

Article

# Coalition Formation Based Compressive Sensing in Wireless Sensor Networks

Alireza Masoum <sup>\*</sup>, Nirvana Meratnia <sup>†</sup> and Paul J. M. Havinga <sup>†</sup>

Pervasive Systems, Department of Computer Science, University of Twente, Drienerlolaan 5, 7522 NB Enschede, The Netherlands; n.meratnia@utwente.nl (N.M.); P.J.M.Havinga@utwente.nl (P.J.M.H.)

<sup>\*</sup> Correspondence: masoum@ieee.org; Tel.: +31-68-251-0228

<sup>†</sup> These authors contributed equally to this work.

Received: 13 May 2018; Accepted: 13 July 2018; Published: 18 July 2018



**Abstract:** Compressive sensing originates in the field of signal processing and has recently become a topic of energy-efficient data gathering in wireless sensor networks. In this paper, we propose an energy efficient distributed compressive sensing solution for sensor networks. The proposed solution utilizes sparsity distribution of signals to group sensor nodes into several coalitions and then implements localized compressive sensing inside coalitions. This solution improves data-gathering performance in terms of both data accuracy and energy consumption. The approach curbs both data-transmission costs and number of measurements. Coalition-based data gathering cuts transmission costs, and the number of measurements is reduced by scheduling sensor nodes and adjusting their sampling frequency. Our simulation showed that our approach enhances network performance by minimizing energy cost and improving data accuracy.

**Keywords:** compressive sensing; coalition; sparsity; belief propagation; joint sparse recovery

## 1. Introduction

Energy efficiency is a continuing concern within wireless sensor networks. Every sensor network is operational as long as they have enough energy resources. Therefore, to ensure the longevity of a network, energy efficient techniques are essential [1]. Compressive sensing is one of the new energy efficient compression based data gathering techniques that is introduced in recent years.

Compressive sensing is a concept originated from the field of signal processing. The promise of compressive sensing is that it can reconstruct sparse or compressible signals from a small number of measurements without having a priori knowledge about the signal structure. This technique utilizes information rate instead of sampling rate to sample and recover the signal [2].

Unlike classic compression techniques, compressive sensing simplifies encoding procedures while making the decoding procedure more complicated. This characteristic is well suited for wireless sensor network data gathering applications. It employs simple encoding on the resource-restricted sensor nodes and a complex decoding procedure at the powerful base station.

Recent compressive sensing solutions proposed in wireless sensor networks have proven advantages in minimizing the number of measurements, but they are still not competitive with the existing data compression techniques [3]. Since transmission cost is the dominant energy consumption parameter, compressive sensing methods should consider minimizing the consumption by each measurement in terms of transmission cost. In some proposed solutions, each sensor node produces its measurements, aggregates these with the measurements of other nodes and transfers them to the base station. Since sensor nodes may be far from each other, the transmission cost is high enough to negate the advantages of compressive sensing-based data gathering [4]. Utilizing distributed

compressive sensing could represent an important step towards improving existing compressive sensing methods in terms of energy consumption, bandwidth use and data quality [5].

In this paper, we propose a new distributed compressive sensing based data gathering approach which utilizes spatial-temporal correlation to improve the network performance in terms of energy consumption and data reconstruction accuracy. In the first step, we propose a spatial correlation based coalition formation algorithm to localize data collection of the sensor nodes. This approach leverages the sparsity distribution of signals in order to group spatially correlated sensor nodes into coalitions. Inside each coalition, we use the sparsity distribution of signals to define a utility function. This function later is used to minimize the number of active sensor nodes, which reduces energy consumption.

After coalition formation, a spatial temporal correlation based compressive sensing solution is proposed inside each coalition. The proposed solution employs a block diagonal measurement matrix to produce a linear combination of sensor node readings. The block diagonal measurement matrix is structured in a way that balances the computation and communication load over the coalitions. This spatial-temporal correlation based compressive sensing is used inside each coalition to compress sensor node readings and transfer it to the base station. Upon receiving compressed data, the base station applies a joint sparse signal recovery mechanism to reconstruct compressed data. In the first step, a joint sparse signal recovery is run to find the common sparsity profile among the coalitions. This recovery procedure then executes inside each coalition to achieve a common profile among sensor nodes. Utilizing this recovery algorithm helps to achieve better accuracy in data reconstruction while the number of required measurements is reduced.

On the other hand, this paper analyses the impact of different transform basis on compressive sensing. The compressible or sparse signal may be represented in different transform bases. Choosing the appropriate transform basis is essential in sparse representation, which can lead to less measurements and more accurate signal reconstruction during the recovery phase. To the best of our knowledge, the selection of an appropriate transform basis has not been adequately studied, as most existing compressive sensing techniques have focussed on optimizing reconstruction algorithms. This paper classifies different signals based on their characteristics and investigates the assignment of an appropriate transform basis for each specific type of signal.

The remainder of this paper is organized as follows: Section 2 presents the related works while Section 3 represents network architecture. Section 4 introduces an overall overview of the proposed solution. Sections 5 and 6 explain coalition formation algorithm and data gathering procedure inside coalitions, respectively. The data reconstruction procedure is mentioned in Section 7 while Section 8 analyzes the performance of the proposed scheme. Finally, conclusions are drawn in Section 9.

## 2. Related Works

The study presented in [6] introduces a belief-propagation-based compressive sensing technique that considers the temporal correlation model of the signal as given information and employs it to reconstruct the signal. To model temporal correlation, one can decrease the number of required measurements and help to achieve better reconstruction accuracy with a specific spike-and-slab Markov model. The accuracy of this Markov model changes over time, and this can influence reconstruction accuracy. Therefore, this algorithm periodically analyzes the model parameters and employs an online estimator to predict its parameter. This estimator is based on sequential expectation maximization (EM) algorithm that is derived by maximizing the Kullback–Leibler (KL) information measure. At the end of each reconstruction period, this estimator refers to reconstructed signal coefficients for the last time period to update the Markov model parameters. This approach focusses on robust accurate data reconstruction and decreases the number of measurements. However, the energy efficiency of this solution is not addressed explicitly.

The authors of [7] discuss how to make the encoding process more efficient in terms of resource management strategy. They proposed a compressive sensing approach that considers temporal correlation among sensor node readings to recover the original signal. They combined their

compressive sensing approach with gossip-based routing to recover the sparse signal in multi-hop wireless sensor networks.

In the case of telemonitoring applications, most signals, like Electrocardiography (ECG) signals, are not sparse in the time domain. Therefore, current compressive sensing solutions cannot provide accurate reconstruction. To address this problem, one study [8] addresses a block-sparse Bayesian learning-based compression technique. The proposed solution represents an extreme performance improvement in recovering non-sparse signals. Furthermore, using the block structure to recover signals lessens reconstruction complexity, which results in lower computational cost and better accuracy.

A combination of model-based data prediction and compressive sensing is addressed in [9] to achieve energy-efficient data gathering in sensor networks. Compressive sensing facilitates the compression of sensor node readings while an adaptive data prediction model is built on this basis, reducing the data sampling frequency. Using this data prediction model, sensor nodes do not require any prior information regarding monitored signals and can adapt their sampling rates to the time-varying sparsity level of the signals. Minimizing the data-sampling rate based on the sparsity level of the signal provided by compressive sensing attenuates the data-transmission rate, which conserves power.

Luo et al. [10] present a compressive data-gathering (CDG) method, which combines data compression and routing for data collection in large-scale wireless sensor networks. The result of this combination is a balanced distribution of energy drain over the network, which improves network lifetime. Furthermore, it shows a significant reduction in communication costs. With regard to scalability, this solution cannot be applied in small-scale networks and large-scale dynamic networks.

A further study [11] introduces a joint sparse signal recovery method which assumes sensor nodes have been distributed into different clusters, and uses a joint sparse signal recovery mechanism (instead of an independent reconstruction process) to recover the compressed data. This solution minimizes energy expenses in terms of data compression and transmission rates. However, it does not ensure that application-defined data accuracy requirements are met.

The authors of [12] target the event-detection application of wireless sensor networks in which the number of sensor nodes detecting an event is considerably lower than the number of all sensor nodes in the network. This may represent sparsity of events. The proposed compressive sensing-based event detection approach recovers the signal through a Bayesian reconstruction algorithm. Significant reductions in the number of measurements, as well as increased event detection probability, are the main performance achievements of this solution.

Realization and detecting an event region is an issue, which is addressed in [13]. The authors suggest a decomposition based compressive sensing (DBCS) data gathering solution for continuous monitoring of event region. The proposed solution is integrated with a Markov random field model to categorize event regions into separate homogenous regions. In this model, each sensor node contributes to detection procedure by providing local estimation the neighborhood information and its local observation. Performance analysis shows that this solution can detect the event regions accurately while minimizing the number of data transmissions required for accurate data reconstruction. A compressed network coding-based distributed data storage (CNCDS) method is proposed in [14]. It takes advantage of spatial correlation among sensor node measurements and aims primarily to achieve good recovery accuracy. Furthermore, it concentrates on reducing the number of data transmission and reception parameters, which results in energy-efficient data gathering. To do so, the proposed solution adjusts the probability of forwarding received data packets to the number of neighbor nodes. This approach ignores the noise and error parameters in the transmission scenario.

The research presented in [15] is one of the first studies to propose a compressive sensing-based data-gathering solution based on a random walk algorithm. The authors employ this algorithm to produce a non-uniform random measurement matrix. To do so, randomly selected sensor nodes in each walk generate a measurement based on a linear combination of both the node itself and

neighboring nodes' observations. At the end of each random walk, the measurements produced are forwarded to the base station. Compared to the existing random projection solutions, this approach shows a significant reduction in data-transmission cost.

Authors in [16] have mentioned that spatial-temporal correlation based compressive sensing solutions impose accuracy degradation in signal reconstruction procedure. To address this issue, they have proposed a weighted spatial-temporal compressive sensing solution that improves the data quality metric. However, it increases the number of data transmissions inside the network, which is still acceptable in the proposed solution.

A compressive sensing-based false data detection and correlation framework for crowd sourcing applications is presented in [17]. In crowd sourcing applications, a large number of participants employ their mobile devices to collect environmental data and report it to the base station. Misbehaving or erroneous participants can produce faulty data and packet loss in the network. Missing or faulty data is reconstructed with a spatial-temporal correlation-based compressive sensing technique. This approach considers collected data as multidimensional time-series and groups sensor nodes into smaller grids based on their location. In the first step, this solution gathers raw data from participants as a training data set and groups them into different grids by considering spatial correlation among sensor nodes. Then, the compressive sensing algorithm reconstructs the sensor data with spatial-temporal correlation and the low-rank feature among sensor node readings. The proposed solution guarantees high data quality in environments with high packet loss and noisy characteristics, but the study did not address energy savings.

A hybrid network coding and compressive sensing-based solution in [18] represents a clustered spatial-temporal correlation-based compressive sensing method. Each sensor node applies a well-defined Gaussian code matrix to encode its reading and forwards it to the base station. Proper selection of the measurement matrix and network coding coefficients is utilized at the base station to implement a low-complexity data reconstruction algorithm that reconstructs data more accurately. In addition, the proposed solution reduces the number of measurements, resulting in energy-efficient data transmission.

The authors of [19] propose a novel compressive sensing-based data-gathering solution for massive lossy data-transmission scenarios. The proposed compressive sensing solution models environmental data dynamicity using an environmental matrix (EM) and then employs the spatial-temporal correlation observed in this matrix to reconstruct the data. Furthermore, this approach has a multiple attribute assistant (MAA) component, which correlates environmental attributes to discover data loss patterns in reported encoded data. Combining MAA and EM, the compressive sensing-based reconstruction method provides better reconstruction accuracy. However, this approach suffers from higher computational complexity.

Another study [2] employs sequential data gathering and progressive data reconstruction to address a sliding window-based compressive sensing solution. Sensor nodes use this method to periodically encode and deliver their readings to the base station. The progressive data reconstruction method at the base station exploits Kronecker sparsification to recover the portion of sensor node readings reported during the last time period. This progressive method uses the recovered sensor readings from the last time period to restore encoded data in the current time period. There is a trade-off between data reconstruction accuracy and reconstruction complexity, which is adjusted by the sliding window size. In addition, the dynamic adjusting window size modifies the number of required measurements, which results in adaptive data transmission.

The authors in [20] address a two-step compressive sensing based data aggregation method in the clustered sensor network. Each cluster head utilizes a measurement matrix seed received from the base station to customize a localized measurement matrix and broadcast it into its cluster. Then, sensor nodes compress their measurements and forward it towards the cluster head. Upon receiving those readings, each cluster head aggregates and forwards their reading to the base station using an existing data routing tree among cluster heads. Simulation results show that the proposed solution improves

network performance regardless of intra-cluster or inter-cluster on the total energy consumption of network.

### 3. Preliminaries

#### 3.1. Compressive Sensing

This section presents the mathematical description of compressive sensing theory as presented in [21,22].

Let us assume that a discrete signal denoted by  $X \in R^N$  represented by  $N \times 1$  column vector has sparse representation in some basis (for example in a Fourier or wavelet basis). Considering the sparsity concept, this signal can be expressed in terms of its basis as follows:

$$X = \sum_{k=1}^N a_k \psi_k = \Psi a, \quad (1)$$

where  $\Psi$  is an  $N \times N$  orthogonal basis matrix  $\Psi = [\psi_1, \psi_2, \dots, \psi_N]$ ,  $\psi_i, i = 1, 2, \dots, N$  and  $a = [a_1, a_2, \dots, a_N]$  is an  $N \times 1$  column vector of the coefficient sequence of  $X$  in  $\Psi$  domain.

Signal  $X$  is compressible or sparse in the  $\Psi$  basis if most of the elements in  $a$  are zero. Compressive sensing theory states that if signal  $X$  is  $K$ -sparse on  $\Psi$  basis, it can capture and recover from  $M$  non-adaptive, linear measurements ( $K < M \ll N$ ) with a certain restriction.

Compressive sensing theory also states that, rather than acquiring the entire signal and then doing compression, it should be possible to construct a sampling framework to capture only the useful information in the sparse signal to begin with. The sampled signal via compressive sensing is described as follows:

$$Y = \Phi X, \quad (2)$$

where  $Y = [y_1, y_2, \dots, y_M]$  is the  $M \times 1$  measurement matrix,  $\Phi = [\phi_1, \phi_2, \dots, \phi_M]$  represents a  $M \times N$  and each  $\phi_i, i = 1, 2, \dots, M$  is  $N \times 1$  vector. It should be noted that  $\Phi$  is a random matrix, which can be considered a second basis. Each elementary  $y_i$  in the measurement matrix is a product of  $X$  and vector  $\phi_i$  from the sensing matrix. We can replace  $X$  with  $\Psi a$  and rewrite  $Y$  as:

$$Y = \Phi X = \Phi \Psi a = \Theta a, \quad (3)$$

where  $\Theta = \Phi \Psi$  is a  $M \times N$  matrix.

Compressive sensing demonstrates that sparse signals can be recovered from  $M$  measurements if the process can satisfy the restricted isometric property (RIP) [23,24]. The RIP states that the correlation between  $\Phi$  and  $\Psi$  must be low (be incoherent) to restrict the number of measurements required to recover the signal. Coherence is a metric to measure the maximum correlation between any row of  $\Phi$  and any column of  $\Psi$  and can be defined by the  $\mu$  as follows:

$$\mu(\Phi, \Psi) = \max_{1 \leq k, j \leq N} |\langle \phi_k, \psi_j \rangle|. \quad (4)$$

When comparing  $\Phi$  and  $\Psi$ , if the relationship between them is weak, the condition is called incoherence. Formally speaking, matrix  $\Theta$  of size  $M \times N$  satisfies the RIP of order  $K$  if it is a minimum number such that

$$(1 - \delta_k) \|a\|_2^2 \leq \|\Theta a\|_2^2 \leq (1 + \delta_k) \|a\|_2^2, \quad (5)$$

where  $\delta_k \in (0, 1)$  is a restricted isometric constant (RIC). Equation (3) must hold true for all values of  $a$  where  $\|a\|_0 \leq K$  and  $\|a\|_0$  is  $l_0$  norm, which shows that the number of non-zero elements in  $a$ . A  $l_p$  norm of vector  $a$  is defined as follows:

$$\|a\|_p^p = \sum_{i=1}^N |a_i|^p. \quad (6)$$

The RIP guarantees the exact recovery of  $X$  with a high probability if

$$M \geq CK \log \frac{N}{K}, \quad (7)$$

where  $N$  is the dimensionality of the original signal and  $C$  is the small positive constant. This formula is a conservative estimate of the lower bound on the number of measurements required to recover the original sparse signal. Since  $K$  represents the amount of important information inside the signal, we can conclude that the number of required samples to recover the signal is proportional to the information content of the signal.

The recovery of the signal  $X$  from  $Y$  is an NP hard problem, but it can be achieved through optimization. To do so, we can use  $l_0$  minimization as follows:

$$\hat{a} = \arg \min_{a \in R^N} \|a\|_{l_0} \quad s.t. \quad Y = \Phi X, \quad (8)$$

where  $\hat{X} = \Psi \hat{a}$ ,  $\hat{X}$  is the recovered signal and  $\hat{a}$  is the optimal estimation for  $a$ . Since  $l_0$  minimization is computationally intractable,  $l_1$  minimization is widely used for compressive sensing signal reconstruction. We can recover the coefficients of sparse signal  $a$  by solving  $l_1$  norm minimization as follows:

$$\hat{a} = \arg \min_{a \in R^N} \|a\|_{l_1} \quad s.t. \quad Y = \Phi X. \quad (9)$$

### 3.2. Distributed Compressive Sensing

Standard compressive sensing methods can be implemented on a single sensor node. However, sensor networks consist of many sensor nodes distributed over the network. These sensor nodes have spatial and temporal correlation among their readings. Given this distribution, distributed compressive sensing is proposed as a standard compressive sensing method for several sensor nodes [25,26]. In this distribution, each sensor node employs compressive sensing theory to encode its reading and transfer it to the base station. For the reconstruction procedure, the base station utilizes joint sparse signal recovery to recover all encoded signals precisely.

Joint sparse signal (JSM) recovery involves a set of signals that are jointly sparse. Each joint sparse signal can be represented by a combination of the common part and the innovative part [27]. There are three different joint sparsity models that can be used to recover the original signals. In this case, we have an ensemble of signals with notation and measurement models as follows:

Each signal is denoted by  $x_j, j \in \{1, 2, \dots, J\}$ , where  $x_j \in R^N$ . For this model, it is assumed that there is a known sparse basis  $\Psi$  for  $R^N$  in which each  $x_j$  can be sparsely represented. On the other hand, each signal  $x_j$  is assigned a measurement matrix  $\Phi$ , which is  $M_j \times N$ . Based on this definition, a measurement signal for each signal  $j$  can be represented by  $y_j = \Phi_j \times x_j$ .

In what follows, we introduce three different types of joint sparse models.

#### 3.2.1. Joint Sparse Model Type-1

The joint sparse model type-1, called JSM1, states that all signals reported by sensor nodes have a common sparse component while each individual signal contains an innovative sparse component:

$$x_j = z_c + z_j \quad \text{with} \quad z_c = \Psi a_c, \|a_c\|_0 = K \quad \text{and} \quad z_j = \Psi a_j, \|a_j\|_0 = K_j, \quad (10)$$

where the  $z_c$  is common to the all of values for  $x_j$  and its sparsity level is the minimum sparsity level of all the signals in base function  $\Psi$ . Signal  $z_j$  is the specific innovative part of  $x_j$  and has its sparsity level in the same base function. In this model, joint signal recovery maximizes reconstruction of the common part to increase the reconstruction accuracy. When the proportion of the common part is far more than the individual part, the reconstruction error decreases.

### 3.2.2. Joint Sparse Model Type-2

The Joint sparse model type-2, namely JSM2, assumes that signals have the same sparsity set profile; however, their coefficients are different. In this model, we formulate each signal as:

$$X_j = \Psi a_j, \quad (11)$$

where  $a_j$  is non-zero in the common coefficient profile. This models data for the sensor nodes that are deployed in the environment and observe the same signal, but, due to signal propagation effects, receive the signals with a phase shift.

### 3.2.3. Joint Sparse Model Type-3

There is some similarity between Joint sparse model type-3 (JSM3) and JSM1, but the common profile is no longer sparse. As with JSM1, we formulate the signal as follows:

$$x_j = z_c + z_j \quad \text{with} \quad z_c = \Psi a_c, \|a\|_0 = K \quad \text{and} \quad z_j = \Psi a_j, \|a_j\|_0 = K_j. \quad (12)$$

In this representation,  $Z_c$  is not sparse. This model is applicable for environments in which several signals are produced from different sources with a background signal that is not sparse.

## 3.3. Sparse Transform Basis

Compressive sensing must utilize an appropriate transform basis to provide an accurate sparse representation for a compressible signal. The selection of these transform bases is very important and relates to the nature of the data or the signal to be compressed. In this section, we propose a classification for signals and then introduce an appropriate transform basis for each of them.

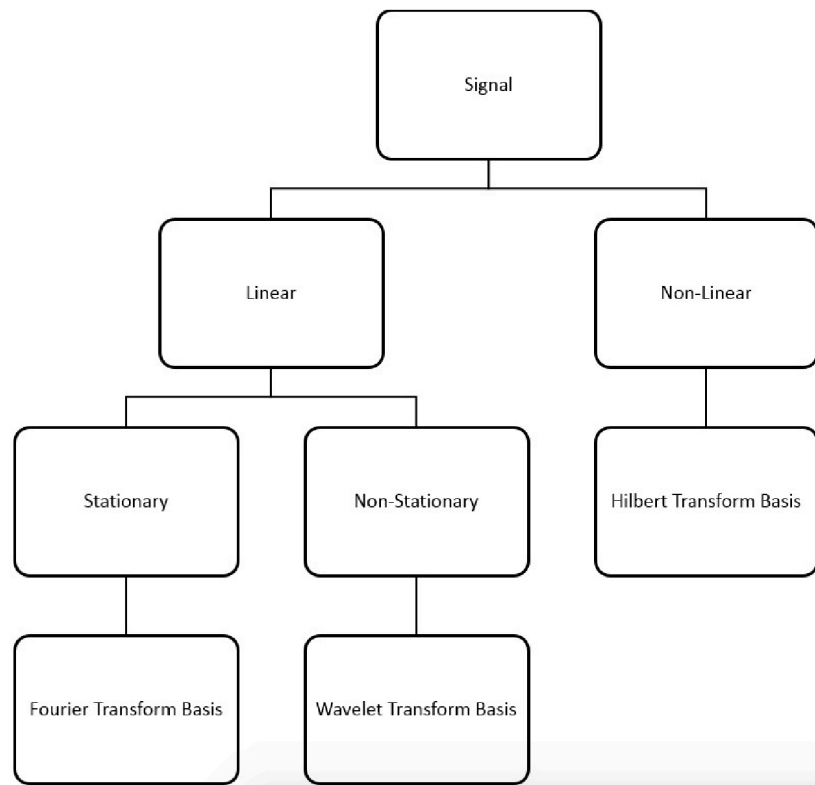
### 3.3.1. Classification

A signal is a physical quantity that is measurable. Various classifications of signals have been suggested based on the nature of the signal and the system that produces it. These include continuous versus discrete, periodic versus non-periodic, energy versus power, deterministic versus random, stationary versus non-stationary, and linear versus nonlinear [28,29]. In our study, we classify existing signals into a linear and nonlinear signals. Linear here means that the natural frequencies of the signal are not dependent on the amplitude of oscillation. A signal can also be stationary or non-stationary. A stationary signal is one whose frequency or spectral contents change over time.

The proposed signal classification and transform basis allocation are illustrated in Figure 1.

Based on the proposed signal classification, in this section, we introduce wavelet, Fourier and Hilbert transformations and discuss which transform basis belongs to which type of signal.

The Fourier transform is suitable for a wide range of purposes, such as signal transmission and stationary signal processing. However, this transform is not appropriate for non-stationary signals because it cannot provide any information about signal frequency changes as a function of time for these signals. To cope with this limitation, the short time Fourier transform (STFT) employs a fixed-size window to segment the signal into stationary parts but suffers from time or frequency resolution depending on its window size [28]. Most of the compressive sensing-based data-gathering approaches in wireless sensor networks use the Fast ourier transform (FFT) as their basis [30,31]. This basis implements transformation over the entire length of the signal. In signal reconstruction, this transform basis provides no information about time resolution and as such fails to report damage locations since all possible time information is lost. The STFT makes location information available. Although it provides a time-frequency representation of a signal, there is a major drawback with respect to utilizing STFT due to the fixed size of its window width [30,32].



**Figure 1.** Signal classification and appropriate transform allocation.

The Wavelet Transform (WT) is a time-frequency technique that overcomes the poor time/frequency resolution and fixed-size window limitations of the STFT. In contrast to the FT, the WT is well localized, and, with it, few coefficients are needed to represent local transient structures. Therefore, the wavelet basis provides sparse representation of piecewise regular signals, which may include transients and singularities [29,33].

The variable window size employed by WT allows it to implement time-frequency transformation. When dealing with low- and high-frequency signals, it employs large windows and short windows, respectively (corresponding to long and short time intervals). Therefore, the accurate location of a transient signal can be determined, as well as fundamental frequency and its low-order harmonics [28].

The WT can be either continuous or discrete. The Discrete Wavelet Transform (DWT) is used for signal decomposition, while a Continuous Wavelet Transform (CWT) is used for spectral analysis [34]. Decomposing the CWT produces wavelet series. These series are highly redundant and their computation is costly and time-consuming. The DWT is a fast computation of WT that reduces time and resource costs [34]. Wavelet Multi Resolution (WMR) is a method that implements DWT with filters. It employs a high-pass filter to process rapidly changing details (high-frequency components) and a low-pass filter to find slowly changing features (low-frequency components). In contrast with the WMR, the Wavelet Packet Transform (WPT) is a method that decomposes both wavelet details and wavelet approximation components at each level. Wavelet packets inherit properties such as orthonormality and time-frequency localization from their corresponding wavelet functions [29,33].

There are number of basis functions that can be utilized as a mother wavelet for the wavelet transform, including Haar wavelet transform, Symlet wavelet transform, Coiflet wavelet transform, and Meyer wavelet transform. In terms of the signal characteristics and the attributes of specific wavelet transforms, each of them can be employed to represent sparsity of signal. The WT's ability to provide good frequency resolution for high-frequency signals and time resolution for low-frequency



signals makes this technique a good basis for non-stationary signals that require time and frequency resolution [29,34].

Hilbert transform is another transform basis that takes a function as an input and produces a function with the same domain. The Hilbert transform can be either continuous or discrete. The nonlinearity and non-stationary nature of some signals make the Hilbert transform a powerful tool as the transform basis for signals with those properties. Hilbert transform decomposes a signal posteriori, enabling the extraction of the inner scales of each signal [29].

### 3.4. Belief Propagation

Belief propagation (BP) [35] is an iterative message passing algorithm that can calculate the marginal distribution or find estimates such as the most probable assignment (MAP) in Bayesian networks [36,37]. In the sparse signal recovery area, belief propagation is implemented on factor graphs and considered a fast decoder in Bayesian compressive sensing frameworks. This factor graph is a bipartite graph providing a visual representation of the sparse signal recovery procedure. It consists of two disjoint nodes: variable nodes and connection nodes connected through undirected edges whenever there is a dependency between these nodes. On the basis of belief propagation, these edges contain probability distribution functions on the variable nodes. There are only two directions for edges between variable and connection nodes [35]:

1. Edges or messages from a variable node to the connection node:

These edges contain probability which is calculated by gathering all incoming edges (excepts edges coming from node  $c_m$ ) and multiplying them, which is described as follows:

$$\mu_{x \rightarrow c}(x) = \prod_{x \in N(x) \setminus \{c\}} \mu_{v \rightarrow x}(x), \quad (13)$$

where  $\mu_{v \rightarrow x}$  shows the edges from node  $v$  to node  $x$ ,  $N(x)$  denotes the neighbor nodes of  $x$  and  $N(x) \setminus \{C\}$  denotes the neighbor nodes of  $x$  except for node  $C$ . For an example, see Figure 2, which shows part of a factor graph. The message from variable nodes  $X$  to the connection node  $c_m$  is given by:

$$\mu_{x \rightarrow c_m}(x) = \mu_{c_1 \rightarrow x}(x) \times \mu_{c_2 \rightarrow x}(x) \times \dots \times \mu_{(c_m-1) \rightarrow x}(x). \quad (14)$$

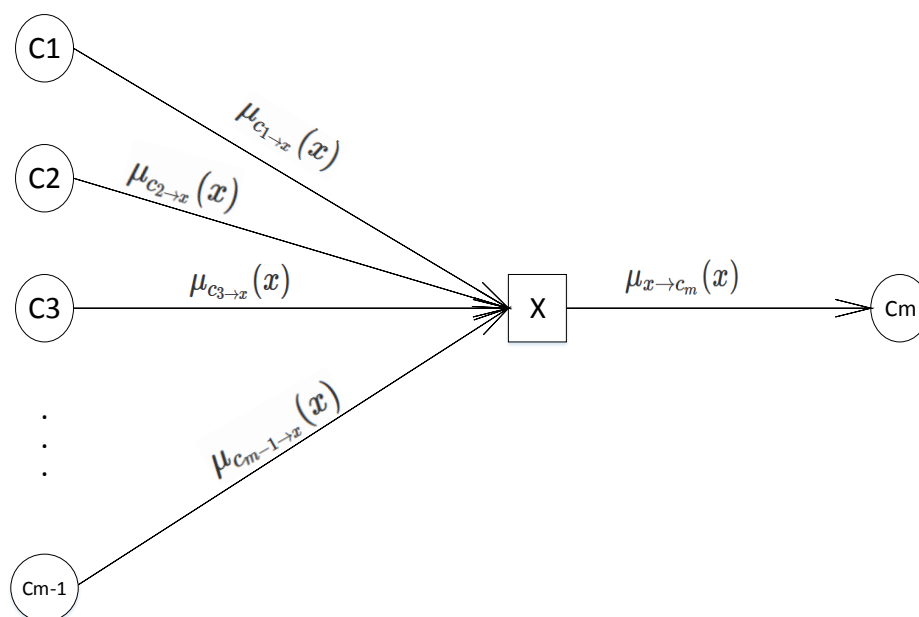


Figure 2. Message from variable node to the connection node.

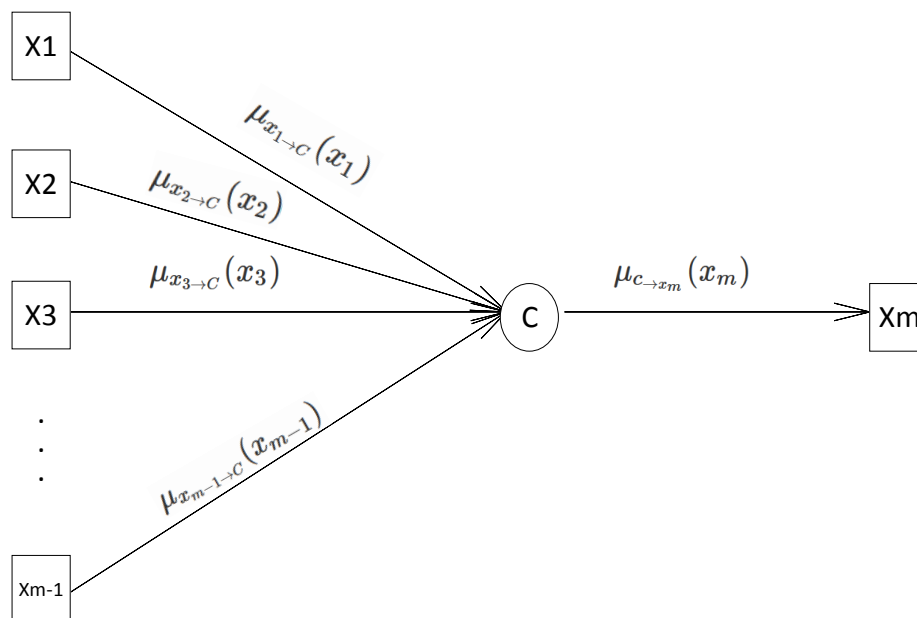
## 2. Edges from a connection node to the variable node:

This probability is computed by obtaining all the incoming edges to node  $C$  except the link from node  $x$ , multiplying them by  $C$  and finally finding the sum of all connected variable nodes except for node  $x$ . In general, the edges going from the connection nodes to the variable node can be described as follows:

$$\mu_{c \rightarrow x}(x) = \sum_{\sim \{x\}} c(N(c)) \prod_{v \in N(c) \setminus \{x\}} \mu_{v \rightarrow c}(v), \quad (15)$$

where  $N(C)$  is all variable nodes connected to  $C$  and  $\sigma$  is the sum of all connected variables except for  $x$ .  $\mu_{v \rightarrow c}(v)$  shows the edges from variable node  $v$  to the connection node  $C$ . An example is seen in Figure 3, which shows part of a factor graph. The message from connection nodes  $C$  to variable node  $x_m$  is represented by

$$\mu_{c \rightarrow x_m}(x_m) = \sum_{(x_1, x_2, \dots, x_{m-1})} c_{(x_1, x_2, \dots, x_{m-1})} \times \mu_{x_1 \rightarrow c}(x_1) \times \mu_{x_2 \rightarrow c}(x_2) \times \dots \times \mu_{x_{m-1} \rightarrow c}(x_{m-1}). \quad (16)$$



**Figure 3.** Message from connection node to variable node.

### 3.5. Network Model and Assumptions

Our heterogeneous network consists of  $N$  static sensor nodes regularly placed inside a region as shown in Figure 4. It utilizes multi-hop communication to forward the readings to the base station. Each sensor node is assigned an integer ID,  $n$ , which is in the range from 1 to  $N$ .

There are three types of sensor nodes in the network: normal nodes, coalition coordinator and base station. The normal sensor nodes have limited computational and storage capacity. Moreover, these sensor nodes are equipped with limited battery power, as recharging or replacing their batteries is difficult and expensive. There are a few powerful but also resource-restricted nodes: the coalition coordinators, which have more memory and computation capabilities. The base station is a powerful node with significant computational and storage resources. The network follows a periodic monitoring scenario to gather data from the environment. In such a long-term monitoring application, the end user requires a continuous stream of environmental data, although nothing specific

is being detected. In these sorts of applications, energy consumption and data accuracy requirements are more important than real-timeness.

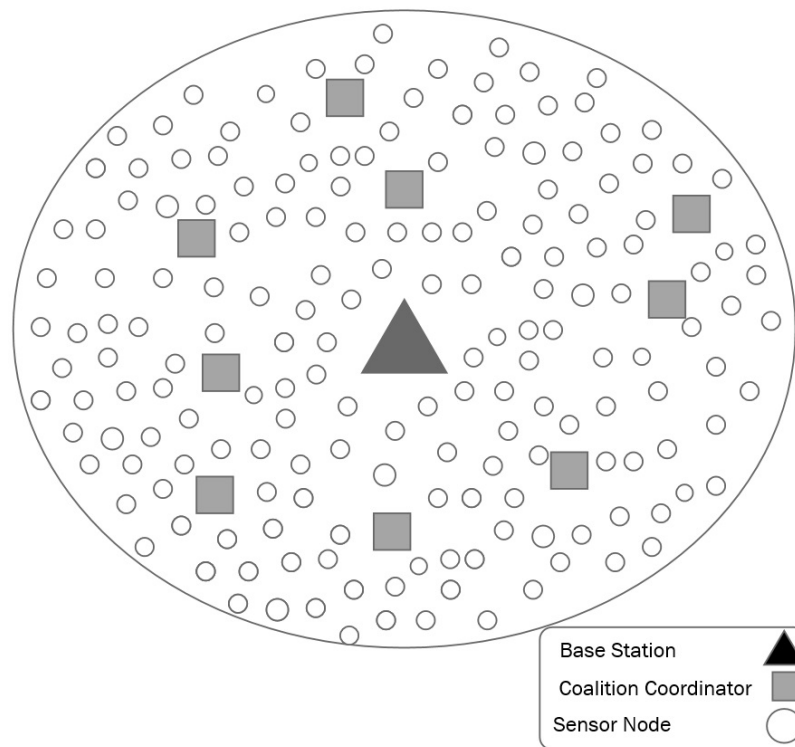


Figure 4. Network architecture.

The sensor nodes are grouped into coalitions. To form these coalitions, we consider the sparsity distribution of signals among sensor nodes and place all correlated sensor nodes in the same group (Figure 5).

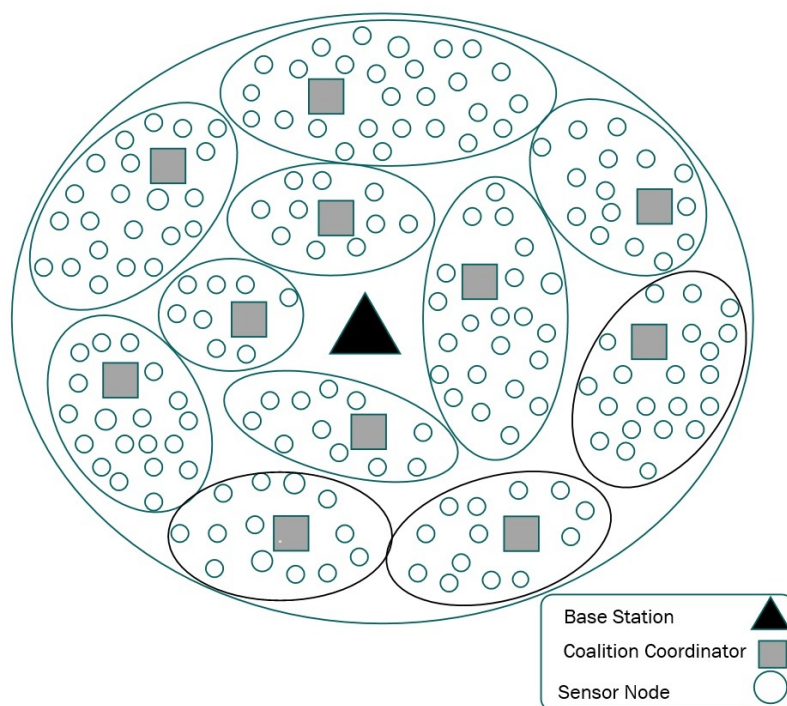


Figure 5. Coalition representation.

#### 4. Overview of the Proposed Approach

In order to achieve an energy-efficient and quality-aware compressive sensing method, we introduce a distributed compressive sensing approach with spatial correlation among sensor nodes to group them into coalitions. The proposed coalition-formation method is represented by a block diagonal measurement matrix in which each diagonal entity corresponds to one of the coalitions. Upon forming coalitions, the proposed spatial-temporal correlation-based compressive sensing approach is implemented inside each coalition in order to schedule sensor nodes and encode their readings. Temporal correlation among sensor node readings allows our approach to adjust the number of measurements with regard to the temporally changing sparsity level.

After applying the proposed compressive sensing solution inside each coalition and forwarding compressed data, the base station employs a two-step joint sparsity-based recovery algorithm to reconstruct the original signal. In the first step, it runs a joint sparsity model to find the common sparsity profile among the coalitions. In the second step, it calculates the common sparsity model within each coalition, which allows it to achieve data reconstruction with high accuracy and fewer measurements.

We summarize our contributions in this paper as follows:

- Proposing a new compressive sensing-based coalition-formation mechanism that includes distributed compressive sensing-based data gathering. This mechanism balances energy and quality of service.
- Introducing a sparsity distribution-based sensor node scheduling technique that minimizes the number of active sensor nodes.
- Proposing an adaptive measurement mechanism that adjusts the number of measurements inside each coalition according to the temporal sparsity level of the signal.
- Proposing a two-level belief propagation-based reconstruction algorithm that provides acceptable data quality in terms of data accuracy.

For a graphical view of the entire data-gathering procedure proposed in this paper, see the flowchart in Figure 6. This flowchart will be described in more detail in the following sections.

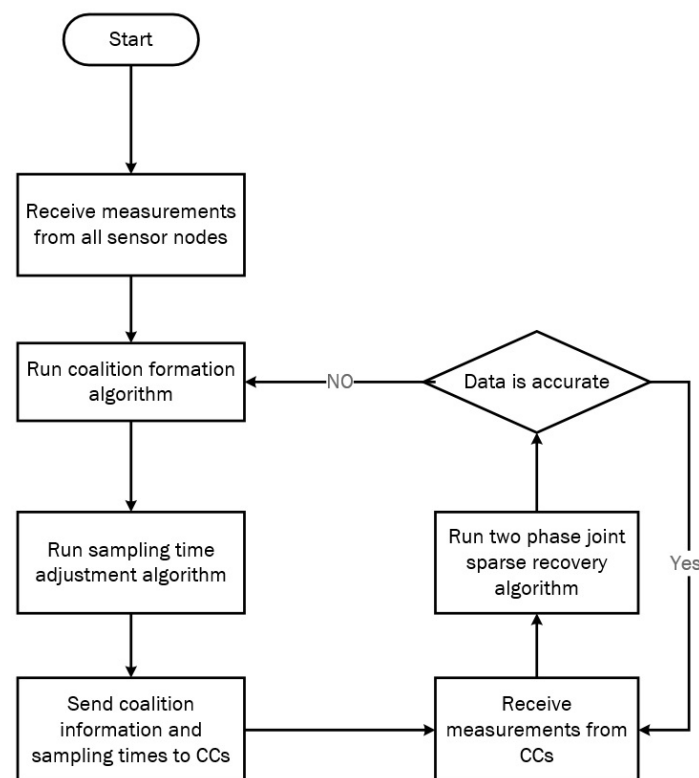


Figure 6. Approach overview flowchart.

## 5. Coalition Formation

We utilize spatial correlation among sensor nodes and a base function distribution pattern in order to group sensor nodes into several coalitions. Using coalitions leads to an energy-efficient compressive sensing-based measurement system that puts correlated sensor nodes close to each other in the same coalition. Moreover, coalitions make our measurement system sparse, which means each measurement is obtained through a linear combination of samples captured by a few sensors within a single coalition. Using this coalition structure, our approach decreases transmission cost. In addition, within each coalition, we remove the redundancy among compressed measurements of different sensors, decreasing the number of data packets.

The compressive sensing-based coalition formation introduced here is inspired by the fact that the subject signals are distributed in the environment. On the other hand, the distribution of the signal elements is different for the sensor nodes based on their locations. Therefore, the sensor nodes observe the same signal with different resolution. On the other hand, the sparse representation of the signal can be shown in a sparsity base function. We apply the base function distribution over the network to define a cover parameter. This cover metric shows the degree of the base function covered by the sensor nodes. This parameter helps sensor nodes inside each coalition produce informative measurements, which results in accurate data recovery by the base station.

To make a concrete and accurate coalition, we define a utility function  $U$  based on cover, transmission and sensor node correlations. This function is used to evaluate the efficiency of the coalitions produced. To do so,  $U$  makes a trade-off between reconstruction accuracy (correlation between sensor nodes and base function distribution pattern) and data-transmission cost (distance between sensor nodes).

The following sections explain the prerequisites for the coalition-formation algorithm and conclude by presenting its procedure.

For the sake of readability, we have summarized a list of notations used in the following sections in Table 1.

**Table 1.** Notations.

$N$	number of sensor nodes	$N_C$	number of coalitions
$X$	signal	$Y$	measurement
$J$	coalition index	$C$	coalition matrix
$\Phi$	basis function	$E_{Co}$	energy cost
$Dist(i, j)$	distance between nodes $i, j$	$CR$	correlation metric
$SCD$	sparsity cover degree	$Col_j$	coalition ID
$n_i$	node index	$l_{i,j}$	link ID
$CC$	coalition coordinator	$N_{ANode}$	number of active nodes
$SN$	set of sensor nodes	$ST$	set of sampling times
$M$	number of measurements	$K$	sparsity degree
$Comm$	number of common non-zero elements	$SL$	similarity level
$CSP$	common sparsity profile	$CR$	compression ratio
$SNR$	signal to noise ratio	$P_j$	number of active nodes in coalition $j$

### 5.1. Measurement Matrix

Since we divide the network into different coalitions, the data is gathered through these different coalitions. Therefore, whole data may be divided into discrete blocks, in which each block is acquired via a local measurement operator. Assume that we have divided the network into  $N_C$  coalitions and our signal  $X$  is partitioned into  $N_C$  blocks  $X_1, X_2, \dots, X_{N_C} \in R^N$ . Each block  $j$  which shows coalition  $j$  is assigned with a local measurement sub-matrix  $\Phi^j : R^N \rightarrow R^{N_j}$ . Each measurement sub-matrix  $\Phi^j$  represents the measurement pattern in each coalition  $j$ .

Each of our measurement sub-matrices  $\Phi^j$  is assigned to one coalition matrix  $C$ , which permutes the entities of the original signal assigned for the specific coalition  $j$ . In order to assign each coalition  $j$ ,

through its measurement matrix  $\Phi^j$  with the permuted signal coefficients  $x_j^C$ , we multiply  $C$  with  $X$ , which produces  $CX = X^C = [x_1^C, x_2^C, \dots, x_{N_C}^C]$  and finally we have:

$$Y = \Phi CX, \quad (17)$$

where:

$$\Phi = \begin{pmatrix} \Phi^1 & & & & \\ & \Phi^2 & & \mathbf{0} & \\ & & \cdot & & \\ & \mathbf{0} & & \cdot & \\ & & & & \Phi^{N_C} \end{pmatrix}. \quad (18)$$

The resulting matrix  $\Phi C$  represents the distribution of sensor nodes inside each coalition and  $C$  is a coalition matrix.

Based on Equation (18) and sparse representation of the signal  $X = \tilde{\Psi}a$ , we obtain:

$$Y = \Phi CX = \Phi(C\tilde{\Psi})a = \Phi\Psi a, \quad (19)$$

where  $\Psi = C\tilde{\Psi}$  is a permuted version of  $\tilde{\Psi}$ . As we have seen, our coalition matrix permutes the base functions. Through this permutation, each row in  $\Psi$  is a permuted row of  $\tilde{\Psi}$  as follows:

$$\psi_i^T = \sum_{j=1}^N P(i,j) \tilde{\psi}_j^T, \quad (20)$$

where  $\psi_i^T$  and  $\tilde{\psi}_j^T$  are the row vectors of  $\Psi$  and  $\tilde{\Psi}$ , respectively. If  $P(i,j) = 1$ , then the  $i^{\text{th}}$  row of  $\Psi$  is replaced with  $j^{\text{th}}$  row of  $\tilde{\Psi}$ .

After permutation with the coalition matrix, we have a proper pattern for each  $\Phi^j$  to measure data from the sensor node inside each coalition  $j$ .

The coalition-formation mechanism proposed here leads to a block diagonal measurement matrix with the appropriate coalition matrix  $C$  relative to the location of the sensor nodes.

## 5.2. Utility Function

The proposed coalition matrix  $C$  can be assigned to any coalition-formation method. During the coalition-formation procedure, we aim to make the best trade-off between energy savings and reconstruction accuracy. To do so, we introduce a utility function to evaluate the efficiency of the coalition-formation process. This utility function is defined based on the energy, correlation and cover degree parameters. The following sections explain these parameters in more detail and conclude by introducing the utility function.

### 5.2.1. Energy

We define the energy parameter for coalition-formation scenarios based on transmission, processing and measurement costs. Energy consumption is described as follows:

$$E_i = E_{comm} + E_{pro} + E_{meas}, \quad (21)$$

where  $E_{comm}$ ,  $E_{pro}$ ,  $E_{meas}$  are communication, processing and measurement energy parameters, respectively. In this section, we focus on data transmission and measurement cost. We normalize these costs and replace energy requirements for the processing and measurement with the measurement costs. The transmission cost relates directly to the distance, while the measurement cost is influenced by the number of measurements. We replace the energy parameter with the normalized parameters of

distance between two nodes inside the coalition and the number of measurements is represented as follows:

$$ECo(i, j) = \frac{Dist(i, j)}{Dist_{max}} + \frac{M_i}{M_{max}}, \quad (22)$$

where  $ECo_i$  is the energy cost in terms of the normalized distance between nodes  $i, j$   $Dist(i, j)$  and the normalized number of measurements taken by node  $i$  ( $M_i$ ).

### 5.2.2. Correlation Degree

Since sensor nodes are located close to each other and they sense the same signal with different resolutions, we are able to find different levels of spatial correlation among them. Existing compressive techniques usually ignore this correlation and transfer redundant compressed data, which costs energy. Our approach tries to remove the redundancy among compressed data, thus leveraging this spatial correlation. To do so, in the coalition-formation phase, we force the algorithm to consider this correlation along with other parameters. When we have correlated sensors in the same coalitions, we remove redundancy in the compressed data using the algorithm described in the next section. The correlation metric among sensor nodes is defined as  $Corr$ :

$$Corr(i, j) = \frac{Cov(y_i, y_j)}{\sigma(y_i)\sigma(y_j)}. \quad (23)$$

According to this formula, we define a binary variable  $CR$  which indicates whether two sensor nodes are sufficiently correlated. For this purpose, we rely on a user-defined correlation threshold  $TH1$ :

$$CR(i, j) = \begin{cases} 1 & \text{if } Corr(i, j) > TH1, \\ 0 & \text{if } Corr(i, j) \leq TH1. \end{cases}$$

### 5.2.3. Cover Degree

Measured signals can be represented by a sparsity function distribution over the networks. These functions can also be grouped into one or more coalitions. The performance of our coalition mechanism in terms of recovered data accuracy greatly depends on the nature of the sparsity base functions. To evaluate this performance, we define a sparsity base cover degree ( $SCD$ ) metric, which measures the degree of overlap between each coalition with base functions  $\Psi$ . Essentially, the  $SCD$  shows the energy overlap between the base functions and coalitions.

We define the  $SCD$  metric between each base function  $i$  and coalition  $Col_j$  as follows:

$$SCD(j, i) = \sum_{m \in Col_j} \psi^2(i, m), \quad (24)$$

where  $m$  is a sensor node located in coalition  $j$ . This metric  $SCD(j, i)$  indicates that measurements collected from coalition  $j$  contain information about the measurements of other coalitions that cover the same base function  $i$ . Taking this coverage degree among different coalitions into account, we use a joint sparse signal recovery approach to recover the original signal. However, there are situations in which  $\Psi$  is covered only by one coalition. This means that, for a  $K$ -sparse signal, the sparsity bases are contained in one coalition. Since this coalition is not known beforehand, we need to gather data from all coalitions, which is not energy efficient. Meanwhile, for accurate data recovery, we must have  $O(K \log N)$  measurements, meaning that we need  $O(KN_C \log N)$  measurements from all coalitions. On the other hand, the redundant measurements from other coalitions that do not overlap with the base do not contribute to improving data accuracy. If  $\Psi$  even has overlap among different coalitions,

our data recovery accuracy is increased. To quantify the cover level of  $\Psi$  over coalitions, we define the maximum  $SCD$  as follows:

$$SCD_{max}(\Psi) = SCD_{max}(C\bar{\Psi}) = \max_{j,u} \sum_w \Psi^j(w,u), \quad SCD(\Psi) \in [0,1]. \quad (25)$$

$SCD_{max}$  shows the maximum cover level of each coalition with the sparsity base, while  $\Psi^j$  is the base sub-matrix assigned to coalition  $j$ .

Forming coalitions such that the sparsity function can be recovered with several coalitions increases the quality of the recovered data. On the other hand, we must lower the number of correlated coalitions and the number of measurements to cut energy consumption. To find this balance, we must define a utility function as described in the next section. We discuss the number of coalitions and measurements and how  $SCD_{max}$  makes a trade-off between energy and quality in terms of the minimum number of measurements and data recovery accuracy.

#### 5.2.4. Utility Function Formulation

In this section, we calculate a utility function  $U$  to evaluate the candidate coalition structures.

In the coalition-formation phase, we aim to build  $N_C$  coalitions to achieve minimum energy consumption while meeting data-quality requirements. Minimizing transmission and measurement costs, which is the main energy consumption parameter here, depends on  $SCD_{max}$ ,  $M_i$  and  $Dist_{ij}$ .

The proposed utility function should evaluate adding a new sensor node to any coalition in terms of the trade-off between energy and quality. To do so, we define the utility function  $U$  for each combination of  $(n_i, Col_r)$  as follows:

$$U(n_i, Col_r) = CR(n_i) \times (ECo(n_i, Col_r) + \alpha SCD_{max}(n_i, Col_r)), \quad \alpha > 0, \quad (26)$$

where  $CR(n_i)$  defines the correlation level of sensor node  $i$  with other sensor nodes in  $Col_r$ . Regarding the  $SCD_{max}(n_i, Col_r)$  parameter, adding a sensor node to different coalitions may change the value of  $SCD_{max}$ .

#### 5.3. Coalition Formation Algorithm

In the model introduced in this paper, the network consists of  $N$  sensor nodes  $SN = \{n_1, n_2, \dots, n_N\}$  and  $L = \{l_{n_i, n_j}\}$  is the set of all possible connections between sensor nodes. Two sensor nodes are considered to be connected if they are placed in communication range of each other. We assume that  $\Psi$  is known and all sparsity bases are normalized to 1 so that  $SCD \in [0,1]$ . This algorithm aims to cut energy consumption, provided that the data-quality requirement is met. Adding a new sensor node to the coalition is a selection procedure that evaluates which coalition is the best candidate for adding a sensor node. To make a good selection, we run an optimization algorithm on  $U$ .

Before describing the optimization algorithm, the  $SCD$  parameter must be redefined. During coalition formation, when we add a new sensor node to the coalition,  $SCD$  evaluates the effect of adding this node by assigning a weight for the link between existing node in the coalition and the new sensor node. Our coalition-formation method, in order to examine the effect of adding a sensor node to each coalition, defines the  $SCD$  factor as  $SCD_{max}(l_{n_i, n_j}, Col_r)$  by considering the link,  $l_{n_i, n_j} \in L$  and a given coalition  $Col_r$ . This link does not connect two nodes in the same coalition, which means  $n_i \in Col_r$  and  $n_j \notin Col_r$ .

Now, we redefine the utility function  $U$  for each combination of  $(l_{n_i, n_j}, Col_r)$ :

$$U(n_i, Col_r) = CR(n_i, n_j) \times (ECo(l_{n_i, n_j}) + \alpha SCD_{max}(l_{n_i, n_j}, Col_r)). \quad \alpha > 0 \quad (27)$$

$SCD_{max}(l_{n_i, n_j}, Col_r)$  here represents the maximum cover level when a new node is added to coalition  $Col_r$  using the link  $l_{n_i, n_j}$ .



Now, we can formalize our optimization algorithm. The main goal in the improvement procedure is to lessen energy consumption and data recovery errors by adding new sensor nodes to the coalitions. When selecting an appropriate coalition for a new sensor node, the optimization algorithm examines the utility of adding the new sensor node to each coalition. For each coalition, this utility is defined based on the link connecting the new sensor node to an existing node in the coalition. Running the optimization algorithm finds the coalitions that pare down utility function's link cost. We formulate this optimization as follows:

$$(n_i, l_{min}, Col_{r_{min}}) = \arg \min U(n_i, Col_r) = \arg \min [CR(n_i, n_j) \times (ECo(l_{n_i, n_j}) + \alpha SCD_{max}(l_{n_i, n_j}, Col_r))] \quad (28)$$

*subject to*  $n_i \in SN, n_j \in Col_r, CR(n_i, n_j) \in \{0, 1\}$ .

Now let us describe the algorithm according to the optimization equations. To find a set of sensor nodes for each coalition, this algorithm uses Equation (28) to find a set of links such that the total  $U$  of the links is minimized. In the initialization step, we assume a set of candidate nodes and candidate links to be added to the coalitions defined by a set of  $SN$  and  $L$ . The algorithm then assigns each coalition coordinator node  $CC$  to one of the  $N_C$  coalitions. In addition, it defines  $SN_{Col_r}$  and  $L_{Col_r}$  as a set of sensor nodes and connections of coalition  $r$ , respectively. Along with the initialization step, the algorithm runs an iterative procedure where, in each iteration, it allocates one sensor node to one of the coalitions. To do so, it first finds the utility function for all possible connections defined in  $L$ . It then runs the optimization function defined in Equation (28) and finds the minimum utility. However, it finds the utility for sensor nodes that satisfies the minimum correlation requirements. The output of this optimization is a connection link with minimum utility  $l_{min}$ . This link connecting  $(n_i, n_j)$  adds sensor node  $n_i$  to the coalition of which node  $n_j$  is a member. Upon adding a new sensor node, the  $SCD$  parameter of the all links connected to the coalitions changes so that the link utility of the sensor nodes may change. It then removes this link and sensor node  $n_i$  from the list of candidate links and nodes. This procedure continues until all sensor nodes are assigned to the coalitions. Algorithm 1 represents this coalition formation procedure.

---

#### Algorithm 1 Coalition formation

---

```

1:  $SN = n_1, n_2, \dots, n_N$ 
2: Define  $L = l_{ij}$  as a set of all possible links
3: Define  $N_C$  coalitions with one coalition coordinator
4: Define set of nodes  $N_{col_k}$  and links for each coalitions  $L_{col_k}$ 
5: for ( $P = 1; P \leq (N - N_C); P++$ ) do
6:   for ( $Q = 1; Q \leq |L|; Q++$ ) do
7:     Find  $L_{poss} = l_{ij}; i \in N_{col_k}, j \in SN$ 
8:   end for
9:   Calculate  $U(l_{ij}, Col_k)$ 
10:  Add  $n_j$  to  $Col_{k_{min}}$ 
11:  Add  $l_{min}$  to  $L_{Col_{k_{min}}}$ 
12:  Find  $(n_j, l_{min}, Col_{k_{min}})$ 
13:  Remove  $n_j$  from  $SN$ 
14:  Remove  $l_{min}$  from  $L$ 
15: end for

```

---

## 6. Data Gathering inside Coalitions

For the sensor nodes inside each coalition, we use a distributed compressive sensing technique to take measurements and recover the original signal. Before describing the data-gathering approach, in this section, we take advantage of the  $SCD$  parameter to find the minimum number of active sensor nodes inside each coalition. The active sensor nodes inside the coalition send their reading through a multi-hop network structure to the coalition coordinator.

### 6.1. Number of Active Sensor Nodes

In this section, we identify the number of sensor nodes required to be active for each coalition. This requirement helps to implement scheduling among sensor nodes and put some of the sensor nodes into sleep mode. Since the number of active sensor nodes measuring the signal is in direct relation with the number of measurements, we use Equations (4) and (7) to find the number of active sensor nodes.

To do so, we first define a minimum threshold based on Equation (4) to measure the minimum data reconstruction quality. We then measure the probability of having coherence, as defined in Equation (4), higher than this threshold to make sure that the number of sensor nodes is enough to recover a compressed signal with high accuracy. This probability is defined and represented as follows:

If all measurement matrixes  $\phi_j$  are orthogonal and the sparsifying basis function  $\tilde{\Psi}$  and coalition-formation matrix  $C$  are known a priori, then  $\mu(\theta)$  is bounded as follows:

$$\Pr[\mu(\theta) \leq O\sqrt{SCD_{max} \frac{N_C}{N} \log N}] = 1 - O\left(\frac{1}{N}\right). \quad (29)$$

Considering Equations (7) and (29), we can define the maximum number of sensor nodes required for providing high quality data as follows:

$$N_{ANodes} = O(SCD_{max} K N_C \log^2 N). \quad (30)$$

Fine-tuning the coalition-formation matrix leads to a smaller  $SCD_{max}$ , which in turn produces a lower number of sensor nodes while still ensuring data quality. The number of active sensor nodes associated with each coalition is dependent on its  $SCD_{max}$  level. Coalitions, which cover more sparsity functions, are more informative. Therefore, we must gather more data from those coalitions.

The number of active sensor nodes for each coalition is represented as follows:

$$P_j = N_{Col_j} = \frac{SCD_{Col_j}}{SCD_{max}} N_{ANodes}. \quad (31)$$

### 6.2. Compressive Sensing Based Data Gathering

We now construct a block diagonal measurement matrix using the spatial-temporal correlation among sensor nodes.

Let  $SN_{Col_j} = \{1, 2, \dots, N_j\}$  denote the set of sensor nodes for the  $j^{th}$  coalition, where  $P_j$  of these sensor nodes are randomly scheduled to be active. In contrast to existing compressive sensing methods, we define a new structure for the measurement matrix that is compatible with our coalition-formation method. We use a temporal block diagonal measurement matrix,  $\Phi_t$ , to gather data. During each sampling instance, we gather spatial observations of all sensor nodes at time  $t$  and produce a discrete spatial signal  $X_t$  at this time. The temporal observation of all active sensor nodes together is a spatial-temporal signal  $[X_1^{tr}, X_2^{tr}, \dots, X_{ST}^{tr}]$ , where  $ST$  is a parameter representing the number of samples in each sampling round  $T$ . Each sampling period consists of  $T$  sampling instances equal to the Shannon–Nyquist rate. To reduce this number of sampling times, the base station adjusts the number of sampling times according to the signal sparsity level and defines a number of sampling points  $ST$  for each sampling period.

For each sampling period  $t$ , we consider  $\Phi_t$  as a measurement matrix;  $\Phi_t$  is a  $P_j \times SN_{Col_j}$  matrix. The measurement vector  $Y_{Col_j}$  consists of  $ST_j$  sub-vector of  $ST$  sampling times such that  $Y^{Col_j} = [Y_1^{tr}, Y_2^{tr}, \dots, Y_{ST_j}^{tr}]$  where each  $Y_i$  is a  $P_j \times 1$  vector.

Inside each coalition, we utilize a block diagram measurement matrix that compactly represents several temporal measurement sub-matrices. Combining these spatial-temporal measurements together yields:

$$Y^j = Y_{Col_j} = \Phi^j X^j, \quad (32)$$

$$Y^j = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_{ST_j} \end{bmatrix} = \begin{pmatrix} \Phi_1 & & & \\ & \Phi_2 & & \mathbf{0} \\ & & \ddots & \\ & \mathbf{0} & & \Phi_{ST_j} \end{pmatrix} \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_{ST_j} \end{bmatrix}, \quad (33)$$

where for each  $1 \leq t \leq ST$ ,  $\Phi_t$  has  $P_j$  rows and  $SN_{Col_j}$  columns. At the end of each sampling round, every sensor node transfers its measurement vector  $Y_j$  to its neighbor node. The neighbor node receives the data and forwards it to the coalition coordinator.

Introducing this temporal block diagonal measurement inside each coalition provides energy-balanced data aggregation within the coalitions. For instance, consider the sample coalition in Figure 7, where active sensor nodes forward their compressed data through active neighbor nodes to the coalition coordinator. Existing compressive sensing-based data collection approaches introduce unbalanced data transmission, which leads to communication and data overhead in the network. In our approach, each sensor node transfers a total of  $M$  number of measurements where  $M_j = m_1 + m_2 + \dots + M_{SR_j}$ . For each sampling time  $t$ , sensor node  $i$  sends its measurement  $y_i, t$  to the next node  $j$ .

For each time  $t$ , we represent the data-gathering process as follows:

$$y^t = \Phi^t x^t, \quad (34)$$

where  $\Phi^t = [\phi_1, \phi_2, \dots, \phi_{N_{Col_k}}]$  is the measurement matrix of size  $P_{Col_k} \times N_{Col_k}$  and  $y^t$  is the compressed measurements of signal  $x^t$ .

Each element  $y_i$  of measurement vector  $y^t$  is a linear combination of all sensor node readings inside the coalition, which is weighted with a row of the measurement matrix. This representation of each measurement gathered from active sensor nodes at time  $t$  can be represented as follows:

$$y_i^t = \sum_{j=1}^{N_{Col_k}} \phi_{ij}^t x_j^t, \quad i \in 1, 2, \dots, P_{Col_k}. \quad (35)$$

Using this representation, we balance energy usage through an equal number of transmissions among sensor nodes inside each coalition. According to the proposed temporal representation, each sensor node is able to add its measurements  $\{\phi_{ij}^t\}_{i=1}^{P_{Col_k}} x_j^t = \phi_j^t x_j^t$  to the measurements of other sensor nodes separately to produce each measurement. To do so, each  $P_{Col_j}$  active sensor node inside the coalition locally multiplies its readings by measurement vector  $\phi_j^t$  and produces its measurement vector  $M^t$ . It then waits until it receives measurement vectors from other sensor nodes that forward their data to the coalition coordinator (CC) through this sensor node. Upon receiving them, this sensor node aggregates its measurement vector with the received measurement vector and forwards it to the CC. This procedure is depicted in Figure 7.

One question not yet answered concerns the means of routing sensor node measurements to the coalition coordinator. In the next section, we describe how sensor nodes form a routing tree and forward their measurements.

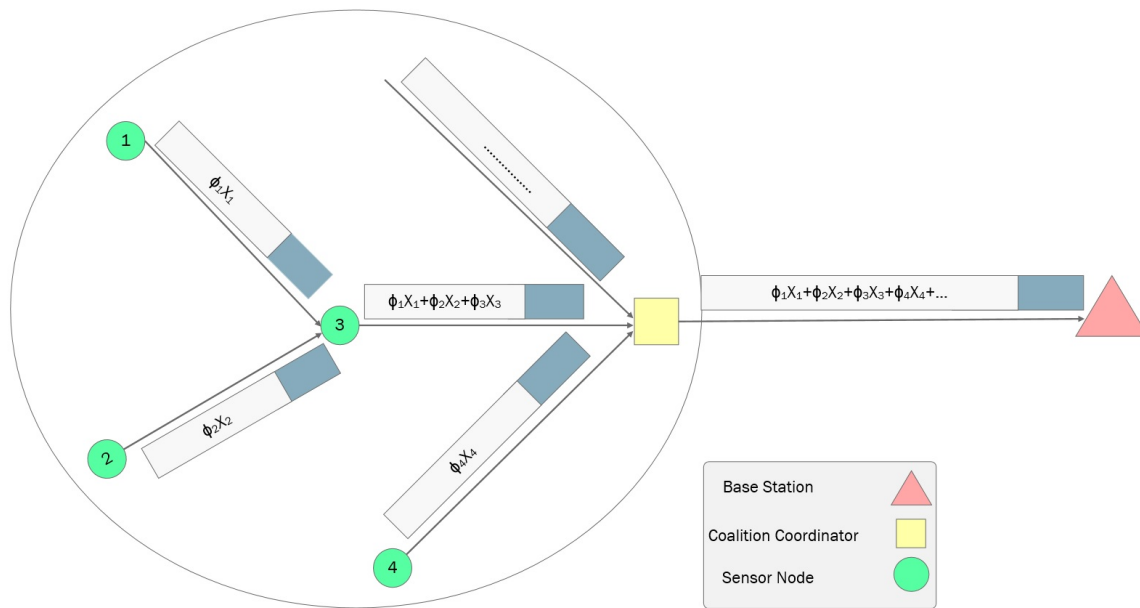


Figure 7. An example gathering scenario.

### 6.3. Data Gathering Trees

Inside each coalition, our network consists of  $P_j$  active nodes connected through a set of links  $L$ . For each time stamp, every sensor node reads environmental data and transmits it to the coalition centre. Since not all sensor nodes have access to the coalition centre, they forward their readings with multi-hop communication. In the compressive sensing technique, each coalition centre requires gathering only  $m$  projections from these sensors instead of all readings from sensor nodes where  $m \ll n$ .

As indicated above, each measurement is the weighted sum of active sensor node readings with non-zero coefficients. Our temporal measurement matrix consists of  $m_j$  rows and  $P_j$  columns where each column is assigned for one active sensor node and each row shows one of the projections. Since each row shows the selected sensor nodes whose readings contributed to the weighted sum  $m_j$ , we can consider each row as a data-gathering tree. In this regard, we define one of the sensor nodes with direct access to the coalition centre as a root of this tree. Therefore, having  $m_j$  projection, we obtain  $m_j$  data-gathering tree. To distribute the non-zero coefficients more evenly in matrix  $\phi$  and make each projection as sparse as possible, the number of non-zeros in each row of matrix  $\phi$  is set at  $\left\lceil \frac{m_j}{P_j} \right\rceil$  such that none of the columns in  $\phi$  has all-zero entries.

We now try to find  $m_j$  data-gathering trees, each of which corresponds to one weighted sum of selected sensor nodes such that it minimizes the energy consumption by minimizing communication links. The following section describes how each projection node builds its measurement tree.

#### 6.3.1. Building Data Gathering Tree

Selecting  $m_j$  projection nodes and assigning a data-gathering tree based on the optimized random basis matrix are the main objectives of this section. To select  $m_j$  projection nodes, the coalition coordinator randomly selects  $m_j$  nodes that are one hop from it, and fixes them as roots of  $m$  projection trees. Later, during the data-gathering procedure, the coalition coordinator can change the role of projection nodes and notify these nodes by sending update messages.

Each projection node initializes one of  $m_j$  data projection procedures using the proposed algorithm. As mentioned above, to build the data-gathering tree for each projection, we construct an optimized basis matrix and assign each row of this matrix to one of the projection trees. Each column of this projection matrix is assigned for one of the active sensor nodes, while each row is assigned

to one projection node. Non-zero coefficients in each row represent the nodes that participate in that projection.

To make each random projection as sparse as possible, this algorithm sets the number of non-zero entities in each row of the measurement matrix  $\phi$  as equal to  $\frac{m_j}{P_j}$ , on the condition that none of the columns in  $\phi$  has all-zero entries. The proposed measurement matrix has been generated in advance and stored in each sensor node. Now, we demonstrate building data projection trees within each coalition.

In each coalition, the coalition coordinator gathers  $m_j$  weighted measurements from all active sensor nodes inside the coalition. Each of these  $m_j$  measurements is collected by one of the measurement nodes. To do so, each row of our measurement matrix represents the data-gathering tree assigned to one measurement node and each binary entity in this row vector shows whether this node is participating in the current measurement. Each measurement node uses this measurement matrix to detect sensor nodes that will participate in the measurement. It then builds a data-gathering tree based on the minimum spanning tree algorithm to develop a route between participating nodes and the measurement node.

The measurement node initiates the data-gathering tree by introducing itself as a root. It then finds all participating nodes located within one hop and adds them to the tree. For other nodes, it finds the closest node and follows the shortest path method to connect that node to the tree. This procedure repeats until all participating nodes join the tree. When the tree is complete, each sensor node takes its sample  $x_i$ , multiplies it by its coefficients, adds it to the data received from its child nodes and finally transmits the weighted sum to its parent node. At the end of this procedure, the measurement node receives all the weighted measurements from the tree and forwards them to the base station. Since the measurement matrix consists of  $m_j$  rows, we will have  $m_j$  trees, each of which gathers one of the measurements through the assigned measurement node.

Upon reading a sample from the environment, each sensor node in each tree multiplies its reading by its coefficient, adds it to the data received from its child nodes and sends the weighted sum to the parent node. Repetition of this procedure in the tree terminates when the measurement node receives weighted sums from the child nodes and then sends these readings to the coalition coordinator. In this way, all sensor nodes transmit only one packet to the coalition coordinator. Consequently, for  $m$  data-gathering trees, the network transmits  $m$  packets.

## 7. Joint Sparse Signal Recovery Procedure

The base station is responsible for recovering the original signals from the measurements received from the coalitions. Considering several coalitions, signals gathered by the coalition coordinators are correlated in spatial and temporal domains. In essence, we have two types of correlations: spatial-temporal correlation inside each coalition and spatial correlation among coalitions. The distributed sparse signal approach proposed here engages these correlations and suggests a two-step joint sparse signal recovery process. In the joint sparse signal recovery concept [38,39], this correlation can be defined based on location or amplitude of non-zero coefficients of the signal. In our approach, correlation among different coalitions or sensor nodes is depicted as a similarity level and is defined based on the location of the non-zero coefficients as follows:

$$SL = \frac{Com}{K}, \quad (36)$$

where  $Com$  and  $K$ , respectively, are the total number of common non-zero coefficient locations among different coalitions or sensor nodes and the total number of non-zero coefficients of the signals (sparsity level).

At the end of each data reconstruction procedure, the base station calculates and finds the common sparsity profile  $CSP_{Col}$  among different coalitions. Coalitions that satisfy minimum similarity

requirements are considered in the joint sparse signal recovery process; otherwise, their recovery proceeds separately.

Based on this  $SL$  definition, in this section, we develop a two-step joint sparse signal recovery algorithm which uses the spatial-temporal prior information to reconstruct the original signal. Using this algorithm, our approach decreases the number of measurements while increasing the reconstruction accuracy.

Our signal recovery procedure is performed both inside the coalitions and among the coalitions. In the first step, the base station runs the joint sparse signal recovery algorithm among coalitions and finds their common sparsity profile ( $CSP_{Col}$ ). In the second step, using  $CSP_{Col}$  as an input for the reconstruction algorithm, it runs joint sparse recovery inside each coalition to complete the common sparsity profile of each coalition ( $CSP_{Col_j}$ ). Since the sensor nodes inside each coalition are more highly correlated, running the second round yields other parameters of the common sparsity profile. Upon finding the common profile among sensor nodes ( $CSP_{Col_j}$ ), the base station runs the individual recovery procedure to find individual parts. For the whole recovery process, the base station exploits the belief-propagation-based recovery algorithm introduced in our previous work [40]. Algorithm 2 summarizes the proposed recovery procedure.

---

**Algorithm 2** Two-step joint signal recovery

---

- 1: Find  $SL$  among different coalitions
  - 2: Find  $SL_{min}$
  - 3: Run spatial-temporal joint recovery inside coalitions with  $SL_{min}$
  - 4: **for** ( $J = 1; J \leq N_{Col}; J++$ ) **do**
  - 5:   **if**  $SL_{Col_j} > \theta$  **then**
  - 6:     Use the common sparsity profile from coalition with  $SL_{min}$  as input
  - 7:   **end if**
  - 8:   Run spatial-temporal joint recovery algorithm
  - 9: **end for**
- 

## 8. Performance Evaluation

In order to evaluate our solution, we first analyse the impact of different transform basis on compressive sensing. Then, we compare the distributed compression nature with centralized compressive sensing methods. We then compare our solution with other compressive sensing methods.

### 8.1. Assumption

For simulation, we have three different kinds of nodes: base station, coalition coordinator and sensor nodes. The base station groups these sensor nodes into different coalitions and assigns one coalition coordinator for each coalition. The coalition coordinator is responsible for transferring the coalition information to the base station. The base station is responsible for forming coalitions and running the optimal scheduling algorithm, which schedules communication in the network. Thus, there are no collisions in the network within or between coalitions. We transfer data by multi-hop communication, and each sensor node knows its geographical location and its connectivity radius. Two nodes are connected if and only if they are located in the communication range of each other. Coalition coordinator nodes are powerful nodes that act as aggregation centres.

We use a real dataset collected by Lausanne Urban Canopy Experiment (LUCE) wireless sensor network deployment at EPFL LUC [41]. This dataset gathers real temperature data using 64 ambient temperature sensor nodes. The sensor type is SHT75 sensor Sensirion. This dataset is sparse, which made it a very interesting dataset to evaluate recent compressive sensing solutions [42–45].

Our simulation is implemented in a MATLAB R2018a and we have used the toolbox provided for Bayesian compressive sensing in [38].

### 8.2. Impact Analysis of Transform Basis on Compressive Sensing Performance

We analyse the effect of HT, WT (including Haar WT, Symlet WT, Coiflet WT, and Daubichi WT), and FT on the signal reconstruction accuracy. In order to compare the signal reconstruction accuracy, we use two metrics: SNR and normalized root mean square error (NRMSE). Figure 8 illustrates the comparison between HT, WT and FT in terms of SNR.

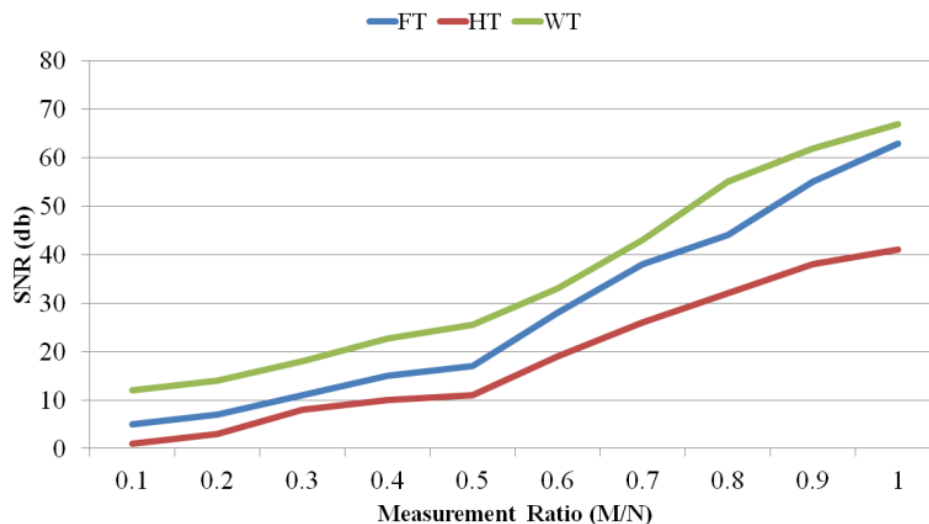


Figure 8. Comparing HT, WT, and FT in terms of SNR.

As the figure shows, WT performs the best. Since WT is able to provide time and frequency information about the signal, it provides better sparsity than the other transform bases. Comparing them in terms of signal reconstruction error, as illustrated in Figure 9, shows that HT and WT have the worst and the best accuracy, respectively.

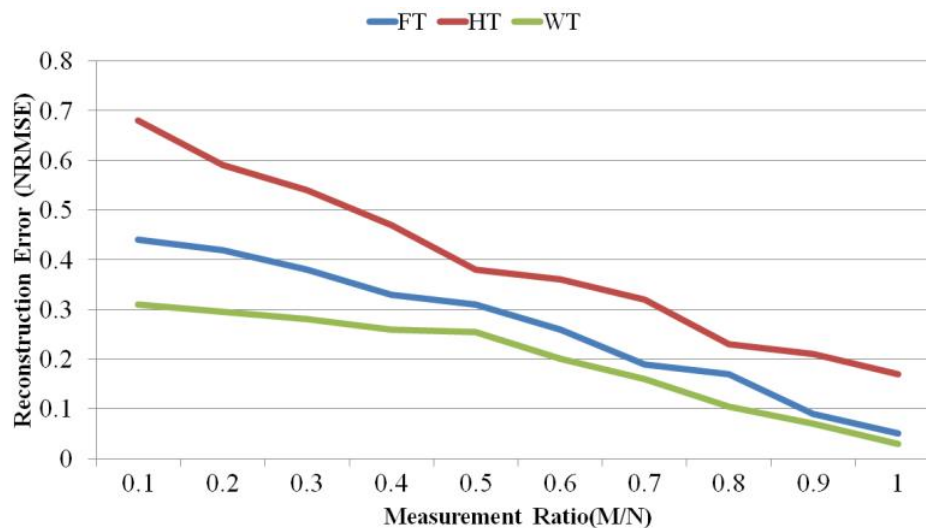
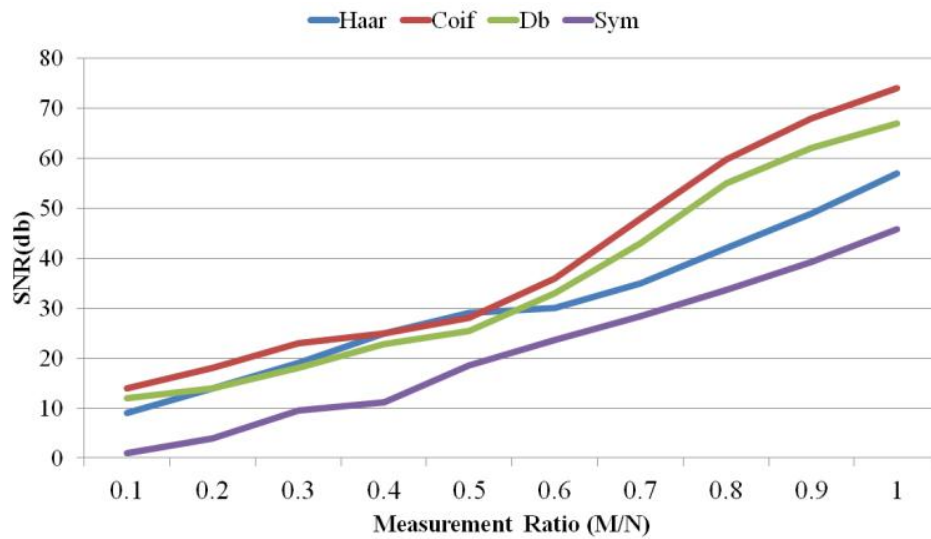
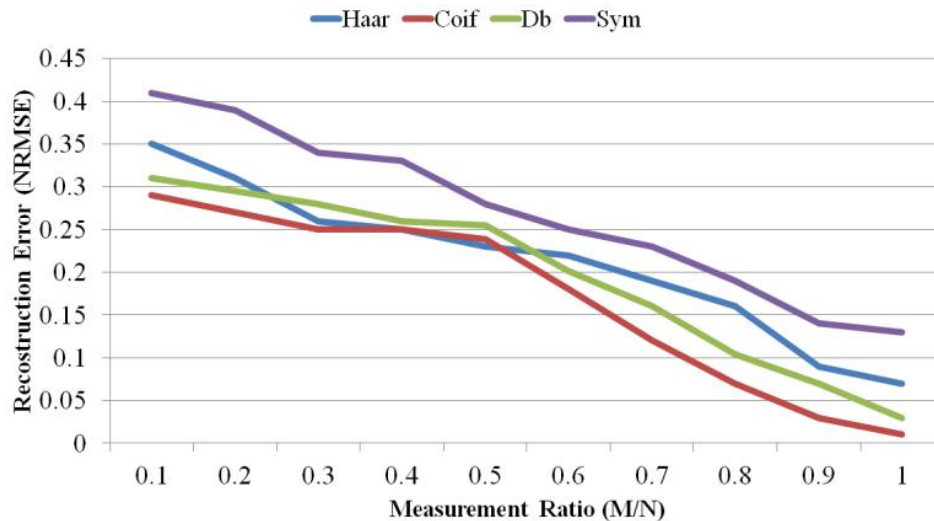


Figure 9. Comparing HT, WT, and FT in terms of signal reconstruction error.

Because WT shows the best performance, we further analyze a number of transform sub-types. Figures 10 and 11 illustrates the comparison between Haar WT, Symlet WT, Coiflet WT, and Daubichi WT. As we see, most of the time, Coiflet WT performs the best in terms of signal reconstruction error (measured as NRMSE) and SNR. Symlet WT performs the worst. However, when the measurement ratio is between 0.4 and 0.5, Haar WT shows better performance.



**Figure 10.** Comparing Haar WT, Symlet WT, Coiflet WT, and Daubichi WT in terms of SNR.



**Figure 11.** Comparing Haar, Symlet, Coiflet, and Daubichi transforms in terms of reconstruction error.

### 8.3. Comparing Distributed and Individual Compressive Sensing

The first part of the performance evaluation section investigates and compares distributed compressive sensing and individual compressive sensing approaches.

Typical signals measured by wireless sensor networks are sparse signals consisting of a common sparse component and different innovation components. Reconstructing these sorts of signals using joint sparse signal recovery models is more efficient than individual signal recovery models. It ultimately reduces the energy consumption and prolongs network lifetime.

For this simulation, we change the number of common and innovative parts. As depicted in Figure 12, distributed compressive sensing provides accurate reconstruction with fewer measurements. On the other hand, increasing the number of common sparsity parameters provides better reconstruction accuracy. When the number of the common sparsity level is minimized, its performance approaches that of a separate compressive sensing method.



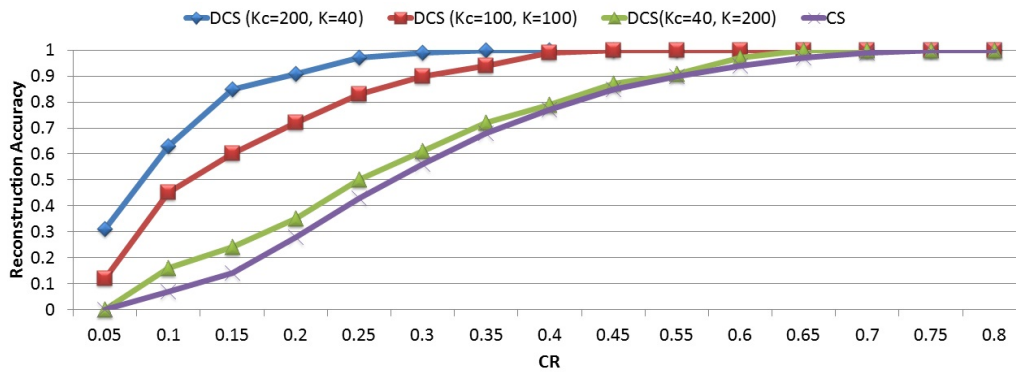


Figure 12. Reconstruction accuracy in terms of different common sparsity levels.

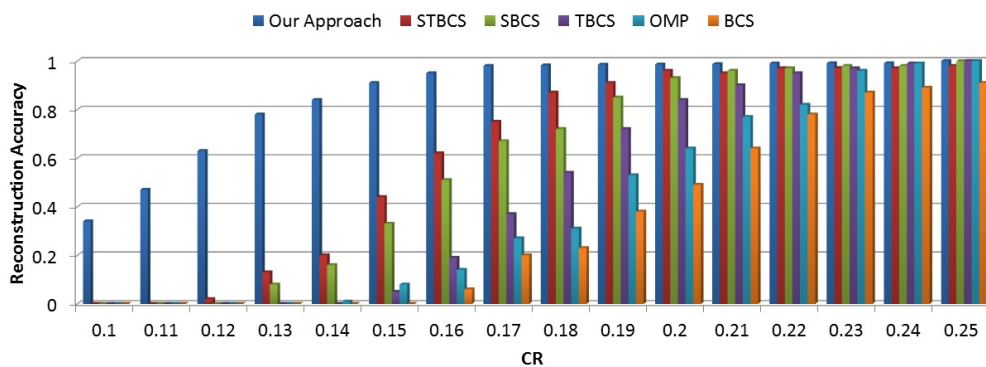
#### 8.4. Comparing with Other Compressive Sensing Based Methods

In this section, we compare the proposed spatial-temporal compressive sensing approach with the existing data compression techniques. The approaches used in the comparison are Bayesian compressive sensing (BCS) [46], clustered spatio-temporal Bayesian compressive sensing (STBCS) [47], temporal belief-propagation based compressive sensing (TBCS) [6], orthogonal matching pursuit (OMP) based compressive sensing [48] and spatial Bayesian compressive sensing (SBCS) [49].

##### 8.4.1. Data Accuracy

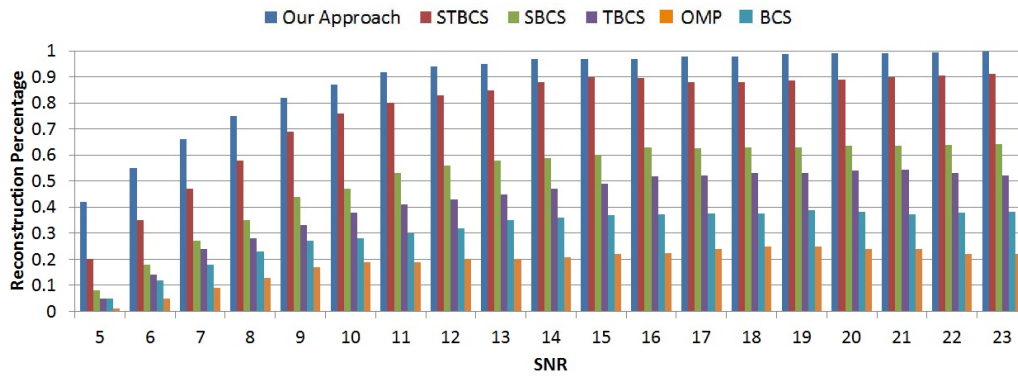
In order to measure the data accuracy, we first compare the accurate reconstruction percentage for the different data compression values. As Figure 13a indicates, our approach provides accurate data reconstruction with a minimum number of measurements. The main reason for this performance is the value of information gained using the coalition-formation method. Our approach tries to group sensor nodes based on their sparsity similarity, which leads to transmitting more information with a lower number of measurements. However, increasing the compression ratio decreases the gap between our approach and other methods. On the other hand, our approach adjusts the number of measurements based on the signal sparsity level, which removes redundant data transmission. As seen in Figure 13a, for more compression ratios, the reconstruction accuracy is almost the same.

An important parameter in the reconstruction procedure is the level of SNR. Figure 13b represents the reconstruction accuracy based on the SNR parameter. For lower SNR values, our approach represents better reconstruction accuracy. When increasing the SNR value, the STBCS model provides a similar level of accuracy but still lower than in our method. At SNR levels higher than 19, our method produces almost perfect reconstruction. Among other methods, the STBCS technique is the only one which shows comparable performance.



(a)

Figure 13. Cont.



(b)

Figure 13. (a) Reconstruction accuracy percentage for different compression ratios. (b) Reconstruction accuracy percentage for different SNRs.

8.4.2. Number of Transmission

One of the most important parameters affecting energy consumption is the number of data transmissions. The number of sensor nodes participating in data transmission is used to measure the number of transmissions. Figure 14 represents the number of transmissions for different compression techniques by changing number of sensor nodes.

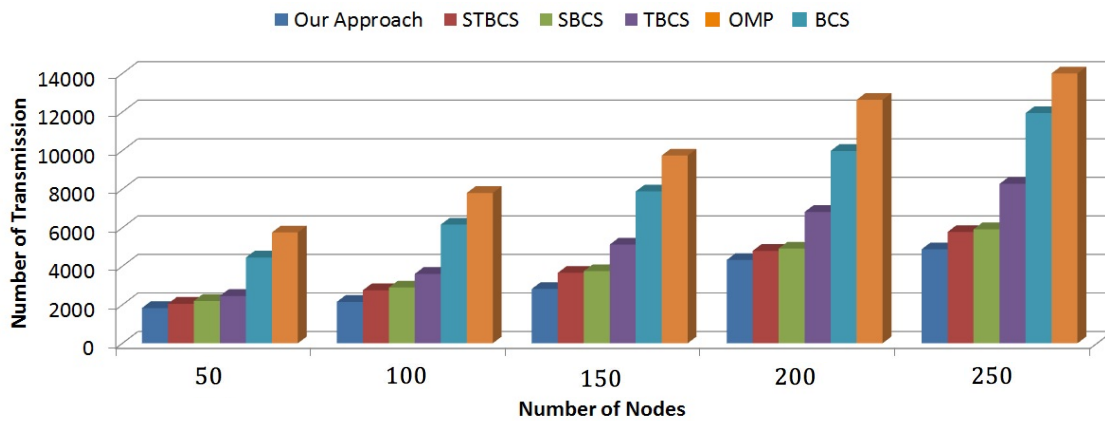


Figure 14. Number of data transmission versus number of sensor nodes.

For a network with a lower number of sensors, our approach has a minimum number of transmissions, but this number is comparable with STBCS and SBCS approaches. As the number of sensor nodes increases, the difference in the number of transmissions becomes more significant. This graph also demonstrates the scalability of our method. Our approach localizes sensor nodes based on the similarity of their sparsity levels in separate coalitions. For each coalition, it then curtails the number of transmissions by lessening the number of active sensor nodes participating in data transmission and adjusting the number of measurements to the sparsity level. Therefore, our approach involves a minimum number of transmissions.

8.4.3. Energy Consumption

In this section, we compare energy consumption parameters among the different methods. As we can see in Figure 15, our approach provides the best energy resource consumption performance among the approaches compared. Our approach adapts both the number of measurements and the number of active sensor nodes to minimize the energy expenditure. The STBCS and SBCS are the second and third best approaches, which provide performance somewhat more comparable to our method than

the other methods do. The main common feature of these three approaches is grouping sensor nodes and localizing sensor node measurements.

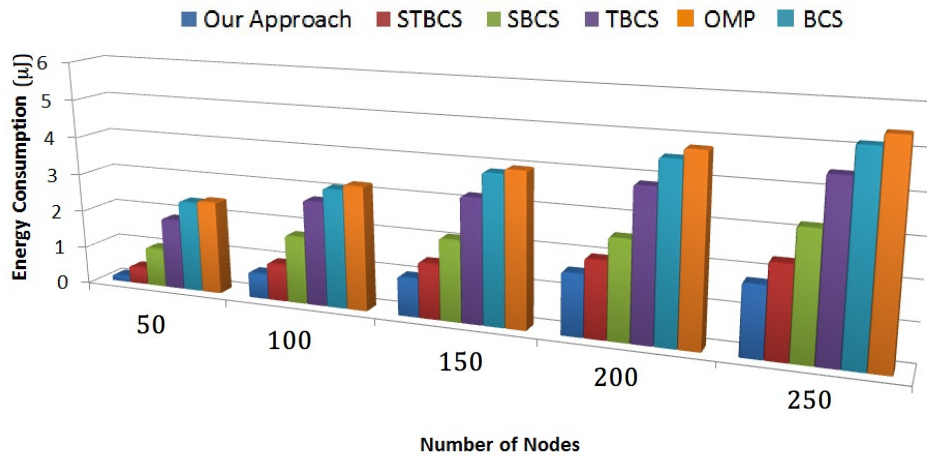


Figure 15. Energy consumption.

The means of grouping sensor nodes and gathering measurements from each group are the main difference among these methods. Since our approach groups the sensor nodes with reference to sparsity similarity, the number of active nodes is diminished. Moreover, it adjusts the number of measurements to the sparsity level, which removes the redundancy in data gathering. STBCS uses a clustering approach to group sensor nodes, but its grouping mechanism is not as accurate as that of our method. In addition, it does not consider adaptability to the number of measurements. Moreover, TBCS, BCS and OMP do not include localization of data gathering, which leads to greater exhaustion of energy.

#### 8.4.4. Trade-Off between Energy and Accuracy

We study the trade-off between network energy usage and data accuracy. To do so, we investigate energy consumption in terms of normalized network lifetime, while for accuracy we consider the error between the reconstructed data and real data.

As we can see in Figure 16, our approach provides approximately 40% network lifetime at a minimum error level, while with less accurate data it provides higher percentages of lifetime. Adapting both the number of measurements and the number of active sensor nodes to the signal sparsity level is the main contribution of our approach to prolong network lifetime. In addition, spatial-temporal correlation-based coalition formation provides sufficient similarity among sensor node measurements, which leads to less error in the recovered data.

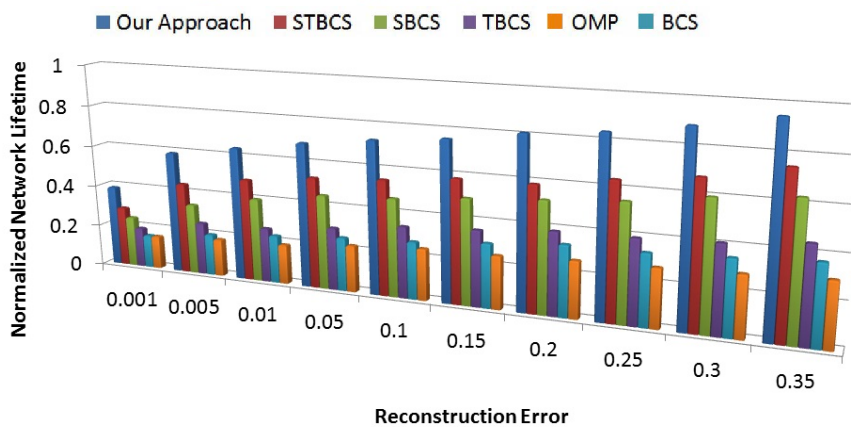


Figure 16. Normalized network lifetime versus reconstruction error trade-off.

## 9. Conclusions

We have presented a distributed compressive sensing scheme that takes the advantage of spatial temporal correlation of sensor node readings to achieve an energy efficient data collection. Spatial correlation among sensor nodes and sparsity distribution of signals over the network is used to group sensor nodes in the coalitions. The proposed coalition-formation method is represented by a block diagonal measurement matrix in which each diagonal entity corresponds to one of the coalitions. The localized spatial temporal correlation inside each coalition builds an efficient measurement matrix that efficiently schedules sensor nodes and balances the data compression and transmission load over the coalition. Upon receiving compressed data in the base station, a two-step belief propagation based signal recovery is implemented to reconstruct the original signal. Having a two-step signal recovery allows the base station to find the common coefficient of original signal among coalitions and inside coalition. Therefore, it has enough data to recover data per sensor node level, which increases data reconstruction accuracy. Coalition-based data gathering is used to scale down transmission costs, while the number of measurements is minimized by scheduling sensor nodes and adjusting their measurement rate. Our simulation on a real world data set proves our findings and shows more reduction on the number of data measurements and compressibility as well as the ability to improve data reconstruction quality.

This paper has also presented a number of transform bases and analyzed their effect on compressive sensing performance in terms of SNR and signal reconstruction error. Since most signals acquired by wireless sensor networks are non-stationary, we considered data collected from a bridge monitoring application as a case study for this impact analysis. The comparison of simulation results using FT, HT and WT demonstrates that WT has the best performance when applying compressive sensing to this case study. Coiflet WT, Daubichi WT, Symlet WT, and Haar WT, the sub-families of WT, were also studied and analysed in terms of SNR and NRMSE.

**Author Contributions:** A.M. proposed the algorithm, performed the theoretical analysis, implemented the simulationss and wrote the entire manuscript. This paper is written under supervision of N.M. and P.J.M.H.

**Funding:** This research is supported by the EU FP7-ICT project “GENESI: Green sEnSOr NETworks for Structural Monitoring”.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Rault, T.; Bouabdallah, A.; Challal, Y. Energy efficiency in wireless sensor networks: A top-down survey. *Comput. Netw.* **2014**, *67*, 104–122. [[CrossRef](#)]
2. Leinonen, M.; Codreanu, M.; Juntti, M. Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks. *IEEE Trans. Wirel. Commun.* **2015**, *14*, 1622–1635. [[CrossRef](#)]
3. Balouchestani, M.; Raahemifar, K.; Krishnan, T. Robust wireless sensor networks with compressed sensing theory. In *Networked Digital Technologies*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 608–619.
4. Middy, R.; Chakravarty, M.; Naskar, K. Compressive Sensing in Wireless Sensor Networks: A Survey. *IETE Tech. Rev.* **2017**, *34*, 642–654. [[CrossRef](#)]
5. Agrawal, A.; Ghosh, D. Distributed compressive data gathering in wireless sensor networks. In Proceedings of the IEEE 11th International Conference on Signal Processing, Beijing, China, 21–25 October 2012; Volume 3, pp. 2110–2115.
6. Shahrabi, B.; Talari, A.; Rahnavard, N. TC-CSBP: Compressive sensing for time-correlated data based on belief propagation. In Proceedings of the 45th Annual Conference Information Sciences and Systems, Baltimore, MD, USA, 23–25 March 2011; pp. 1–6.
7. Haupt, J.; Bajwa, W.U.; Rabbat, M.; Nowak, R. Compressed sensing for net-worked data. *IEEE Signal Process. Mag.* **2008**, *25*, 92–101. [[CrossRef](#)]
8. Zhang, Z.; Jung, T.P.; Makeig, R.; Rao, B.D. Compressed sensing of EEG for wireless telemonitoring with low energy consumption and inexpensive hardware. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 221–224. [[CrossRef](#)] [[PubMed](#)]

9. Aderohunmu, F.A.; Paci, G.; Benini, L.; Deng, J.D.; Brunelli, D. Swiftnet: A data acquisition protocol for fast-reactive monitoring applications. In Proceedings of the 8th IEEE International Symposium on Industrial Embedded Systems (SIES), Porto, Portugal, 19–21 June 2013; pp. 93–96.
10. Luo, C.; Wu, F.; Sun, J.; Chen, C.W. Efficient measurement generation and pervasive sparsity for compressive data gathering. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 3728–3738. [[CrossRef](#)]
11. Hu, H.; Yang, Z. Spatial correlation-based distributed compressed sensing in wireless sensor networks. In Proceedings of the 6th International Conference Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23–25 September 2010; pp. 1–4.
12. Meng, J.; Li, H.; Han, Z. Sparse event detection in wireless sensor networks using compressive sensing. In Proceedings of the 43rd Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, 18–20 March 2009; pp. 181–185.
13. Singh, V.K.; Verma, S.; Kumar, M. DBCS: A Decomposition Based Compressive Sensing for Event Oriented Wireless Sensor Networks. *Wirel. Pers. Commun.* **2018**, *99*, 351–369. [[CrossRef](#)]
14. Yang, X.; Tao, X.; Dutkiewicz, E.; Huang, X.; Guo, Y.J.; Cui, Q. Energy-efficient distributed data storage for wireless sensor networks based on compressed sensing and network coding. *IEEE Trans. Wirel. Commun.* **2013**, *12*, 5087–5099. [[CrossRef](#)]
15. Zheng, H.; Yang, F.; Tian, X.; Gan, X.; Wang, X.; Xiao, S. Data gathering with compressive sensing in wireless sensor networks: A random walk based approach. *IEEE Trans. Parallel Distrib. Syst.* **2015**, *26*, 35–44. [[CrossRef](#)]
16. Mehrjoo, S.; Khunjush, F. Accurate compressive data gathering in wireless sensor networks using weighted spatio-temporal compressive sensing. *Telecommun. Syst. J.* **2018**, *68*, 79–88. [[CrossRef](#)]
17. Cheng, L.; Niu, J.; Kong, L.; Luo, C.; Gu, Y.; He, W.; Das, S.K. Compressive sensing based data quality improvement for crowd-sensing applications. *J. Netw. Comput. Appl.* **2017**, *77*, 123–134. [[CrossRef](#)]
18. Chen, S.; Zhao, C.; Wu, M.; Sun, Z.; Jin, Z. Clustered Spatio-Temporal Compression Design for Wireless Sensor Networks. In Proceedings of the 2015 24th International Conference on Computer Communication and Networks (ICCCN), Las Vegas, NV, USA, 3–6 August 2015; pp. 1–6.
19. Kong, L.; Xia, M.; Liu, X.Y.; Chen, G.; Gu, Y.; Wu, M.Y.; Liu, X. Data loss and reconstruction in wireless sensor networks. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 2818–2828. [[CrossRef](#)]
20. Zhang, D.; Zhang, T.; Zhang, J.; Dong, Y.; Zhang, X. A kind of effective data aggregating method based on compressive sensing for wireless sensor network. *Wirel. Commun. Netw. J.* **2018**, *2018*, 159. [[CrossRef](#)]
21. Candes, E.J.; Wakin, M.B. An introduction to compressive sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 21–30. [[CrossRef](#)]
22. Baraniuk, R.; Davenport, M.A.; Duarte, M.F.; Hegde, L. An Introduction to Compressive Sensing. Connexions e-Textbook. 2011. Available online: <http://legacy.cnx.org/content/col11133/1.5/> (accessed on 2 April 2011).
23. Haupt, J.; Applebaum, L.; Nowak, R. On the restricted isometry of deterministically subsampled Fourier matrices. In Proceedings of the 44th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 17–19 March 2010; pp. 1–6.
24. Kalogerias, D.S.; Petropulu, A.P. Ripboundsofnaivelysubsampledscrambled Fourier sensing matrices. In Proceedings of the 48th Annual Conference on Information Sciences and Systems (CISS), Princeton, NJ, USA, 19–21 March 2014; pp. 1–6.
25. Baron, D.; Duarte, M.F.; Wakin, M.B.; Sarvotham, S.; Baraniuk, R.G. Distributed compressive sensing. *arXiv* **2009**, arXiv:0901.3403.
26. Sundman, D.; Chatterjee, S.; Skoglund, M. Methods for distributed compressed sensing. *J. Sens. Actuator Netw.* **2013**, *3*, 1–25. [[CrossRef](#)]
27. Tan, Z.; Yang, P.; Nehorai, A. Joint sparse recovery method for compressed sensing with structured dictionary mismatches. *IEEE Trans. Signal Process.* **2014**, *62*, 4997–5008. [[CrossRef](#)]
28. Haykin, S.; Van Veen, B. *Signals and Systems*; John Wiley Sons: Hoboken, NJ, USA, 2007.
29. Li, G.; Chang, L.; Li, S. *Signals and Systems: Fundamentals*; Walter de Gruyter GmbH Co KG: Berlin, Germany, 2015.
30. Canli, T.; Gupta, A.; Khokhar, A. Power efficient algorithms for computing fast Fourier transform over wireless sensor networks. In Proceedings of the International Conference on Computer Systems and Applications, Dubai/Sharjah, UAE, 8–11 March 2006; pp. 549–556.

31. Gilbert, A.C.; Strauss, M.J.; Tropp, J.A. A tutorial on fast Fourier sampling. *IEEE Signal Process. Mag.* **2008**, *25*, 57–66. [[CrossRef](#)]
32. Guo, D.; Qu, X.; Xiao, M.; Yao, Y. Comparative analysis on transform and reconstruction of compressed sensing in sensor networks. In Proceedings of the WRI International Conference on Communications and Mobile Computing, Kunming, China, 6–8 January 2009; Volume 1, pp. 441–445.
33. Bentley, P.M.; McDonnell, J.T.E. Wavelet transforms: an introduction. *Electron. Commun. Eng. J.* **1994**, *6*, 175–186. [[CrossRef](#)]
34. Chun-Lin, L. *A Tutorial of the Wavelet Transform*; NTU-EE: Taipei, Taiwan, 2010.
35. Baron, D.; Sarvotham, S.; Baraniuk, R.G. Bayesian compressive sensing via belief propagation. *IEEE Trans. Signal Process.* **2010**, *58*, 269–280. [[CrossRef](#)]
36. Khanna, S.; Murthy, C.R. Decentralized joint-sparse signal recovery: A sparse bayesian learning approach. *IEEE Trans. Signal Inf. Process. Netw.* **2017**, *3*, 29–45. [[CrossRef](#)]
37. Zhang, Z.; Rao, B.A. Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning. *IEEE J. Sel. Top. Signal Process.* **2011**, *5*, 912–926. [[CrossRef](#)]
38. Ziniel, J.; Schniter, P. Efficient high-dimensional inference in the multiple measurement vector problem. *IEEE Trans. Signal Process.* **2013**, *61*, 340–354. [[CrossRef](#)]
39. Tzagkarakis, G.; Starck, J.L.; Tsakalides, P. Joint sparse signal ensemble reconstruction in a wsn using decentralized bayesian matching pursuit. In Proceedings of the 19th European Signal Processing Conference, Barcelona, Spain, 29 August–2 September 2011; pp. 338–342.
40. Masoum, A.; Meratnia, N.; Havinga, P.J.M. A Distributed Compressive Sensing Technique for Data Gathering in Wireless Sensor Networks. *Procedia Comput. Sci.* **2013**, *21*, 207–216. [[CrossRef](#)]
41. Sensor Scope Luce Deployment at Ecole polytechnique Federale de Lausanne. 2008. Available online: <http://lcav.epfl.ch/op/edit/sensorscope-en> (accessed on 22 May 2017).
42. Bhandari, S.; Bergmann, N.; Jurdak, R.; Kusy, B. Time series data analysis of wireless sensor network measurements of temperature. *Sensors* **2017**, *17*, 1221. [[CrossRef](#)] [[PubMed](#)]
43. Caione, C.; Brunelli, D.; Benini, L. Compressive sensing optimization over zigbee networks. In Proceedings of the IEEE Fifth International Symposium on Industrial Embedded Systems (SIES), Trento, Italy, 7–9 July 2010; pp. 36–44.
44. Nguyen M.; Teague, K. Energy efficient data collection in clustered wireless sensor networks employing distributed DCT. *arXiv* **2017**, arXiv:1701.02606.
45. Mahmudimanesh, M. Efficient Saptio-Temporal Sampling in Wireless Sensor Networks Based on Compressive Sampling. Ph.D. Thesis, Darmstadt University of Technology, Darmstadt, Germany, 2015.
46. Masiero, R.; Quer, G.; Rossi, M.; Zorzi, M. A Bayesian analysis of compressive sensing data recovery in wireless sensor networks. In Proceedings of the 2009 International Conference on Ultra Modern Telecommunications and Workshops, St. Petersburg, Russia, 12–14 October 2009; pp. 1–6.
47. Wu, X.; Xiong, Y.; Li, M.; Huang, W. Distributed spatial-temporal compressive data gathering for large-scale WSNs. In Proceedings of the Computing, Communications and IT Applications Conference (ComComAp), Hong Kong, China, 1–4 April 2013; pp. 105–110.
48. Ji, S.; Huang, S.; Wang, J.; Liu, Q. A Reconstruction Method of Compressive Sensing Data Recovery in Wireless Sensor Networks for SHM. In Proceedings of the 2nd International Conference on Next Generation Computer and Information Technology, Qingdao, China, 22–24 September 2013; Volume 27, pp. 99–102.
49. Hwang, S.; Ran, R.; Yang, J.; Kim, D.K. Multivariated Bayesian Compressive Sensing in Wireless Sensor Networks. *IEEE Sens. J.* **2016**, *16*, 2196–2206. [[CrossRef](#)]

