}_r^{(k-K_{tr}+1)}, \ldots, \mathbf{z}_r^{(k-1)}, \mathbf{z}_r^{(k)}\}$. Note that $\widehat{\Sigma}_{\mathcal{T}_r}$ is a full rank matrix irrespective of whether N_r is greater than, less than, or equal to K_{tr}. The eigen vectors of $\widehat{\mathbf{\Sigma}}_{\mathcal{T}_r}$ give a unique regional sparsification matrix $\mathbf{B}_r^{(k)}$; while the eigenvalues form the diagonal elements of the diagonal matrix $\Gamma_r^{(k)} \in \mathbb{R}^{N_r\times N_r}$. Following the same procedure and denoting $\mathbf{y}_r^{(k)} = \widetilde{\mathbf{y}}_r^{(k)} - \mathbf{A}_r^{(k)}\bar{\mathbf{z}}_{\mathcal{T}_r}$, the sparse representation of the system model (4) is given by $\mathbf{y}_r^{(k)} = \mathbf{A}_r^{(k)}\mathbf{B}_r^{(k)}\mathbf{x}_r^{(k)} + \mathbf{n}_r^{(k)}, \forall r$.

APPENDIX B
SBL APPROACH FOR SIGNAL RECOVERY

The SBL framework assumes a Gaussian likelihood model for the estimation of the underlying monitored phenomenon [23]. During each cycle k, the SBL scheme assigns a parameterized Gaussian prior to the unknown weight vector $\mathbf{x}_r^{(k)}$ as

$$p\left(\mathbf{x}_r^{(k)}; \boldsymbol{\gamma}_r^{(k)}\right) = \prod_{n=1}^{N}\left(2\pi\gamma_r^{(k)}(n)\right)^{-1/2} e^{-\frac{\left(x_r^{(k)}(n)\right)^2}{2\gamma_r^{(k)}(n)}}. \tag{34}$$

The component $\gamma_r^{(k)}(n)$ of the hyperparameter vector $\boldsymbol{\gamma}_r^{(k)} = \left[\gamma_r^{(k)}(1), \ldots, \gamma_r^{(k)}(N_r)\right]^T \in \mathbb{R}^{N_r\times 1}$ corresponds to the variance of component $x_r^{(k)}(\cdot)$ of the weight vector. Due to the intractability of the maximum likelihood estimate of $\boldsymbol{\gamma}_r^{(k)}$, i.e. $\widehat{\boldsymbol{\gamma}}_r^k = \arg\max_{\boldsymbol{\gamma}_r^{(k)}\succeq 0} \log p\left(\mathbf{y}_r^{(k)}; \boldsymbol{\gamma}_r^{(k)}\right)$ [23], an iterative expectation maximization (EM) algorithm is employed to estimate $\boldsymbol{\gamma}_r^{(k)}$ considering $\mathbf{x}_r^{(k)}$ as a hidden variable. Let the estimate of the hyperparameter vector $\boldsymbol{\gamma}_r^{(k)}$ in the

l^{th} EM iteration be denoted as $\left(\widehat{\boldsymbol{\gamma}}_r^{(k)}\right)^{(l)}$. The E-step in the l^{th} iteration evaluates the likelihood as $\mathcal{L}\left(\boldsymbol{\gamma}_r^{(k)} \mid \left(\boldsymbol{\gamma}_r^{(k)}\right)^{(l)}\right) = \mathrm{E}_{\mathbf{x}_r^{(k)}|\mathbf{y}_r^{(k)};\left(\boldsymbol{\gamma}_r^{(k)}\right)^{(l)}}\left\{\ \log p\left(\mathbf{y}_r^{(k)}, \mathbf{x}_r^{(k)}; \boldsymbol{\gamma}_r^{(k)}\right)\right\}$. The posterior distribution of $\mathbf{x}_r^{(k)}$ is $p\left(\mathbf{x}_r^{(k)}|\mathbf{y}_r^{(k)};\left(\boldsymbol{\gamma}_r^{(k)}\right)^{(l)}\right) \sim \mathcal{N}\left(\boldsymbol{\mu}_{x_r^{(k)}}^{(l)}, \boldsymbol{\Sigma}_{x_r^{(k)}}^{(l)}\right)$ with $\boldsymbol{\mu}_{x_r^{(k)}}^{(l)} = \sigma^{-2}\boldsymbol{\Sigma}_{x_r^{(k)}}^{(l)}\left(\boldsymbol{\Theta}_r^{(k)}\right)^T \mathbf{y}_r^{(k)} \in \mathbb{R}^{N_r \times 1}, \boldsymbol{\Sigma}_{x_r^{(k)}}^{(l)}$ $= \left(\sigma^{-2}\left(\boldsymbol{\Theta}_r^{(k)}\right)^T \boldsymbol{\Theta}_r^{(k)} + \left(\left(\widehat{\boldsymbol{\Gamma}}_r^{(k)}\right)^{(l)}\right)^{-1}\right)^{-1} \in \mathbb{R}^{N_r \times N_r}$, $\left(\widehat{\boldsymbol{\Gamma}}_r^{(k)}\right)^{(l)} = \mathrm{diag}\left(\left(\widehat{\gamma}_r^{(k)}(1)\right)^{(l)}, \cdots, \left(\widehat{\gamma}_r^{(k)}(N_r)\right)^{(l)}\right)$. A less complex expression of $\boldsymbol{\Sigma}_{x_r^{(k)}}^{(l)}$ is obtained using the matrix inversion lemma [26] as $\boldsymbol{\Sigma}_{x_r^{(k)}}^{(l)} = \left(\widehat{\boldsymbol{\Gamma}}_r^{(k)}\right)^{(l)} - \left(\widehat{\boldsymbol{\Gamma}}_r^{(k)}\right)^{(l)}\left(\boldsymbol{\Theta}_r^{(k)}\right)^T\left(\boldsymbol{\Sigma}_{y_r^{(k)}}^{(l)}\right)^{-1}\boldsymbol{\Theta}_r^{(k)}\left(\widehat{\boldsymbol{\Gamma}}_r^{(k)}\right)^{(l)}$ with $\boldsymbol{\Sigma}_{y_r^{(k)}}^{(l)} = \left(\sigma^2\mathbf{I}_{M_r^{(k)}} + \boldsymbol{\Theta}_r^{(k)}\left(\widehat{\boldsymbol{\Gamma}}_r^{(k)}\right)^{(l)}\left(\boldsymbol{\Theta}_r^{(k)}\right)^T\right) \in \mathbb{R}^{M_r \times M_r}$. Thereafter, the M-step estimates $\left(\widehat{\boldsymbol{\gamma}}_r^{(k)}\right)^{(l+1)}$ as:

$$\left(\widehat{\boldsymbol{\gamma}}_r^{(k)}(n)\right)^{(l+1)} = \arg\max_{\boldsymbol{\gamma}_r^{(k)}(n)} \mathrm{E}_{\mathbf{x}_r^{(k)}|\mathbf{y}_r^{(k)};\left(\boldsymbol{\gamma}_r^{(k)}\right)^{(l)}}\left\{\log p\left(\mathbf{y}_r^{(k)}, \mathbf{x}_r^{(k)}; \boldsymbol{\gamma}_r^{(k)}\right)\right\} = \boldsymbol{\Sigma}_{x_r^{(k)}}^{(l)}(n,n) + \left(\mu_{x_r^{(k)}}^{(l)}(n)\right)^2. \quad (35)$$

After repeating the E and M-step for L_{EM} iterations, the final sparse estimate is obtained as $\widehat{\mathbf{x}}_r^{(k)} = \boldsymbol{\mu}_{x_r^{(k)}}^{(L_{EM})}$. Without prior knowledge of sparsity, the SBL-EM scheme provides a maximally sparse solution $\left(\left(\widehat{\gamma}_r^{(k)}(n)\right)^{(l)} \to 0 \Rightarrow \left(\widehat{x}_r^{(k)}(n)\right)^{(l)} \to 0\right)$. The estimate $\widehat{\mathbf{z}}_r^{(k)}$ is given by $\overline{\mathbf{z}}_{\mathcal{T}_r} + \mathbf{B}_r^{(k)}\widehat{\mathbf{x}}_r^{(k)}$.

APPENDIX C
BAYESIAN CRAMÉR-RAO BOUND (BCRB)

BCRB characterizes the MSE in the estimate of the unknown vector $\mathbf{x}^{(k)}$ in the signal model (6). $\mathrm{BCRB}^{(k)} = \mathrm{Tr}\left\{\mathbf{J}_B^{-1}\right\}$, with the Bayesian Fisher information matrix $\mathbf{J}_B \in \mathbb{R}^{N \times N}$ given below [15]: $\mathbf{J}_B = \underbrace{-\mathrm{E}_{\left(\mathbf{y}^{(k)},\mathbf{x}^{(k)}\right)}\left\{\frac{\partial^2\mathcal{L}\left(\mathbf{y}^{(k)}|\mathbf{x}^{(k)};\boldsymbol{\Gamma}^{(k)}\right)}{\partial\mathbf{x}^{(k)}\left(\partial\mathbf{x}^{(k)}\right)^T}\right\}}_{\mathbf{J}_D} \underbrace{-\mathrm{E}_{\left(\mathbf{x}^{(k)}\right)}\left\{\frac{\partial^2\mathcal{L}\left(\mathbf{x}^{(k)};\boldsymbol{\Gamma}^{(k)}\right)}{\partial\mathbf{x}^{(k)}\left(\partial\mathbf{x}^{(k)}\right)^T}\right\}}_{\mathbf{J}_P}$. The terms $\mathcal{L}\left(\mathbf{y}^{(k)}|\mathbf{x}^{(k)};\boldsymbol{\Gamma}^{(k)}\right), \mathcal{L}\left(\mathbf{x}^{(k)};\boldsymbol{\Gamma}^{(k)}\right)$, and $\mathbf{J}_D, \mathbf{J}_P \in \mathbb{R}^{N \times N}$ respectively denote the log-likelihood functions of vectors $\mathbf{y}^{(k)}$, $\mathbf{x}^{(k)}$ parameterized by $\boldsymbol{\Gamma}^{(k)}$, and FIMs with respect to $\mathbf{y}^{(k)}, \mathbf{x}^{(k)}$. The prior distribution of $\mathbf{x}^{(k)}$ (34) is used to compute the log-likelihood function as: $\mathcal{L}\left(\mathbf{x}^{(k)};\boldsymbol{\Gamma}^{(k)}\right) = \left(\widetilde{k} - \frac{1}{2}\left(\mathbf{x}^{(k)}\right)^T\left(\boldsymbol{\Gamma}^{(k)}\right)^{-1}\mathbf{x}^{(k)}\right)$, where $\widetilde{k}$ is a parametric constant. Further, its second order derivative with respect to $\mathbf{x}^{(k)}$ evaluates to $\frac{\partial^2\mathcal{L}\left(\mathbf{x}^{(k)}\right)}{\partial\mathbf{x}^{(k)}\left(\partial\mathbf{x}^{(k)}\right)^T} = \left(\boldsymbol{\Gamma}^{(k)}\right)^{-1}$. Thus, $\mathbf{J}_P = \left(\boldsymbol{\Gamma}^{(k)}\right)^{-1}$. Similarly, after ignoring the constant terms, the log-likelihood function $\mathcal{L}\left(\mathbf{y}^{(k)};\mathbf{x}^{(k)}\right)$ is obtained as: $\mathcal{L}\left(\mathbf{y}^{(k)}|\mathbf{x}^{(k)};\boldsymbol{\Gamma}^{(k)}\right) = \frac{1}{2\sigma^2}\left\|\mathbf{y}^{(k)} - \mathbf{A}^{(k)}\mathbf{B}^{(k)}\mathbf{x}^{(k)}\right\|^2$. Thus, $\mathbf{J}_D = \frac{1}{\sigma^2}\left(\mathbf{B}^{(k)}\right)^T\left(\mathbf{A}^{(k)}\right)^T\mathbf{A}^{(k)}\mathbf{B}^{(k)}$. The BCRB is expressed as:

$$\mathrm{BCRB}^{(k)} = \mathrm{Tr}\left\{\left(\frac{1}{\sigma^2}\left(\mathbf{B}^{(k)}\right)^T\left(\mathbf{A}^{(k)}\right)^T\mathbf{A}^{(k)}\mathbf{B}^{(k)} + \left(\boldsymbol{\Gamma}^{(k)}\right)^{-1}\right)^{-1}\right\}. \quad (36)$$

For colored noise case, BCRB evaluates to: $\mathrm{BCRB}^{(k)} = \mathrm{Tr}\left\{\left(\left(\mathbf{B}^{(k)}\right)^T(\boldsymbol{\Sigma}_n)^{\frac{1}{2}}\,\mathrm{diag}\left(\widetilde{\mathbf{a}}^{(k)}\right)(\boldsymbol{\Sigma}_n)^{\frac{1}{2}}\,\mathbf{B}^{(k)} + \left(\boldsymbol{\Gamma}^{(k)}\right)^{-1}\right)^{-1}\right\}$.

REFERENCES

[1] X. Cao, L. Liu, Y. Cheng, and X. Shen, "Towards energy-efficient wireless networking in the big data era: A survey," *IEEE Commun. Surveys Tut.*, 1st Quart. 2017.

[2] V. Gupta, S. Tripathi, and S. De, "Green sensing and communication: A step towards sustainable IoT systems," *Springer J. Indian Inst. Sci.*, pp. 1–16, 2020.

[3] G. Anastasi, M. Conti, M. Di Francesco, and A. Passarella, "Energy conservation in wireless sensor networks: A survey," *Ad hoc netw.*, vol. 7, no. 3, pp. 537–568, May 2009.

[4] M. Hooshmand, M. Rossi, D. Zordan, and M. Zorzi, "Covariogram-based compressive sensing for environmental wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 6, pp. 1716–1729, Mar. 2016.

[5] Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3816–3827, Jul. 2010.

[6] T. Xue, X. Dong, and Y. Shi, "Multiple access and data reconstruction in wireless sensor networks based on compressed sensing," *IEEE Trans. Wireless Commun.*, vol. 12, no. 7, pp. 3399–3411, Jul. 2013.

[7] Y. Chen and Q. Zhao, "On the lifetime of wireless sensor networks," *IEEE Commun. Lett.*, vol. 9, no. 11, pp. 976–978, Nov. 2005.

[8] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.

[9] S. P. Chepuri and G. Leus, "Sparsity-promoting sensor selection for non-linear measurement models," *IEEE Trans. Signal Process.*, vol. 63, no. 3, pp. 684–698, Feb. 2015.

[10] O. M. Bushnaq, A. Chaaban, and T. Al-Naffouri, "Joint sensor location/power rating optimization for temporally-correlated source estimation," in *Proc. IEEE Intl. Wksp. Signal Process. Advances Wireless Commun. (SPAWC)*. Sapporo, Japan, 2017, pp. 1–5.

[11] O. M. Bushnaq, A. Chaaban, S. P. Chepuri, G. Leus, and T. Y. Al-Naffouri, "Sensor placement and resource allocation for energy harvesting IoT networks," *Elsevier Digital Signal Process.*, vol. 105, pp. 1–14, 2020.

[12] W. Chen and I. J. Wassell, "Optimized node selection for compressive sleeping wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 65, no. 2, pp. 827–836, Feb. 2016.

[13] G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3447–3461, Oct. 2012.

[14] J. Hao, B. Zhang, Z. Jiao, and S. Mao, "Adaptive compressive sensing based sample scheduling mechanism for wireless sensor networks," *Pervasive and Mobile Comput.*, vol. 22, pp. 113–125, Sep. 2015.

[15] V. Gupta and S. De, "SBL-based adaptive sensing framework for WSN-assisted IoT applications," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4598–4612, Dec. 2018.

[16] V. Gupta and S. De, "Adaptive multi-sensing in EH-WSN for smart environment," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. IEEE, 2019, pp. 1–6.

[17] H. Jamali-Rad, A. Simonetto, and G. Leus, "Sparsity-aware sensor selection: Centralized and distributed algorithms," *IEEE Signal Process. Lett.*, vol. 21, no. 2, pp. 217–220, Feb. 2014.

[18] H. Jamali-Rad, A. Simonetto, X. Ma, and G. Leus, "Distributed sparsity-aware sensor selection," *IEEE Trans. Signal Process.*, vol. 63, no. 22, pp. 5951–5964, Nov. 2015.

[19] S. Liu, S. P. Chepuri, G. Leus, and A. O. Hero, "Distributed sensor selection for field estimation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*. New Orleans, LA, USA, Mar. 2017, pp. 4257–4261.

[20] F. Altenbach, S. Corroy, G. Böcherer, and R. Mathar, "Strategies for distributed sensor selection using convex optimization," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*. Anaheim, CA, USA, Dec. 2012, pp. 2367–2372.

[21] S. Hwang, R. Ran, J. Yang, and D. K. Kim, "Multivariated Bayesian compressive sensing in wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 7, pp. 2196–2206, Apr. 2015.

[22] R. Masiero, G. Quer, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, "Data acquisition through joint compressive

sensing and principal component analysis," in *Proc. IEEE Global Telecommun. Conf.* Honolulu, HI, USA, Nov. 2009, pp. 1–6.

[23] D. P. Wipf and B. D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Process.*, vol. 52, no. 8, pp. 2153–2164, Aug. 2004.

[24] A. Mishra, V. Gupta, S. Dwivedi, A. K. Jagannatham, and P. K. Varshney, "Sparse bayesian learning-based target imaging and parameter estimation for monostatic MIMO radar systems," *IEEE Access*, vol. 6, pp. 68 545–68 559, 2018.

[25] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge univ. press, 2004.

[26] K. B. Petersen, M. S. Pedersen *et al.*, "The matrix cookbook," *Tech. Univ. Denmark*, vol. 7, p. 15, Nov. 2008.

[27] Z. Fei, B. Li, S. Yang, C. Xing, H. Chen, and L. Hanzo, "A survey of multi-objective optimization in wireless sensor networks: metrics, algorithms, and Open Problems," *IEEE Commun. Surveys Tut.*, vol. 19, no. 1, pp. 550–586, 1st Quart. 2017.

[28] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[29] A. D. Marbini and L. E. Sacks, "Adaptive sampling mechanisms in sensor networks," in *Proc. London Commun. Symp.*, vol. 174. London, UK, Sep. 2003.

[30] M. A. Razzaque and S. Dobson, "Energy-efficient sensing in wireless sensor networks using compressed sensing," *Sensors*, vol. 14, no. 2, pp. 2822–2859, 2014.

[31] V. Tiwari, S. Malik, and A. Wolfe, "Power analysis of embedded software: a first step towards software power minimization," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 2, no. 4, pp. 437–445, Dec. 1994.

[32] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, Oct. 2002.

[33] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for k-medoids clustering," *Expert syst. appl.*, vol. 36, no. 2, pp. 3336–3341, 2009.

[34] 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.

[35] J. V. Zidek, W. Sun, and N. D. Le, "Designing and integrating composite networks for monitoring multivariate Gaussian pollution fields," *J. Royal Statist. Soc.: Series C (Appl. Statist.)*, vol. 49, no. 1, pp. 63–79, 2000.

[36] A. Sinha and A. P. Chandrakasan, "JouleTrack: a web based tool for software energy profiling," in *Proc. Des. Autom. Conf.* ACM, Jun. 2001, pp. 220–225.

[37] V. Gupta and S. De, "Collaborative Multi-sensing in Energy Harvesting Wireless Sensor Networks," *IEEE Trans. Signal Inform. Process. over Netw.*, vol. 6, pp. 426–441, May 2020.

[38] D. Hooper, J. Coughlan, and M. Mullen, "Structural equation modelling: Guidelines for determining model fit," *The Electron. J. Bus. Res. Methods*, vol. 6, pp. 53–60, Jan. 2008.

[39] P. Bodik, W. Hong, C. Guestrin, S. Madden, M. Paskin, and R. Thibaux, "Intel lab data," *Online dataset*, Feb. 2004.