

Article

Research on Data Fusion Scheme for Wireless Sensor Networks with Combined Improved LEACH and Compressed Sensing

Yu Song ^{1,2,3,*}, Zhigui Liu ^{1,*}, Xiaoli He ^{1,3,4} and Hong Jiang ¹

¹ School of Information Engineering, South West University of Science and Technology, Mianyang 621010, China; hexiaoli@suse.edu.cn (X.H.); jianghong@swust.edu.cn (H.J.)

² Department of Network Information Management Center, Sichuan University of Science and Engineering, Zigong 643000, China

³ Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong 643000, China

⁴ School of Computer Science, Sichuan University of Science and Engineering, Zigong 643000, China

* Correspondence: songyu@suse.edu.cn (Y.S.); liuzhigui@swust.edu.cn (Z.L.); Tel.: +86-159-8416-9508 (Y.S.); +86-137-7858-9198 (Z.L.)

Received: 27 August 2019; Accepted: 25 October 2019; Published: 29 October 2019



Abstract: There are a lot of redundant data in wireless sensor networks (WSNs). If these redundant data are processed and transmitted, the node energy consumption will be too fast and will affect the overall lifetime of the network. Data fusion technology compresses the sampled data to eliminate redundancy, which can effectively reduce the amount of data sent by the node and prolong the lifetime of the network. Due to the dynamic nature of WSNs, traditional data fusion techniques still have many problems. Compressed sensing (CS) theory has introduced new ideas to solve these problems for WSNs. Therefore, in this study we analyze the data fusion scheme and propose an algorithm that combines improved clustered (ICL) algorithm low energy adaptive clustering hierarchy (LEACH) and CS (ICL-LEACH-CS). First, we consider the factors of residual energy, distance, and compression ratio and use the improved clustered LEACH algorithm (ICL-LEACH) to elect the cluster head (CH) nodes. Second, the CH uses a Gaussian random observation matrix to perform linear compressed projection (LCP) on the cluster common (CM) node signal and compresses the N-dimensional signal into M-dimensional information. Then, the CH node compresses the data by using a CS algorithm to obtain a measured value and sends the measured value to the sink node. Finally, the sink node reconstructs the signal using a convex optimization method and uses a least squares algorithm to fuse the signal. The signal reconstruction optimization problem is modeled as an equivalent ℓ_1 -norm problem. The simulation results show that, compared with other data fusion algorithms, the ICL-LEACH-CS algorithm effectively reduces the node's transmission while balancing the load between the nodes.

Keywords: wireless sensor networks; data fusion; clustering routing; compressive sensing

1. Introduction

A wireless sensor network (WSN) is a multi-hop self-organizing network system consisting of a large number of sensor nodes with sensing, computing, and communication functions [1]. This type of network is widely used in environmental monitoring, agricultural production, defense military surveillance, industrial automation control, medical monitoring, intelligent robots, and smart cities (such as intelligent transportation, smart homes, and smart travel). Nodes typically have low power and battery-powered features, and once the battery is exhausted, it is unrealistic to replace or charge

the battery. Therefore, effectively reducing the energy consumption of WSNs remains a focus of current research [2].

WSNs process specific information in the network coverage area through cooperative sensing, information collection, and data fusion (aggregation) and transmit the information to the base station (BS) by multi-hop forwarding [3]. Data fusion is used to process redundant data or information to combine more energy-efficient and more accurate information in order to improve the efficiency of data collection. Data fusion plays an important role in the research of WSNs and is an effective method to reduce the energy consumption of WSNs [4].

Data fusion can effectively decrease the amount of data transmission and reduce energy consumption. However, due to the strong dynamics of WSNs, solving the problem of the real-time accuracy and reliability of data fusion has become a research hotspot. Compressed (or compressive) sensing (CS) as a new research direction of information science has provided new ideas for solving these problems [5,6]. Therefore, how to apply the theory of compressed sensing to WSN data fusion has become an important research topic.

1.1. Related Works

At present, there are many algorithms for WSN data fusion. These studies focus on the following aspects: energy-balanced data fusion [7,8], tree-based data fusion [9], performance-based data fusion [10,11], and security-based data fusion [12–14]. These data fusion solutions have driven the development of WSNs. However, the data fusion efficiencies of these traditional data fusion algorithms were generally not high. At the same time, the above studies were based on the Nyquist sampling theorem, which meant that the original signal could be reconstructed from the sampled signal only when the sampling frequency was higher than twice the maximum frequency of the original signal [5]. Of course, this theorem is still valid. However, in many new fields, this method will greatly increase the cost and waste of studies and may even fail to reach the sampling frequency due to hardware limitations. Scientists such as E. J. Candes, J. Romberg, T. Tao, and D. L. Donoho proposed the CS theory in 2004, which can acquire discrete samples of the signal at a much smaller rate than the Nyquist sampling rate, ensuring distortion-free reconstruction of the signal [6]. As a new sampling theory, CS has always been the focus of academia and industry [15,16]. Therefore, CS is considered when studying the data fusion of WSNs to reduce the amount of data transmission.

Recently, the application of CS theory in the data fusion of WSNs has been studied [17,18]. The research in reference [17] studied the problem of joint sparse support recovery with 1-bit quantized compressive measurements in a distributed sensor network. For sparsity pattern recovery with 1-bit quantized measurements, the study of reference [18] have developed two computationally tractable centralized algorithms (i.e., MLA and C-BIHT) and proposed a new iterative dictionary learning algorithm. It can be used to reconstruction the noise sparse signal under a one-bit CS framework without previously unknown the sparse domain. Using the dictionary learning algorithm to train the two matrices (i.e., the measurement matrix and sparse domain matrix), the experimental results showed that the algorithm was effective for a case with no dictionary learning.

The above methods gradually add the information of each node to the data being transmitted by encoding during the process of data transmission, thereby solving the problem of an unbalanced network load. However, due to the large number of nodes and the wide distribution in the WSNs, the node can potentially transmit the gathered data directly to the sink node without increasing the energy consumption; this is an urgent problem that needs to be solved. Data fusion based on clustering routing has been proposed and has received extensive attention. In the study of [19], Heinzelman proposed a low energy adaptive clustering hierarchy (LEACH) algorithm, which is the first hierarchical routing protocol for data fusion and is a classic clustering protocol. It solves the problem of the limited network scale caused by the planar routing protocol and supports larger scale networks.

Energy is limited in the WSNs, so how to choose an effective data fusion solution is especially important. The idea of combining clustering algorithm and CS technology to process the information

has become a new trend in WSNs data fusion research [20–26]. The research in reference [20] investigated an energy-efficient data aggregation scheme to reduce communication costs and prolong the network's lifetime for a cluster-based underwater acoustic sensor network inspired by the theory of distributed compressed sensing (DCS). A joint reconstruction model highly consistent with JSM-2 and the observation matrix were newly designed in [20]. The study in reference [21] introduced clustering-based CS on the basis of clustering (cluster compression sensing, CCS) to further compress the data. Since the CH of CCS sent the linear compressed projection information to the sink node, the linear compressed projection projected the high-dimensional signal to the low-dimensional measurement value, and the information reconstruction of the sink node increased the dimension from low-dimensional to high-dimensional.

The authors of reference [22] proposed a novel energy-efficient CS-based clustering routing (EECSR) protocol that combined the merits of the clustering strategy and the CS-based scheme. To alleviate the “hot spot problem” and reduce the energy consumption resulted from the rotation of the role of CHs, a third role of backup CH (BCH) as well as the corresponding mechanism to rotate the roles between the CH and BCH was proposed in reference [22]. Experiments shown that the EECSR was superior to the existing clustering algorithms and the CS-based algorithms in terms of energy efficiency and extending the lifetime of WSNs. The study in reference [23] investigated a kind of effective data aggregating method based on the CS and the clustering. The aggregation process was divided into two parts, i.e., intra-cluster and inter-cluster. In the cluster, the sink node sent the seed vector to each CH. The CH generated the corresponding random interval sparse matrices and collected data by CH. Among clusters, the cluster nodes forwarded the measurements along the multi-hop routing tree to the sink node. In this paper, we divide the data fusion process into two parts (i.e., in the cluster and among the clusters) according to the idea of reference [22]. The study in reference [24] focused on the energy-efficient data gathering method based on the integration of the clustering and the CS. In a clustered WSN, the CH collected the data from the non-CH nodes, added all received data to its own data, and then sent the combined measurements to the BS. Experimental verification was performed using K-means and LEACH. However, the energy consumption of multi-hop inter-cluster communication in the network is not considered in reference [24].

The most relevant research to our work is reference [25]. A routing algorithm, based on a dual CH redundant mechanism combined with a CS data fusion algorithm, was proposed in reference [25]. A dual CH alternation mechanism was used to balance the energy consumption of the CH node. Eliminating redundancy through CS data fusion technology effectively improved the reliability of WSNs and reduced data redundancy. Inspired by reference [25], we combine clustering routing, data fusion, and CS. The CH selection strategy of reference [25] was similar to a traditional WSN clustering routing scheme, but the difference was that a dual CH alternation mechanism was considered. Although this dual CH mechanism could alleviate the burden on the CH node, it created additional energy consumption. Therefore, designing an effective data fusion scheme based on clustering routing and CS while considering node energy efficiency is an important area of research for WSNs. To the best of our knowledge, this work remains an open topic for those who study WSN data fusion.

1.2. Contributions

In this research, we investigate a clustering data fusion scheme based on CS for WSNs. We aim to reduce the node energy consumption, reduce the amount of data transmission, improve the signal reconstruction accuracy, and prolong the network's lifetime. Thus, the main innovations of this paper can be summarized as follows:

- Using the spatial correlation between nodes in WSNs, this paper proposes a clustering data fusion method based on an improved LEACH clustering protocol and sparse hybrid CS. The data fusion process is divided into two parts, clustering and CS.
- In the clustering process, this paper improves the shortcomings (e.g., clustering is not uniform, and it is easy to make a node repeat when the CH energy's premature consumption is completed

as a dead point) of the LEACH algorithm. We consider the influence of residual energy, distance, and compression ratios on the CH node selection and propose a new threshold function.

- In the CS process, we convert the solution of the ℓ_0 -norm to the solution of the ℓ_1 -norm and transform the non-convex optimization problem into a convex optimization problem. The convex optimization Lagrangian dual function is used to solve the optimization problem and reconstruct the sensor node information. This algorithm has a fast convergence speed and low time complexity. At the same time, it is possible to optimize the network topology and balance the energy consumption of each node in the network.
- The performance analysis and comparison of the experimental results and related methods show that the proposed algorithm can reduce the sampling information of the nodes and better reconstruct the node source, thereby making the network more adaptable and robust.

1.3. Organization

The remainder of this paper is organized as follows. In Section 2, the system model is introduced. The algorithm (i.e., the clustering algorithm and compressive sensing data fusion algorithm) for a WSN scenario is designed and implemented in Section 3. Numerical results are presented in Section 4. Finally, Section 5 concludes the paper.

2. System Model

2.1. Network Model

In this paper, we consider a distributed WSN scenario, in which energy is limited. Therefore, choosing an effective energy-saving clustering algorithm and data fusion algorithm is very important for solving WSN data transmission and data processing. The system model diagram is shown as Figure 1. More specifically, we assume that there are a total of N sensor nodes in the network, which are randomly and evenly deployed within the region, and the sink node is at the center.

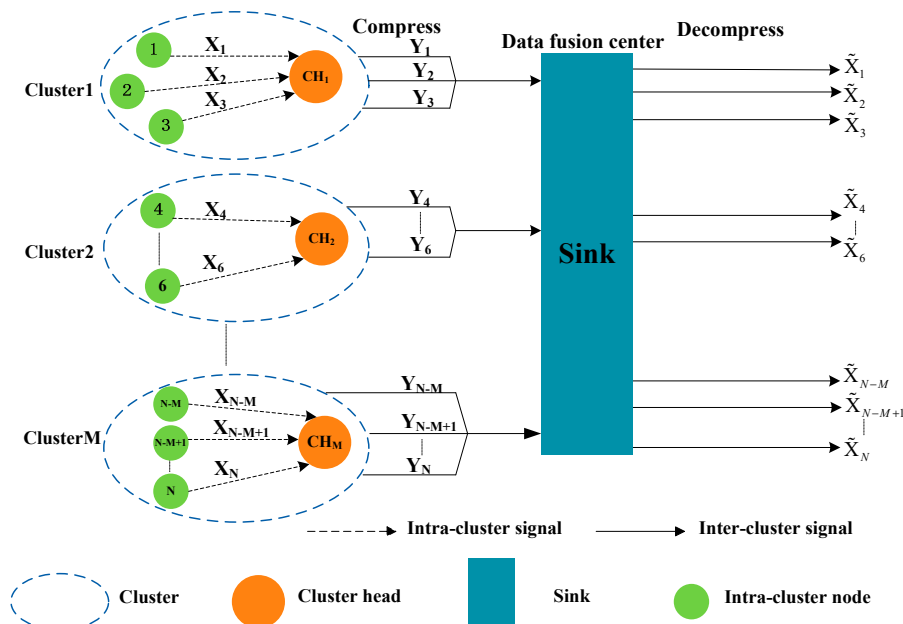


Figure 1. Wireless sensor network (WSN) system model based on clustering data fusion with compressed sensing (CS).

We can see from Figure 1 that, with the clustering algorithm, the nodes of the WSNs are divided into M clusters, and each cluster has a CH node, where $\Phi_{CH} = \{1, 2, \dots, M\}$ is a set of CH. In the cluster, the i -th node transfers the data X_i ($0 \leq i \leq N$) to the CH node in a single hop manner. The CH node

observes the gathered information $Y_i (0 \leq i \leq N)$ and then transfers the observations to the aggregation node (i.e., the sink node), which recovers the original data estimated value $\hat{X}_i (0 \leq i \leq N)$ via the reconstruction algorithm. At the same time, we also assume that the nodes in each cluster are evenly distributed—that is, there is one CH node in each cluster, and there are $n = \frac{N}{M} - 1$ cluster member (CM) nodes. In this paper, it is assumed that there are two levels of transmissions, namely, intra-cluster (i.e., nodes to CH node) signal transmission without CS, and inter-cluster (i.e., CH node to sink node) signal transmissions based on CS. Data are transmitted in a single hop, either within the cluster or outside the cluster.

2.2. Energy Consumption Model

In this paper, energy consumption includes the required energy for transmitting, receiving, and processing data packets. According to the classical wireless communication energy consumption model [19], we assume that the energy of the transmitter includes both signal processing and power amplification, while the energy of the receiver is only used for signal processing. The Euclidean theory can be used to describe the distance between two nodes. Then, the distance between node i and node j can be expressed as follows:

$$d_{ij} = \sqrt{(d_i - d_j)^2 + (\hat{d}_i - \hat{d}_j)^2}, \forall i, j \in \{1, 2, \dots, N\} \quad (1)$$

where the node i and node j can also be denoted as $(d_i, \hat{d}_i)$ and $(d_j, \hat{d}_j)$.

We assume that the free space model is used when the distance between the CM nodes and the CH node is less than d_0 , and the multipath fading model is used when the distance between the CH node and the sink node is greater than d_0 .

The energy that consumed by the transmission of l -bit messages between the two nodes with a distance of d is as follows:

$$\begin{aligned} E_{TX} &= \underbrace{E_{TX-elec}}_{\text{energy consumed by transmitting circuit}} + \underbrace{E_{TX-amp}}_{\text{energy consumed by amplifying circuit loss}} \\ &= \begin{cases} lE_{elec} + l\varepsilon_{fs}d^2, d < d_0, \text{ intra-cluster (CM to CH)} \\ lE_{elec} + l\varepsilon_{mp}d^4, d \geq d_0, \text{ inter-cluster (CH to Sink)} \end{cases} \end{aligned} \quad (2)$$

where $d_0 = \sqrt{\varepsilon_{fs}/\varepsilon_{mp}}$, E_{elec} is the energy required for signal processing, and the amplifier power consumption ε_{fs} and ε_{mp} are determined by the transmission distance and the received bit error rate.

The energy consumed by the l -bit messages' reception is as follows:

$$E_{RX} = \underbrace{E_{RX-elec}}_{\text{energy consumed by reception circuit}} = lE_{elec} \quad (3)$$

When a message is transmitted by means of cluster communication, the total energy consumption of the network is composed of the energy consumption of CH and CM. Furthermore, the energy consumption of the intra-cluster CH node consists of receiving the data from the CM node, and data fusion. While the energy consumption of the inter-cluster CH node consists of the transmitting data to the sink and the CS. The specific energy consumption is as follows:

$$\begin{aligned} E_{CHi-intra} &= \underbrace{E_{CHi-RX-intra}}_{\text{energy consumed by reception}} + \underbrace{E_{CHi-DF-intra}}_{\text{energy consumed by data fusion}} \\ &= nlE_{elec} + (n+1)lE_{df} \\ &= l(nE_{elec} + (n+1)E_{df}) \end{aligned} \quad (4)$$

where n is the number of the CM nodes, and E_{df} is the data fusion cost of 1-bit messages in a cluster. Likewise, the energy consumption of the inter-cluster CH node is given by Equation (5).

$$\begin{aligned}
 E_{CH_i-inter} &= + \underbrace{E_{CH_i-TX-inter}}_{\text{energy consumed by transmitting}} + \underbrace{E_{CH_i-CS-inter}}_{\text{energy consumed by compressed sensing}} \\
 &= (n+1)\alpha \left(lE_{elec} + l\epsilon_{mp}d_{CH_i \text{ to sink}}^4 \right) + (n+1)lE_{cs} \\
 &= (n+1)l \left(\alpha \left(E_{elec} + \epsilon_{mp}d_{CH_i \text{ to sink}}^4 \right) + E_{cs} \right)
 \end{aligned} \quad (5)$$

where α is the compressed ratio, which is related to CS, $d_{CH_i \text{ to sink}}$ is the distance between the i -th CH and the sink, and E_{cs} is the compressed sensing cost of 1-bit messages in a cluster.

Hence, according to Equations (4) and (5), the total energy consumption of the i -th CH will be as follows:

$$\begin{aligned}
 E_{CH_i} &= E_{CH_i-intra} + E_{CH_i-inter} \\
 &= l \left(nE_{elec} + (n+1)E_{df} \right) + (n+1)l \left(\alpha \left(E_{elec} + \epsilon_{mp}d_{CH_i \text{ to sink}}^4 \right) + E_{cs} \right) \\
 &= l \left(\left(nE_{elec} + (n+1)E_{df} \right) + (n+1) \left(\alpha \left(E_{elec} + \epsilon_{mp}d_{CH_i \text{ to sink}}^4 \right) + E_{cs} \right) \right)
 \end{aligned} \quad (6)$$

Meanwhile, the total energy consumption of the CM nodes in the i -th cluster is calculated by Equation (7):

$$E_{CM-CH_i} = \sum_{j=1}^n E_{CM_j-CH_i} = \sum_{j=1}^n l \left(E_{elec} + \epsilon_{fs}d_{CM_j \text{ to } CH_i}^2 \right) \quad (7)$$

where $E_{CM_j-CH_i}$ is the amount of energy consumed by the j -th CM node in the i -th cluster, and $d_{CM_j \text{ to } CH_i}$ is the distance between the j -th CM and the i -th CH.

As a result, the total energy consumption of the WSNs is expressed according to in Equation (8).

$$\begin{aligned}
 E_{total} &= \sum_{i=1}^M (E_{CH_i} + E_{CM-CH_i}) \\
 &= \sum_{i=1}^M \left(l \left(\left(nE_{elec} + (n+1)E_{df} \right) + (n+1) \left(\alpha \left(E_{elec} + \epsilon_{mp}d_{CH_i \text{ to sink}}^4 \right) + E_{cs} \right) \right) + \sum_{j=1}^n l \left(E_{elec} + \epsilon_{fs}d_{CM_j \text{ to } CH_i}^2 \right) \right) \\
 &= l \sum_{i=1}^M \left(\left(\left(nE_{elec} + (n+1)E_{df} \right) + (n+1) \left(\alpha \left(E_{elec} + \epsilon_{mp}d_{CH_i \text{ to sink}}^4 \right) + E_{cs} \right) \right) + \sum_{j=1}^n \left(E_{elec} + \epsilon_{fs}d_{CM_j \text{ to } CH_i}^2 \right) \right)
 \end{aligned} \quad (8)$$

Then, we can obtain the residual energy of the CH node and CM node in the r -th round, as follows:

$$E_{CH_i-residual}^r = \begin{cases} \min(E_{\max}, E_{CH_i-residual}^{r-1} - E_{CH_i}^r), & r > 0 \\ E_{\text{initial}}, & r = 0 \end{cases} \quad (9)$$

$$E_{CM-CH_i-residual}^r = \begin{cases} \min(E_{\max}, E_{CM-CH_i-residual}^{r-1} - E_{CM-CH_i}^r), & r > 0 \\ \hat{E}_{\text{initial}}, & r = 0 \end{cases} \quad (10)$$

where E_{initial} and $\hat{E}_{\text{initial}}$ are the initial energies of the CH and CM nodes, respectively; $E_{\max}$ is the maximum capacity of the battery; and $E_{CH_i}^r$ and $E_{CM-CH_i}^r$ are the energy consumption of the CH and CM nodes in the r -th round.

2.3. CS Data Fusion Model

In this section, we will introduce CS theory and its application in WSNs. The model diagram combining CS theory and WSN data fusion is shown in Figure 2.

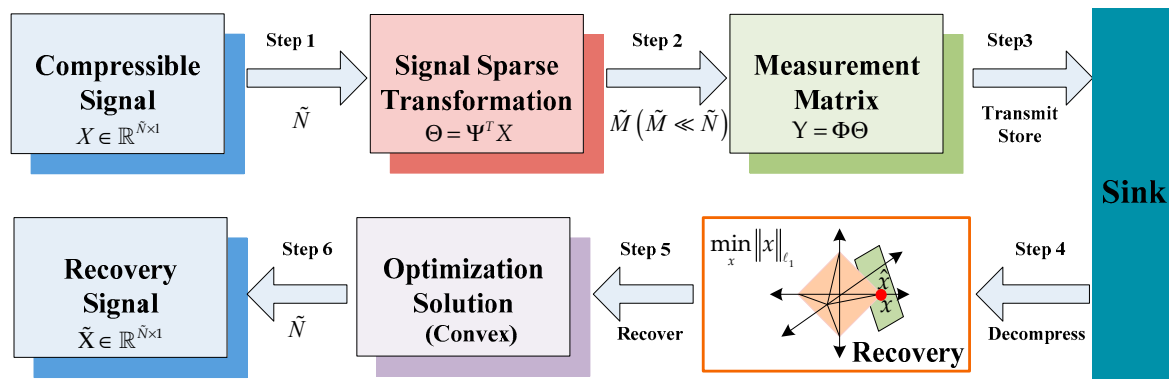


Figure 2. Compressed sensing data fusion model.

As can be seen from Figure 2, the idea of CS data fusion theory is as follows. For signal $X \in \mathbb{R}^{\tilde{N} \times 1}$, we will first judge whether it is sparse or compressible. If it is not a sparse signal, then we need to find a sparse basis $\Psi \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$, which makes X sparse based on the orthogonal transform basis Ψ . The sparse signal Θ (i.e., Θ is K sparse) and the measurement matrix $\Phi \in \mathbb{R}^{\tilde{M} \times \tilde{N}} (\tilde{M} \ll \tilde{N})$ are operated to obtain a measurement value $Y \in \mathbb{R}^{\tilde{M} \times 1}$, provided that the measurement matrix Φ is not related to the sparse base Ψ (i.e., the restricted isometry property (RIP) is satisfied). The sink node performs data fusion on the received signals. It receives and stores the low-dimensional measurement vector Y of the original signal X and recovers the original signal using a recovery algorithm. We intend to use the convex algorithm to optimize the solution and recover the original signal to get $\tilde{X}$.

As can be seen from Figure 2, there are three important aspects in the research of CS data fusion—signal sparse transformation (the orthogonal basis for sparse representation), measurement matrix design (the measurement matrix satisfying the RIP property) [6], and signal reconstruction (a robust and efficient reconstruction algorithm). Next, we will introduce these three aspects in detail.

2.3.1. Signal Sparse Transformation

Suppose any compressible signal $X = [x_1, x_2, \dots, x_{\tilde{N}}]^T$ of length $\tilde{N}$ can be represented by Equation (11).

$$X = \sum_{i=1}^{\tilde{N}} \psi_i \theta_i \quad \text{or} \quad X = \Psi \Theta \quad (11)$$

$$\theta_i = \langle X, \psi_i \rangle \quad \text{or} \quad \Theta = \Psi^T X \quad (12)$$

where X and Θ are different expressions of the signal X in two different transform domains (i.e., the time domain and the frequency domain). $\Psi = [\psi_1, \psi_2, \dots, \psi_{\tilde{N}}]$ is a set with a standard orthogonal basis. ψ_i is the column of Ψ . $\Theta = \{\theta_i\}_{i=1}^{\tilde{N}}$ is the column vector of element θ_i . If most of the elements of the vector Θ are 0, then X is called sparse. The number of non-zero is denoted as $K (K \approx \tilde{M} \ll \tilde{N})$, i.e., $\|\Theta\|_0 = K$, and X is said to be K sparse.

2.3.2. Measurement Matrix Design

If the projection vector Θ is strictly sparse or compressible, then we can perform a compression measurement. Compression measurement is a process of linearly measuring a high-dimensional original signal to obtain a low-dimensional vector. The matrix $\Phi = [\phi_1, \phi_2, \dots, \phi_{\tilde{M}}]^T$ in Figure 2 is the measurement matrix. The restricted isometry property (RIP) [6] theory has proven that the necessary

and sufficient condition for accurately recovering signal X is that Φ and Ψ are irrelevant. In this case, we can project the original signal X via the measurement matrix Φ to obtain the measurement vector $Y = [y_1, y_2, \dots, y_M]$. The inner product y_j of X and $\{\phi_j\}_{j=1}^{\tilde{M}}$ can then be obtained from Equation (13).

$$y_j = \langle X, \phi_j \rangle \quad (j = 1, 2, \dots, \tilde{M}) \quad (13)$$

According to Equations (11)–(13), the following expression can be obtained:

$$Y = \Phi X = \Phi \Psi \Theta = \hat{\Phi} \Theta \quad (14)$$

where Y is the column vector of $\tilde{M} \times 1$, Φ is the measurement matrix of $\tilde{M} \times \tilde{N}$ ($\tilde{M} \ll \tilde{N}$), and $\hat{\Phi} = \Phi \Psi \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$ is the “sensing” matrix. $\tilde{M}/\tilde{N}$ is called the compression ratio, which reflects the compression performance of the CS algorithm. The lower the compression ratio, the lower the cost of sampling the original signal, and the more efficient the compression.

2.3.3. Signal Reconstruction

In this section, the signal reconstruction (i.e., we will recover $X \in \mathbb{R}^{\tilde{N} \times 1}$ from $Y \in \mathbb{R}^{\tilde{M} \times 1}$ ($\tilde{M} \ll \tilde{N}$)) remains a serious problem. Since the original signal X is K sparse in a certain transform domain, Equation (14) has a unique sparse solution when $K \approx \tilde{M} \ll \tilde{N}$. Therefore, when the “sensing” matrix $\hat{\Phi} \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$ satisfies the RIP constraints (Please refer to Appendix A for the details on RIP), the under-determined equation problem of the Equation (14) can be transformed into the minimum ℓ_0 -norm problem.

$$\begin{aligned} \min & \|\Psi^T X\|_{\ell_0} \\ \text{s.t. } & Y = \hat{\Phi} \Theta \end{aligned} \quad (15)$$

The ℓ_0 -norm represents the number of non-zero entries in the vector, which is a non-deterministic polynomial (NP) hard problem. Therefore, the solution to the ℓ_0 -norm can be equivalent to the solution of the ℓ_1 -norm, that is, the non-convex optimization problem is transformed into a convex optimization problem. Consequently, we can recover X from Y by solving the following convex optimization algorithm:

$$\begin{aligned} \min & \|\Psi^T X\|_{\ell_1} \\ \text{s.t. } & Y = \hat{\Phi} \Theta \end{aligned} \quad (16)$$

3. Algorithm Design and Implementation

The energy of the nodes in WSNs is limited. Designing an efficient energy-saving clustering algorithm and data fusion algorithm will greatly reduce the energy consumption of the network. Therefore, in this section, we introduce the clustering algorithm and the clustering data fusion algorithm based on CS. The detailed algorithm is as follows. The network model is shown in Figure 1. After choosing the CH nodes and the forming clusters, all nodes in each cluster send packets to the CH node in a single hop manner. Then, the CH node sends the fused packets to the sink node by single hop or multi-hop. Finally, the sink node chooses a reasonable CS reconstruction algorithm to reconstruct the network data according to the characteristics of the network and the node data.

3.1. Improved LEACH Cluster Algorithm

The LEACH algorithm is an adaptive clustering topology algorithm. The idea of the LEACH algorithm is to randomly select the CH node in a round manner and distribute the energy load of the entire network evenly to each sensor node in the network, thereby reducing the network’s energy consumption and improving the network’s lifetime. Compared with other planar multi-hop routing

protocols and static layering algorithms, LEACH could extend the network's lifetime by 15%. Although LEACH can improve the lifetime of WSNs, there are still some problems, as follows.

- In the LEACH algorithm, each round of loops must reconstruct the cluster, and the energy cost of constructing the cluster is relatively large.
- Since the LEACH algorithm assumes that all nodes can communicate directly with the sink node, this protocol is not suitable for use in large-scale WSNs.
- The LEACH algorithm does not consider the current energy status of the CH node. If the node with lower energy is selected as the CH node, it might accelerate the death of the node and affect the lifetime of the entire network.
- The LEACH algorithm does not consider the distance between the CH node and the sink node. When the CH node is far away from the sink node, it still uses single-hop communication, which causes the node to consume significant energy and even exhausted energy.
- The number and distribution of the CH nodes were not considered. Therefore, there might be an unbalanced distribution of the selected CH nodes. In some places, there were many CH nodes, and some places did not have any CH nodes.

Considering the above deficiencies of the LEACH algorithm, in this paper, the chance that each node is selected as CH is determined according to the residual energy of the node and the communication distance. A node that can be selected as a CH node with a high probability has a minimum time delay compared to its neighbor nodes. Based on the LEACH protocol, this paper proposes an improvement clustering algorithm (ICL-LEACH) that comprehensively considers the residual energy, communication distance, and data fusion rate of nodes. The energy of this algorithm is sufficient, its energy consumption is slow, the threshold for a node to have a good position and high data fusion rate is larger, and its chance of being the CH is increased, so that the network load is evenly distributed and the network's lifetime is prolonged.

3.1.1. Setup Phase

At the initial phase of the CH election, the node i randomly picks a number between 0 to 1. If the number is less than the threshold values T_i , then node i becomes the CH in this current round. Considering the influence of the residual energy, distance and compression ratio on the selection of CH nodes, this paper introduces an improved threshold function T_{i-new} , which is defined as in Equation (17).

$$T_{i-new} = \begin{cases} \frac{p}{1-p \times (r \bmod p^{-1})} \times \omega, & i \in I \\ 0, & otherwise \end{cases} \quad (17)$$

where p is the expected ratio that selected as a CH node, and r is the current round number. $(r \bmod p^{-1})$ is the number of the CH in this CH selection period. I is the set of nodes that have not been selected as the CH node in the last p^{-1} round, and ω is the value of a dynamic variable, which is defined as in Equation (18),

$$\omega = \lambda_1 \times \frac{E_{i-residual}}{E_{i-max}} + \lambda_2 \times \frac{d_{max} - d_{i-to\ sink}}{d_{max}} + \lambda_3 \times \alpha \quad (18)$$

$$\alpha = \frac{l_{original}}{l_{current}} \quad (19)$$

$$\lambda_1 + \lambda_2 + \lambda_3 = 1 \quad (20)$$

where $E_{i-residual}$ and E_{i-max} are the residual energy and the maximum energy of node i , respectively. This may increase the probability that a node with higher residual energy is selected as a CH. $d_{i-to\ sink}$ and d_{max} are the distance from node i to the sink node and the maximum distance from the entire node to the sink node, respectively. When the distance from the node to the sink node is too large, it needs to consume too much energy, and the probability that this node is selected as the CH is also greatly

reduced. $l_{current}$ -bits and $l_{original}$ -bits are the current and original transmission message lengths of the node. α is the compression ratio. When α is higher, the probability that the node is selected as the CH is also higher. λ_1 , λ_2 , and λ_3 are the weighting factors that represent the relationship between the energy, distance, and compression.

Once the CH node is selected, the CH broadcasts the selected message to all the nodes. The other common node selects the cluster to be joined according to the strength of the received signal and sends the information to the corresponding CH node. Then, the CH node uses time division multiple access (TDMA) to allocate the time slots for transmitting data to the member nodes in the cluster.

3.1.2. Stabilization Phase

In the stabilization phase, the member nodes in the cluster transfer the collected data to the CH node. When the CH node receives the data sent by all the nodes in the cluster, the CH node performs information fusion on the data. The fused data are then transmitted to the sink node. The sink node transmits the data to the monitoring center for data processing. Since the direct data transmission of each cluster is not performed separately, the communication in each cluster may be affected by the adjacent cluster. Therefore, each cluster communicates with different code division multiple access (CDMA) codes to reduce the interference of other cluster nodes. After the stabilization phase continues for a period of time, the network re-enters the cluster setup phase and begins to select a new round of CH nodes, which continually circulate. In order to save the resource overhead, the duration of the stabilization phase is longer than the duration of the setup phase during each round of clustering.

In general, the process of the ICL-LEACH algorithm can be described as the process in Figure 3, which clearly illustrates the concept of the algorithm.

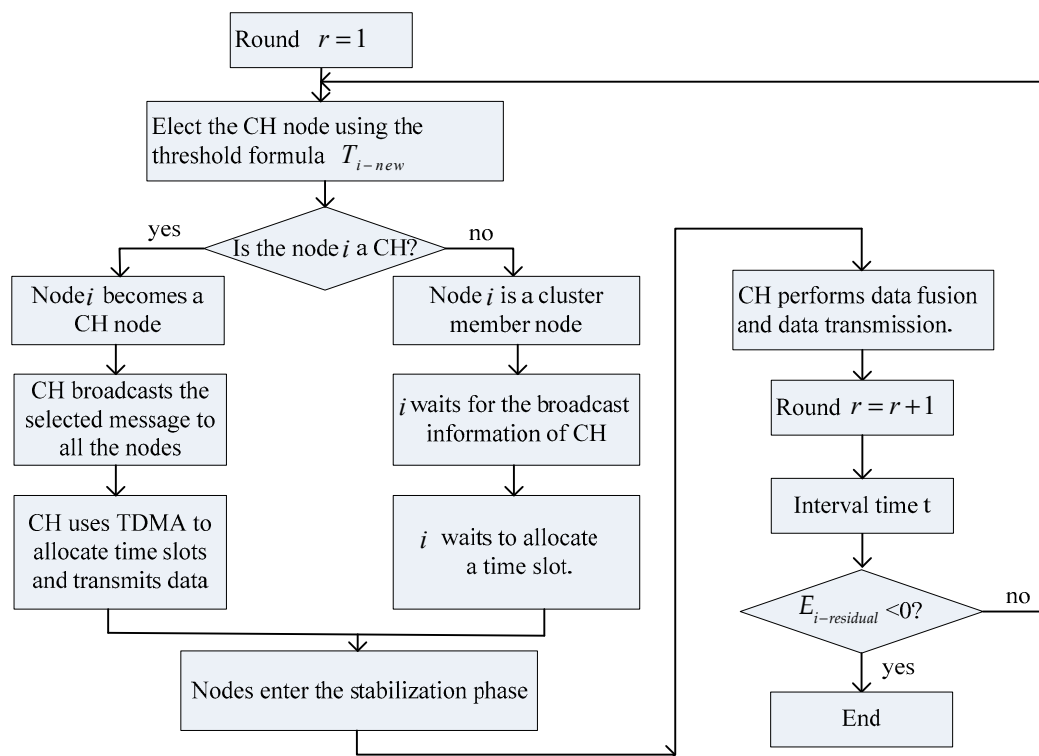


Figure 3. The improved clustered low energy adaptive clustering hierarchy(ICL-LEACH) algorithm flow chart.

3.2. Clustering Data Fusion Algorithm Based on Compressive Sensing

In WSNs, the energy resources are limited and valuable. Thus, the use of energy-saving algorithms (i.e., the combination of the clustering algorithm and the CS algorithm) to process the information can

effectively reduce the network's energy consumption. In this section, we will use CS for data fusion after the clustering in the previous section. For the physical characteristics of the WSNs, the CH node completes the data compression processing, and the sink node completes the data reconstruction.

3.2.1. Selection of the Sparse Transform Basis

According to the system model diagram of Figure 1, it can be seen that, with the ICL-LEACH algorithm, the WSN is divided into M clusters. In other words, there are M CH nodes. We use $CH_m (m \in \Phi_{CH})$ to indicate the m -th CH node. In the m -th cluster, the CH_m node collects the data in its own cluster, and the i -th $(0 \leq i \leq n = \tilde{N})$ node (i.e., CM_i) transfers the sensing signal x_{mi} to the CH_m node. At time t , the signal sent by each node to the CH_m in the cluster is $X_m = [x_{m1}, x_{m2}, \dots, x_{m\tilde{N}}]^T$. $X_m \in \mathbb{R}^{\tilde{N} \times 1}$ is a time-dependent $\tilde{N}$ -dimensional discrete signal. We assume that, in this node-intensive WSN, the signals perceived by the nodes have a spatial-temporal correlation [26]. Generally speaking, signals are not sparse in the time domain. Therefore, before applying CS theory, the signal must first be sparsely transformed to find its sparse basis. Typical signals are sparse on a Wavelet transform basis, a fast Fourier transform basis, and a discrete cosine transform basis. In this paper, we choose the classical Fourier transform sparse method. This method has the advantages of good stability and low computational complexity. Signal sparse representation is as follows:

$$X_m = \Psi_m \Theta_m = \begin{bmatrix} \psi_{m1} & \psi_{m2} & \dots & \psi_{m\tilde{N}} \end{bmatrix} \begin{bmatrix} \theta_{m1} \\ \theta_{m2} \\ \dots \\ \theta_{m\tilde{N}} \end{bmatrix} = \sum_{i=1}^{\tilde{N}} \psi_{mi} \theta_{mi} \quad (21)$$

where $\Psi_m \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$ is the standard orthogonal basis in the m -th cluster, and if X_m is sparse, then Ψ_m is the identity matrix. $\Theta_m \in \mathbb{R}^{\tilde{N} \times 1}$ represents the coefficient of the signal X_m on this sparse basis, and Θ_m has $K (K \approx \tilde{M} \ll \tilde{N})$ non-zero elements. Θ_m can also be expressed in Equation (22).

$$\Theta_m = \Psi_m^T X_m \quad (22)$$

3.2.2. Measurement Matrix Optimization

If the signal $X_m \in \mathbb{R}^{\tilde{N} \times 1}$ can be represented by a sparse basis $\Psi_m \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$, then X_m can be compressed. Specifically, we can project signal X_m to get the main data without losing important information. Therefore, it is very important to design a suitable measurement matrix. The measurement matrix not only reduces the number of dimensions but also minimizes the loss of information of the original signal X_m . We generate a Gaussian random measurement matrix $\Phi_m \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$ at the CH_m and use $\Phi_m \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$ to project $X_m \in \mathbb{R}^{\tilde{N} \times 1}$ to obtain $Y_m \in \mathbb{R}^{\tilde{M} \times 1}$.

$$Y_m = \Phi_m X_m + N_m = \Phi_m \Psi_m \Theta_m + N_m = \hat{\Phi}_m \Theta_m + N_m \quad (23)$$

where N_m denotes a noise vector which will be the zero vector when we regard the noiseless case. The component of N_m obeys a Gaussian independent distribution of random variables of $N(0, \sigma^2)$,

and $\hat{\Phi}_m \in \mathbb{R}^{\tilde{M} \times \tilde{N}} = \Phi_m \Psi_m$ is the “sensing” matrix. We can write Equation (23) in the form of Equation (24), as below.

$$\underbrace{\begin{bmatrix} y_{m1} \\ y_{m2} \\ \dots \\ y_{m\tilde{M}} \end{bmatrix}}_{Y: \tilde{M} \times 1} = \underbrace{\begin{bmatrix} \phi_{11} & \phi_{11} & \dots & \phi_{1\tilde{N}} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2\tilde{N}} \\ \dots & \dots & \dots & \dots \\ \phi_{\tilde{M}1} & \phi_{\tilde{M}2} & \dots & \phi_{\tilde{M}\tilde{N}} \end{bmatrix}}_{\Phi: \tilde{M} \times \tilde{N}} \underbrace{\begin{bmatrix} x_{m1} \\ x_{m2} \\ \dots \\ x_{m\tilde{N}} \end{bmatrix}}_{\tilde{N} \times 1} + \underbrace{\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \dots \\ \sigma_{\tilde{M}} \end{bmatrix}}_{\tilde{M} \times 1} = \underbrace{\begin{bmatrix} \phi_{11} & \phi_{11} & \dots & \phi_{1\tilde{N}} \\ \phi_{21} & \phi_{22} & \dots & \phi_{2\tilde{N}} \\ \dots & \dots & \dots & \dots \\ \phi_{\tilde{M}1} & \phi_{\tilde{M}2} & \dots & \phi_{\tilde{M}\tilde{N}} \end{bmatrix}}_{\Phi: \tilde{M} \times \tilde{N}} \underbrace{\begin{bmatrix} \psi_{m1} & 0 & \dots & 0 \\ 0 & \psi_{m2} & \dots & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & \dots & \psi_{m\tilde{N}} \end{bmatrix}}_{\Psi: \tilde{N} \times \tilde{N}} \underbrace{\begin{bmatrix} \theta_{m1} \\ \theta_{m2} \\ \dots \\ \theta_{m\tilde{N}} \end{bmatrix}}_{\Theta: \tilde{N} \times 1} + \underbrace{\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \dots \\ \sigma_{\tilde{M}} \end{bmatrix}}_{\tilde{M} \times 1} \quad (24)$$

It can be seen from Equations (15) and (24) that if the K sparse signal X_m is to be accurately reconstructed to obtain $\tilde{X}_m$, the number of measurements $\tilde{M}$ must satisfy the following conditions [6]:

$$\tilde{M} = O(K \log(\tilde{N}/K)) \quad (25)$$

3.2.3. Reconstruction Algorithm

The reconstruction algorithm is the core content of CS. In this section, we will use the solution of the ℓ_1 -norm problem to accurately reconstruct signal X_m to get $\tilde{X}_m^*$.

$$\tilde{X}_m^* = \operatorname{argmin} \|X_m\|_{\ell_1} \text{ s.t. } Y_m = X_m \Theta_m \quad (26)$$

The minimum ℓ_1 -norm problem is a convex optimal problem. By applying the Lagrange multiplier, the optimization problem can be transformed into an unconstrained convex optimization problem,

$$L(X_m) = \|X_m\|_{\ell_1}^2 + \lambda_m (X_m \Theta_m - Y_m) \quad (27)$$

where λ_m is the Lagrange multiplier, satisfying $\lambda_m \geq 0$ at the same time. By using the Lagrange dual constraint decomposition method, a sub-gradient iterative algorithm is constructed to solve the dual problem. The dual function of Equation (27) is shown as

$$D(\lambda_m) = \min L(X_m) \quad (28)$$

The dual problem of Equation (28) can be written as follows:

$$\begin{aligned} & \max D(\lambda_m) \\ & \text{s.t. } \lambda_m \geq 0 \end{aligned} \quad (29)$$

In order to obtain the optimal value $\tilde{X}_m^*$, X_m can be derived according to the Karush-Kuhn-Tucker (KKT) condition with a value of zero.

$$\frac{\partial L(X_m)}{\partial X_m} = 2X_m + \Theta_m^T \lambda_m = 0 \quad (30)$$

Then, we get the optimal solution

$$\tilde{X}_m^* = -\frac{1}{2} \Theta_m^T \lambda_m \quad (31)$$

Here, it is worth noting that for the optimal value $\tilde{X}_m^*$, we use the reconstruction algorithm to get signal $\tilde{X}_m$. In practical applications, there is a certain error between the original signal and the reconstructed signal. Therefore, the quality of the reconstruction algorithm determines the size of the

error. To verify the performance of our proposed reconstruction method, we use the following two evaluation criteria in this paper: the signal-to-noise ratio (SNR) and root mean-squared error (RMSE).

$$SNR = 10 \times \lg \left(\frac{\|X_m\|_2}{\|X_m - \tilde{X}_m\|_2} \right) \quad (32)$$

$$RMSE = \frac{1}{\sqrt{l}} \sum_{i=1}^l \|X_{mi} - \tilde{X}_{mi}\|_2 \quad (33)$$

where X_m is the original sensor data of the m -th cluster, $\tilde{X}_m$ represents the reconstructed value of the m -th cluster, $\|\cdot\|_2$ denotes the minimization, l is the length of the data signal, and X_{mi} and $\tilde{X}_{mi}$ are the original data signal and the reconstructed value of the i -th node in the m -th cluster, respectively. It can be concluded from Equation (32) that the bigger the SNR is, the better performance the algorithm will be.

3.2.4. Algorithm Process

Our proposed algorithm consists of three sections—(1) CH node election, (2) CH node data compression, and (3) sink node reconstruction data. In this way, the algorithm processes for the clustering data fusion algorithm based on CS are described in Algorithm 1, Algorithm 2, and Algorithm 3, respectively. The detailed pseudo-code for the three algorithms are as follows.

3.3. Algorithm Analysis

3.3.1. Algorithm Convergence Analysis

The algorithm convergence and the convergence speed are important indicators for evaluating the performance of the algorithm. The algorithm convergences fast, which not only shortens the CH election time, but also improves data fusion efficiency. In order to analyze the convergence, we will discuss the convergence of the ICL-LEACH algorithm and the CS data fusion algorithm, respectively.

Algorithm 1 ICL-LEACH Algorithm for CH Node Election

```

1:Require:  $N, M, n, r, R, p, G, T_{i-new}, d_{max}, d_{ij}, \lambda_1, \lambda_2, \lambda_3, E_{i-residual}, E_{i-max}, rand_i, type_i = no, CH_i = no, l_{original}, l_{current}$ 
2:Ensure:  $M$ 
3:While  $1 \leq r \leq R$  and  $E_{i-residual} > 0$  do
4: If  $CH_i = no$ 
5:   For  $i = 1$  to  $N$  do
6:     If  $rand_i \leq T_{i-new}$ 
7:    $type_i = CH$  and  $CH_i = yes$ 
8:   CHnode broadcasts the selected message to all the nodes
9:   CH node uses TDMA to allocate time slots and transmits data
10:    Else  $type_i = CM$  and  $CH_i = no$ 
11:    CM node waits for the broadcast information of CH node
12:    CM node waits to allocate a time slot
13:    End if
14:    nodes enter the stabilization phase
15:  End for
16:  Else CH node performs data fusion and data transmission
17: End if
18:  $r \leftarrow r + 1$ 
19:Endwhile

```

Algorithm 2 CH Node Data Compressed Algorithm

```

1: Require:  $X_m \in \mathbb{R}^{\tilde{N} \times 1}$ ,  $x_{mi}$ ,  $Y_m \in \mathbb{R}^{\tilde{M} \times 1}$ ,  $y_{mj}$ ,  $\Phi_m \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$ ,  $\phi_{mji}$ ,  $\sigma_{mj}$ ,  $\Theta_m \in \mathbb{R}^{\tilde{N} \times 1}$ ,  $\theta_{mi}$ ,  $\Psi_m \in \mathbb{R}^{\tilde{N} \times \tilde{N}}$ ,  $\psi_{mii}$ 
2: Ensure:  $Y_m$ 
3: While  $1 \leq m \leq M$  do
4:   For  $j = 1$  to  $\tilde{M}$  do
5:     For  $i = 1$  to  $\tilde{N}$  do
6:        $X_m = \sum_{i=1}^{\tilde{N}} \psi_{mii} \theta_{mi}$ 
7:     End for
8:    $y_{mj} = \langle X_m, \phi_{mji} \rangle + \sigma_{mj}$ 
9:   End for
10:  CH node sends  $Y_m$  to the sink node
11: Endwhile

```

Algorithm 3 Sink Node Reconstruction Signal

```

1: Require:  $X_m$ ,  $x_{mi}$ ,  $Y_m$ ,  $y_{mj}$ ,  $\Phi_m$ ,  $\phi_{mji}$ ,  $\sigma_{mj}$ ,  $\Theta_m$ ,  $\Psi_m$ ,  $\psi_{mii}$ ,  $\xi$ ,  $\hat{\Phi}_m$ ,
2: Ensure:  $\tilde{X}_m$ 
3: When the sink node received  $Y_m$  from the  $m$ -th CH node then
4:   Reconstruct the signal using the CVX tool to solve the  $\ell_1$ -norm minimum
5:   Cvx_begin
6:     variable  $X_m$ 
7:     minimize (norm( $\Psi_m^T X_m, 1$ ))
8:     subject to
9:        $Y_m = \hat{\Phi}_m \Theta_m + \sigma_m$ 
10:   Cvx_end
11:   IF  $|X_m - \tilde{X}_m| > \xi$  then
12:     change the CH nodes based on the new source
13:   Else go to the Algorithm 1
14:   End if
15: End

```

As described above, the key points of our proposed algorithm include clustering, sparse coding, dictionary learning, and reconstruction. Therefore, the discussion of the convergence of the proposed algorithm is actually a discussion of the convergence of the improved clustering algorithm and the CS algorithm. Firstly, the ICL-LEACH algorithm is based on LEACH. The convergence of the LEACH algorithm has been proven in many well-known papers [19,27]. Therefore, the ICL-LEACH algorithm is convergent. Secondly, in the iterative scheme of convex optimization, the convergence speed depends on the distance from the starting point to the optimal solution [25]. To speed up convergence, we take advantage of the overlap between windows and use the starting point. By selecting the starting point, the expected number of iterations of convergence can be reduced, resulting in a good estimate of X_m . We use this formula to prove the convergence of the CS data fusion algorithm. See Appendix B for specific certification procedures. A comprehensive analysis can guarantee the convergence of our proposed algorithm.

3.3.2. Algorithm Time Complexity Analysis

Time complexity is another indicator used to evaluate the efficiency of an algorithm. According to Algorithm 1, Algorithm 2, and Algorithm 3, the time complexity of our proposed algorithm also includes three parts: the clustering algorithm, compression algorithm and reconstruction algorithm. After analysis, the time complexity of the proposed algorithm is as follows.

The time complexity of the clustering process is $O(R \times N)$, the time complexity of the compression algorithm is $O\left(M \times (\tilde{N} + \tilde{M})\right)$, and the time complexity of the reconstruction algorithm is $O(\tilde{M})$, where R is the largest round of clustering, N is the number of sensor nodes, M is the number of clusters, $\tilde{N}$ is the dimension, and $\tilde{M}$ is the dimension. Then, it can be seen that the overall time complexity of the algorithm is $O(R \times N) + O\left(M \times (\tilde{N} + \tilde{M})\right) + O(\tilde{M})$.

4. System Simulation Analysis

In this section, we use a numerical simulation to evaluate the performance of our proposed algorithm ICL-LEACH-CS. As shown in Figure 1, it is assumed that the coverage of the WSN is 100 (m) $\times$ 100 (m). Further 100 wireless sensor nodes are randomly distributed in this area, and a sink node is located at the center of this area. The simulation parameters used in this paper (according to [9,20,21,27,28]) are shown in Table 1. In order to verify the performance of the algorithm, two experimental methods (i.e., experimental environment establishment and MATLAB simulation) are adopted. The ℓ_1 -Magic toolbox and convex optimization toolbox of MATLAB 2015b are used to solve the ℓ_1 -norm optimization [29].

Table 1. Simulation parameters.

Parameter	Value	Parameter	Value
Node number N	6/100	Noise N_m	0.1
Transmission range	50 m	Sparsity K	25
Initial energy E_{initial}	3 J	Measurement $\tilde{M}$	50
Data size l	4000 bit	Carrier frequency f	25 kHz
E_{elec}	50 nJ/bit	Control packet length N	100 bit
E_{amp}	100 pJ/(bit·m ²)	Sampling frequency f_s	100 kHz
E_{fs}	10 pJ/(bit·m ²)	Setup phase: stabilization phase	1:15
E_{df}	5 nJ/bit	Maximum round R	5000
CH selection ratio p	0.05	ξ	$1 \times e^{-9}$

4.1. Experimental Setup

4.1.1. Experimental Environment

To better verify the performance of our proposed algorithm, we built an experimental environment to test whether it could be deployed in a true WSN. We built the experimental environment with the EMIOT-JX-1, which integrates multiple sensor models and multiple wireless networking modes to run multiple internet of things (IoT) network architectures. The experimental environment is shown in the Figure 4. The specific steps to set up the experimental environment are as follows:

- Install the CC-Debug emulator driver;
- Install the CP210X_vcp_win_xp driver for COM 3;
- Install the CC-Debug emulator driver;
- Install the CP210X_vcp_win_xp driver, the simulation is serial port 3;
- Install Setup_SmartRFProgr;
- Install monitoring software ZigbemPC;
- Install the TinyOS development environment program ZigbemDS.



Figure 4. Building an experimental platform.

Without loss of generality, in this part of the experiment we chose six wireless sensor nodes (i.e., Hall Sensor, Medical Pressure Sensor, Air Pressure Sensor, Fire Sensor, Temperature Sensor, Illumination Sensor) for research. The results of this study can also be extended to the scenarios of multiple wireless sensor nodes. The WSN experiment requires the construction of the TinyOS development environment. TinyOS is an open source IoT operating system that supports CC2430 chip, supports ultra-low power RF transmission, supports mobile independent network, and supports Zigbee protocol. The main development tool is ZigbemDS, which installation package automatically installs rich components into the TinyOS development environment for developers to use.

4.1.2. Experimental Process

We use a CC2430 module to burn the program into the gateway board node and the six sensor nodes, respectively. As shown in Figure 5, we open the code compiler Notepad++ for ZigbemDS (DonHO, Taiwan, China) to compile our program.

Open the ZigbemPC program to observe the operation of the WSN. ZigbemPC is EMIOT-JX-1 PC terminal control software, which is mainly used for map monitoring, display of sensor instant data, and display of network topology. As shown in Figure 6, a six-node WSN topology can be seen in the ZigbemPC program. Node 4 (i.e., FireSensor) is directly connected to the sink node, and node 7 (i.e., IlluminationSensor), node 2 (i.e., HallSensor), node 3 (i.e., TemperatureSensor), node 5 (i.e., Air PressureSensor, FireSensor), and node 6 (i.e., Medical PressureSensor) form a cluster. Node 7 is a CH node and is directly connected to the sink node. The other four nodes send information to node 7.

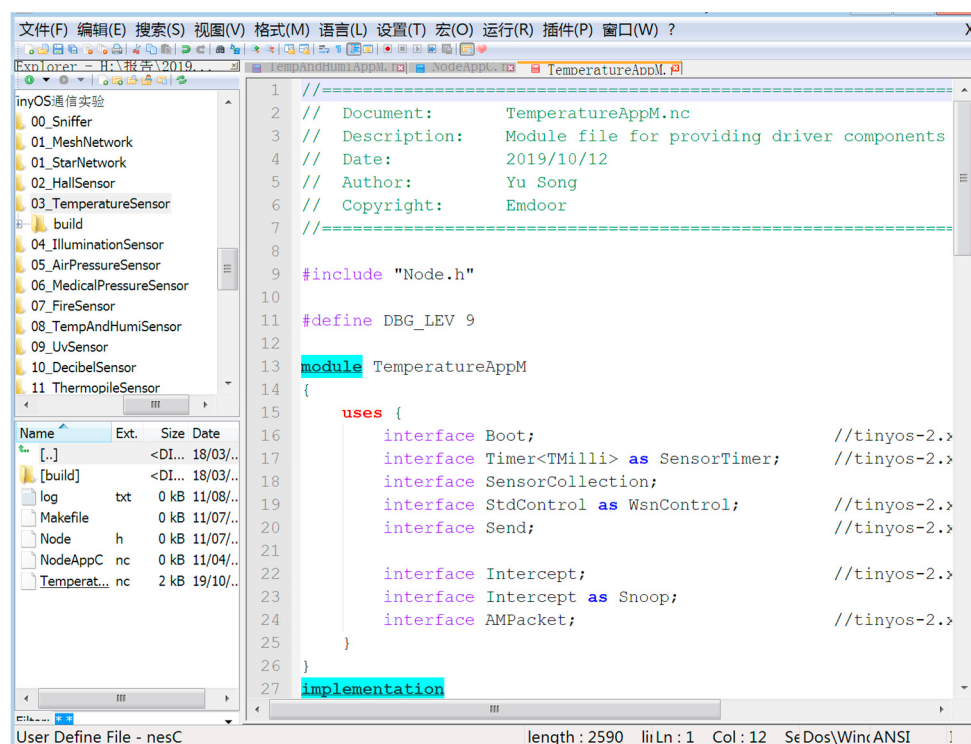


Figure 5. Compile the program.

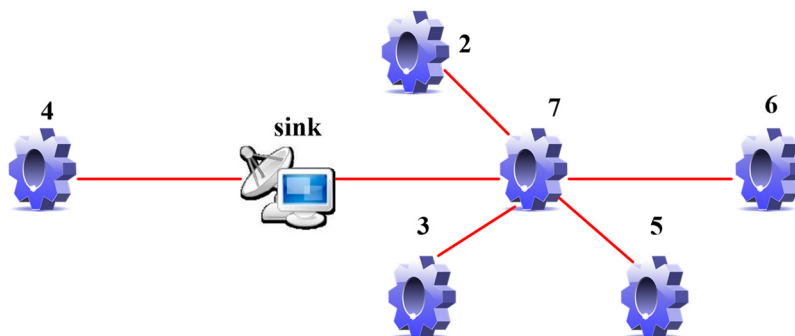


Figure 6. A topology of six wireless sensor network (WSN) sensor nodes.

As shown in Figure 7, we can get the change of data packets between the six nodes by observing ZigBeePC. At the same time, the source node, the relay node, the number of hops the data of different sensor nodes, and the original data can all be seen. Instant data is comprehensive and can be analyzed with our results.

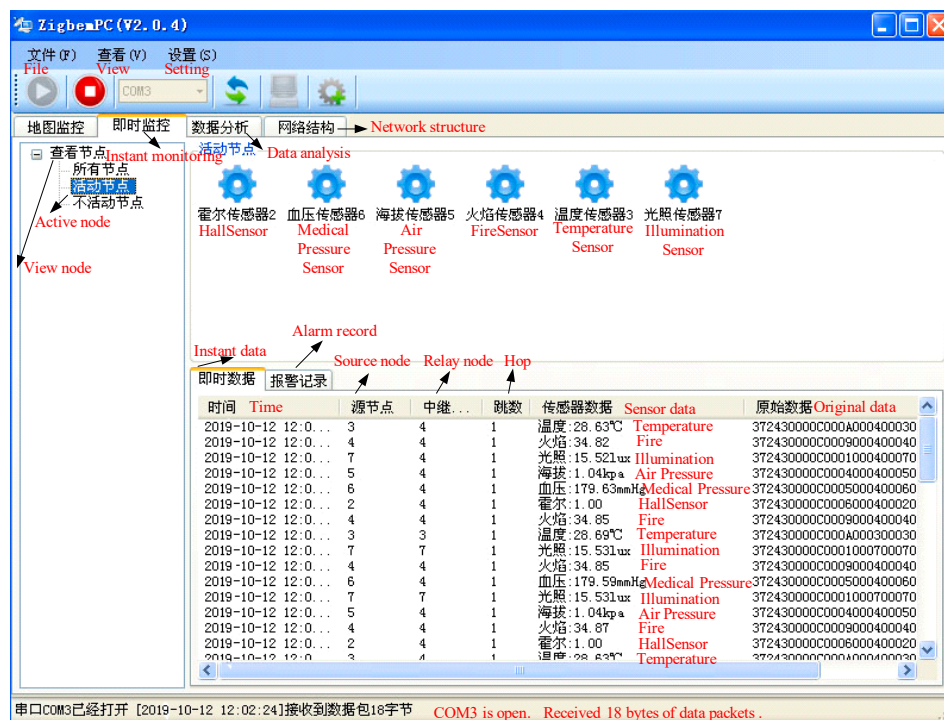


Figure 7. ZigbemPC monitoring data.

Taking node 3 (i.e., Temperature Sensor) as an example, we perform data analysis on it. Through ZigbemPC, we can see the temperature change in node 3 during the observation time. The data chart of node 3 is shown in Figure 8. We burn the algorithm into node 3, and node 3 can run normally, which can prove the correctness of our algorithm.

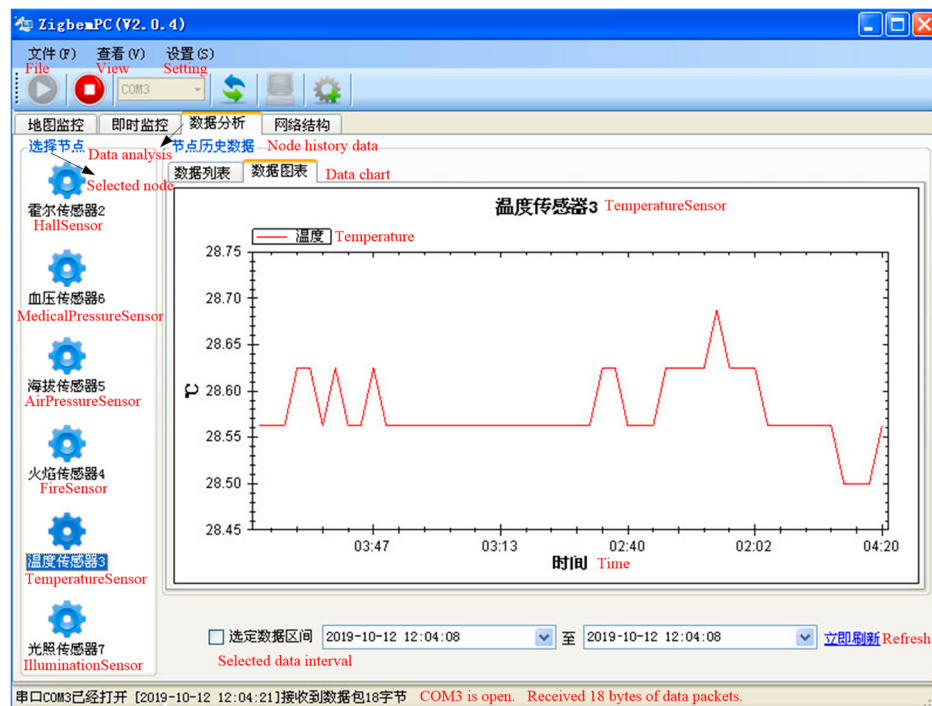


Figure 8. The data chart for node 3.

Through the compilation and operation of the program in the COM debugging assistant, we can see the basic functions of the ICL-LEACH-CS algorithm, i.e., clustering and transmission. The program is then burned into the sensor node. We use the monitoring software ZigBemPC to see how the program is operating in the sensor node. There are many packets between the sensor nodes, and the size of the packets sent from the source to the CH node is normal. The experimental results verify the correctness of our proposed algorithm ICL-LEACH-CS.

In order to evaluate the performance of the ICL-LEACH-CS algorithm, we will carry out simulation analysis in two parts—a clustering algorithm performance analysis and a CS data fusion performance analysis. The specific performance analysis is as follows.

4.2. Performance Evaluation

4.2.1. ICEL-LEACH-CS Algorithm Clustering Process

The simulation scenarios in this paper are set as follows:

- The sensor nodes are randomly distributed in a square area of 100 (m) $\times$ 100 (m);
- WSN is a homogeneous network with each sensor node having the same function and a unique number;
- The sensor node energy is limited, and it does not have an energy harvesting function;
- After the sensor node is deployed, its location is fixed;
- The sink node is located in the center of the area, i.e., the coordinates of the sink node are (50, 50), and the location is also fixed;
- The node communicates with the CH node through a single hop, and the CH node communicates with the sink node in a single hop or multi-hop manner.

Specific parameters are shown in Table 1. As can be seen from Figure 9, the distribution of the nodes is not very uniform. At the same time, it also can be seen that, by running the ICL-LEACH-CS algorithm, the 100 wireless sensor nodes are divided into nine clusters, including nine CH nodes. Each CM node directly communicates with the CH node, and the CH node communicates with the sink node through the CS data fusion. In this way, redundant data transmission can be reduced, the residual energy of the nodes can be saved, and the lifetime of the WSN network can be improved [30].

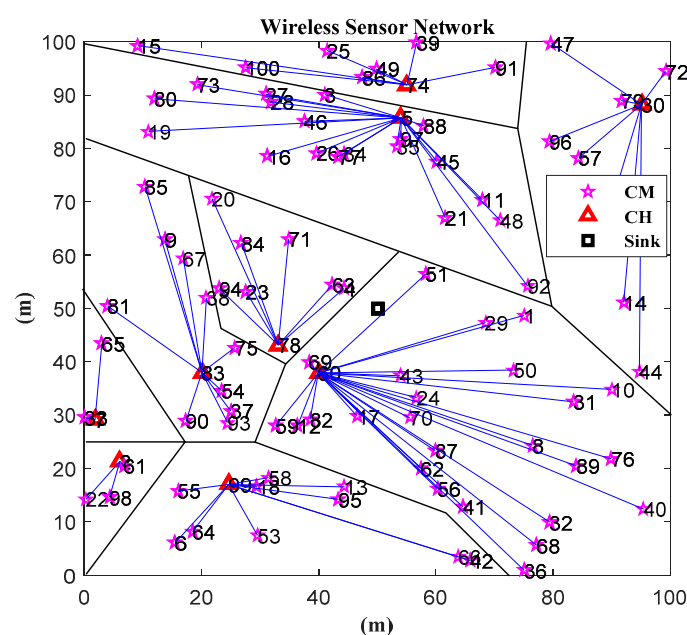


Figure 9. WSN topology and ICL-LEACH-CS algorithm clustering process.

4.2.2. Comparison with Five-Type Cluster Strategy

In order to better and more accurately evaluate the advantages and disadvantages of our proposed ICL-LEACH-CS algorithm, we compare it with LEACH [27], LEACH-C [19], DHLCS [25], LEACH-FL [31], and LEACH-GA [32] from the WSN's lifetime and residual energy. Among them, LEACH-C is an improved algorithm for LEACH. LEACH-C extends the random CH selection algorithm of LEACH via deterministic components. The authors of reference [25] proposed a WSN routing protocol (DHLCS) based on an alternate dual CH mechanism and compressive sensing. LEACH-FL is an improved algorithm of LEACH using fuzzy logic (FL) [31]. The mechanism is similar to LCH, but the variables (parameters) used are power, distance, and node density. Experiments show that the LEACH-FL algorithm has a lower energy consumption rate and longer network's lifetime than the LEACH algorithm. LEACH-GA is another improved algorithm of LEACH using genetic algorithm (GA) [32]. In each round, the LEACH-GA has three phases—set-up, steady-state, and preparation. The preparation phase is only performed once before the first round of set-up phase. The set-up and steady-state process for each round is the same as LEACH.

As shown in Figure 10, we compare ICL-LEACH-CS with five other algorithms (i.e., LEACH-C, LEACH-CS, DHLCS, LEACH-FL, and LEACH-GA) to see how the number of surviving sensor nodes in the WSN varies with the number of rounds. As can be seen from Figure 10, the maximum number of rounds is 5000, and the number of network nodes is 100. When the number of rounds is about 1250, the number of surviving nodes of the six algorithms is 100. After this round, the number of surviving nodes of the LEACH algorithm begins to decline. The LEACH-C algorithm, DHLCS algorithm, ICL-LEACH-CS, LEACH-FL, and LEACH-GA algorithms have little difference in performance. However, after 1900 rounds, the number of surviving nodes of the five algorithms also began to decline. It can be seen that DHLCS can balance CH energy consumption by electing double CH nodes. LEACH-GA uses the GA search node to become the best probability of the CH, and the change is also slow by minimizing the total energy consumption required to complete one round of the sensor field. Since LEACH-C and LEACH-FL do not consider data fusion, the energy consumption is relatively large. Thus, the surviving nodes of DHLCS change slowly. Considering the residual energy, distance, and compression ratio, our algorithm has a slower change than the DHLCS algorithm. When the number of rounds is 4000, there are about 51 surviving nodes in our algorithm. Therefore, our algorithm can extend the lifetime of a WSN.

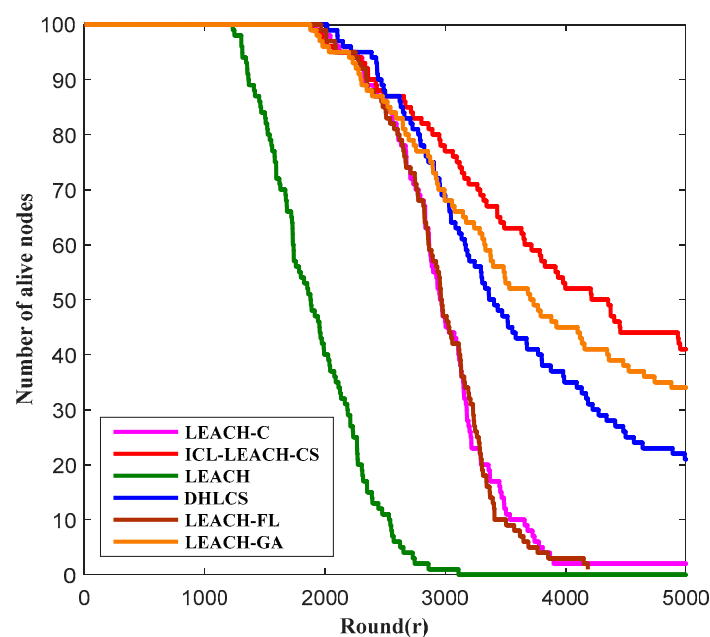


Figure 10. Number of alive nodes vs. round.

The election of the CH nodes in the WSN is very important. Too many or too few CH nodes will cause the energy consumption of the nodes to be too fast. Figure 11 represents the average residual energy of the node as a function of the number of rounds. It can be seen from Figure 11 that LEACH has the least residual energy, followed by LEACH-FL and LEACH-C. Our algorithm ICL-LEACH-CS has the most residual energy. The main reason is that the basic idea of LEACH is to select the CH node by random loop, and the number of CH nodes cannot be guaranteed, so the energy consumption of the algorithm is very fast. However, LEACH-FL is a centralized algorithm that does not have data fusion capabilities. Therefore, it has high energy consumption. LEACH-C has made some improvements to the clustering algorithm, which is no longer the primary way to randomly select CH nodes. The LEACH-C protocol uses nodes with energy no lower than the average energy value as candidate nodes. This method can reduce the energy consumed by communication when electing CH nodes, so more residual energy can be used to transmit data. Although LEACH-GA and DHLCS can achieve better clustering performance, the size distribution of each cluster is more uniform, and the network energy consumption is balanced. However, in terms of the number of clusters per round, our algorithm ICL-LEACH-CS is better than LEACH-GA and DHLCS. Moreover, the data transmission rate of LEACH-GA cannot be estimated [30]. And when using multi-hop transmission, DHLCS does not consider the hotspot near the sink, resulting in a decrease in the residual energy of the network. The ICL-LEACH-CS algorithm proposed in this paper selects the CH node based on the residual energy and communication distance of the node. Therefore, the energy consumption of this algorithm is the slowest, and the residual energy of the nodes is greatly improved [30].

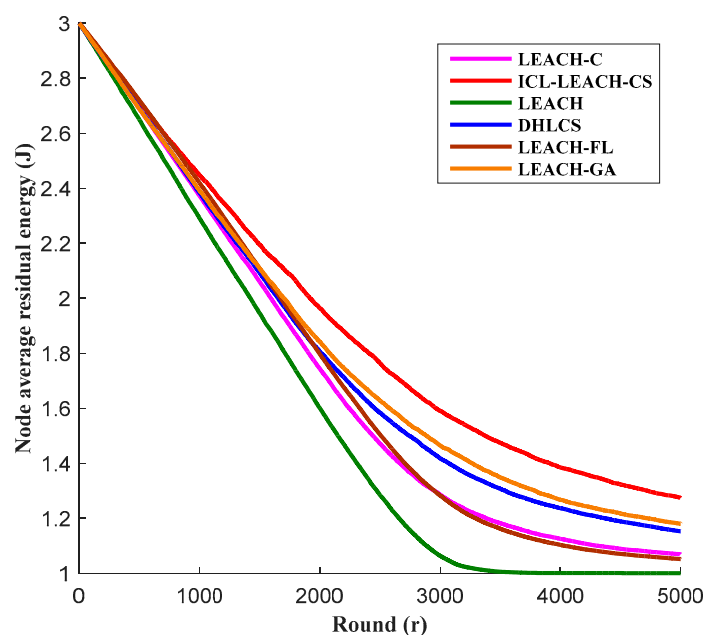


Figure 11. Node average residual energy vs. round.

4.2.3. ICEL-LEACH-CS Algorithm Reconstruction Process

In order to verify the reconstruction performance of our CS algorithm, we will conduct two sets of simulation experiments. The first set of experiments is proof of the recovery performance of our CS algorithm. The second set of experiments is a comparison between different CS algorithms.

We consider the one-dimensional signal, as shown in Figure 12, i.e., $x(t) = \cos(26\pi t) + \cos(10\pi t) + \cos(4\pi t)$. In the actual application scenario, the original signal always has noise. As can be seen from Figure 12, we performed CS on the original signal containing noise. First, the original signal is randomly sampled by the sparsity $K = 25$ to obtain discrete samples of the signal. Then, using the ℓ_1 , the optimization problem is transformed into convex optimization for reconstruction, and the

reconstructed signal is obtained. However, there is an error between the original real signal and the recovery signal because of noise. According to Equation (33), the $RMSE = 6.0641 \times e^{-14}$ can be obtained and the value of $RMSE$ is very small. Here, we assume that the signal recovery is successful if the $RMSE$ is less than $1 \times e^{-9}$. It can be seen that our proposed CS algorithm has strong signal reconstruction capabilities.

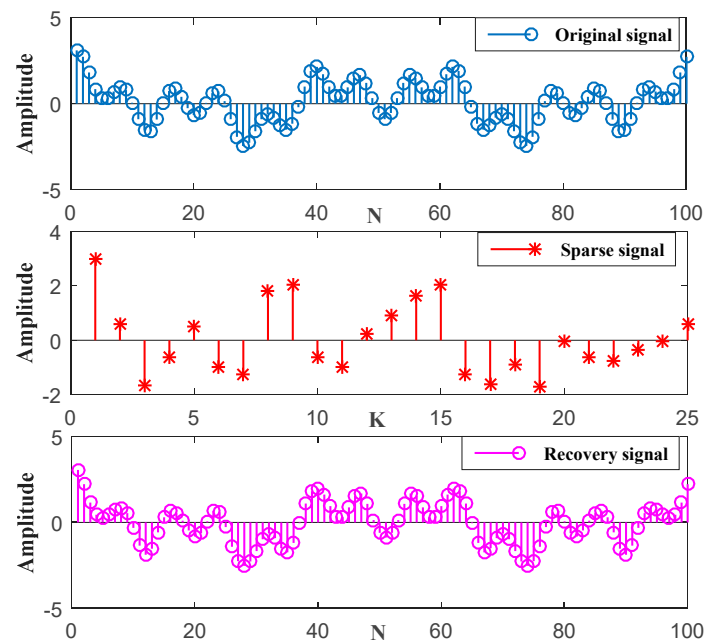


Figure 12. Original signal, sparse signal, and recovery signal.

In general, wireless channel noise is interference for a useful signal. If the noise interference is not removed, the signal will be distorted, and the communication will not be performed correctly and effectively. In order to verify the reconstruction performance of our proposed CS algorithm, we have added the following set of experiments. The original signal is divided into an original signal without noise and an original signal with noise. The recovery signal is divided into a recovery signal without noise and a recovery signal with noise. The noise value is 0.1. As can be seen from Figure 13, at the inflection point of the waveform, the influence of noise on the signal is very obvious. When reconstructing the original signal without noise, the error value is very small, and almost all information can be completely reconstructed. When considering the reconstruction of a noisy original signal, the error value is slightly larger but is within a controllable range. Therefore, our algorithm can efficiently reconstruct the original information in both noisy and noise-free environments. It can effectively reduce the amount of data being transmitted and also improve the accuracy of data transmission.

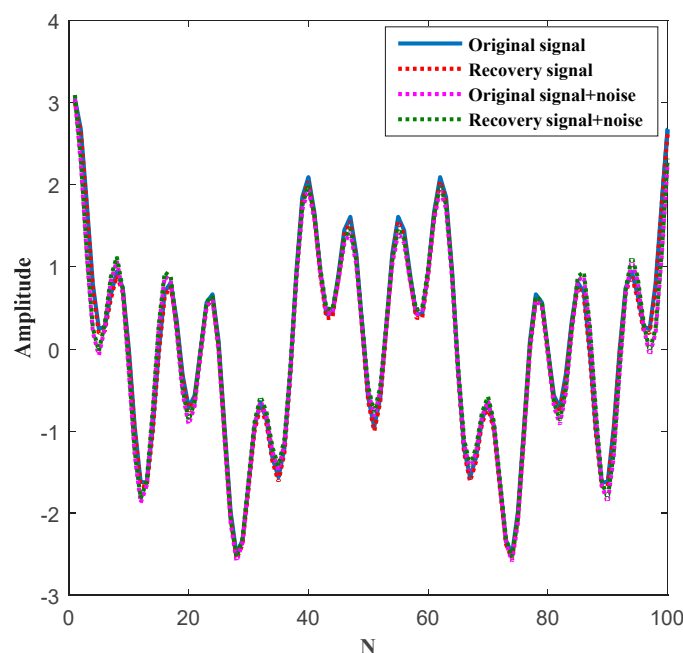


Figure 13. The influence of noise on the original signal and recovery signal.

4.2.4. Comparison with Five-Type Data Fusion Strategy

In WSNs, as network size increases, data fusion and data transmission cannot be ignored. The use of clustering and CS in data fusion can reduce the amount of data being transmitted and balance the traffic load of the entire network. In order to evaluate the performance of our proposed ICL-LEACH-CS algorithm, we intend to compare it with existing data fusion algorithms. These comparative data fusion strategies include the method based on CS and cluster [23], integration of the cluster and CS [24], CS data fusion without cluster [26], cluster data fusion without CS [28], and cluster with hybrid CS [21].

Figure 14 describes the energy consumption of the network as a function of time. In WSNs, network energy consumption is mainly derived from CM nodes and CH nodes. The energy consumption of the CM node is mainly used for data collection and transmission, and the energy consumption of the CH node is mainly used for data collection, receiving data from the CM node, data fusion processing, and transmitting the fused data to the sink node. Therefore, the data fusion algorithm has an impact on energy consumption. As can be seen from Figure 14, the method based on CS and cluster, integration of the cluster and CS, the cluster with hybrid CS, CS data fusion with cluster, and ICL-LEACH-CS all utilize CS, so the sensor node actually transmits the measured value of the node-aware data. These five algorithms convert high-dimensional data into low-dimensional data, and the transmitted data dimension is much lower than the original node-aware data. Therefore, the energy consumption of the sensor node is much lower than that of the cluster data fusion without CS. Our proposed ICL-LEACH-CS algorithm improves the CH election, so the energy consumption is the lowest. It can be seen that the ICL-LEACH-CS algorithm proposed in this paper is very suitable for a WSN with limited energy.

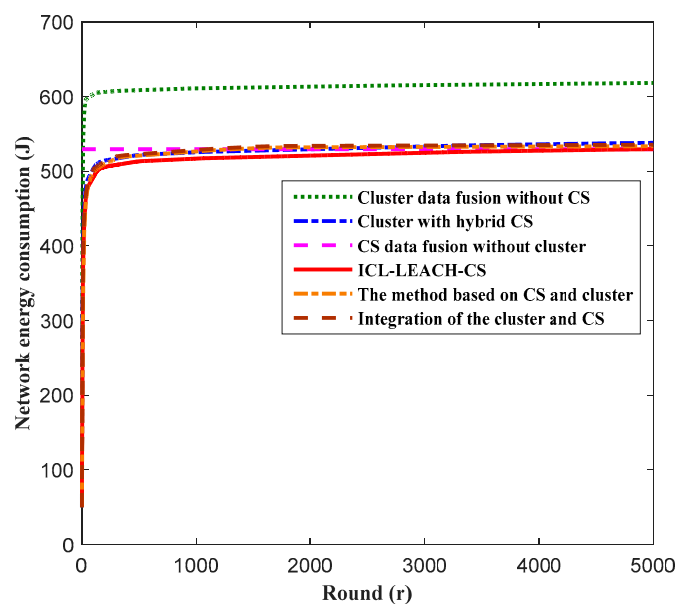


Figure 14. Network energy consumption vs. round.

The lifetime of the WSN affects the amount of data received by the sink node. The longer the lifetime is, the more data the sink node receives. Figure 15 illustrates the number of packets received by the sink nodes of the six algorithms (i.e., the method based on CS and cluster, integration of the cluster and CS, cluster data fusion without CS, cluster with hybrid CS, CS data fusion without cluster, and ICL-LEACH-CS) as the number of rounds increases. It can be seen that, with an increase of rounds, the amount of data received by the sink node also increases. When the round is 2000, our proposed ICL-LEACH-CS algorithm outperforms the other five algorithms (i.e., the method based on CS and cluster, integration of the cluster and CS, cluster data fusion without CS, cluster with hybrid CS, and CS data fusion without cluster). The main reason for this result is that we consider the residual energy and distance. In the CH election, the nodes with lower residual energy are avoided as the CH, so the transmission capacity of the nodes is increased.

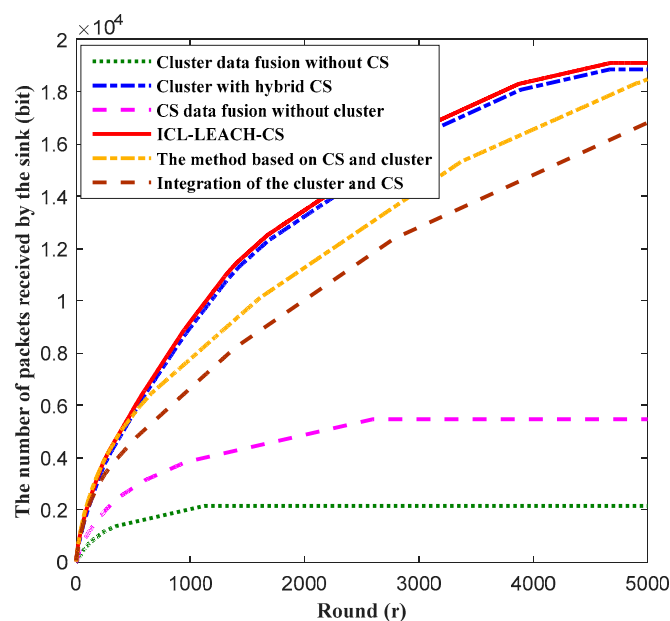


Figure 15. Data transmission at the sink node vs. round.

In WSNs, the network load balance refers to the balance load (i.e., task or energy consumption) of multiple wireless sensor nodes to coordinate tasks or reduce energy consumption. The clustering algorithm, multi-hop routing algorithm, and data fusion algorithm can solve the problem of the network load balance. However, the traditional clustering algorithm and data fusion algorithm can easily form a “hot zone” phenomenon and an “energy hole” problem. The “hot zone” phenomenon, or “energy hole,” refers to the fact that the node that is closer to the sink node has a larger amount of data to be forwarded, resulting in the energy consumption of the CH node near the sink node being too fast and dying. Figure 16 depicts the load balance change of a node as time increases. In Figure 16, it is clear that the proposed ICL-LEACH-CS load balance is better. This occurs because we have improved the clustering algorithm, considering the distance, residual energy and compression ratio of the nodes. When the CH is elected, avoiding being close to the sink node is always elected as the CH. Multi-hop routing can be performed between the CH nodes, and the distant CH transmits data to the CH node that is close to the sink. The energy consumption of the entire network is, therefore, greatly reduced, the network transmission rate is accelerated, and the network lifetime is improved.

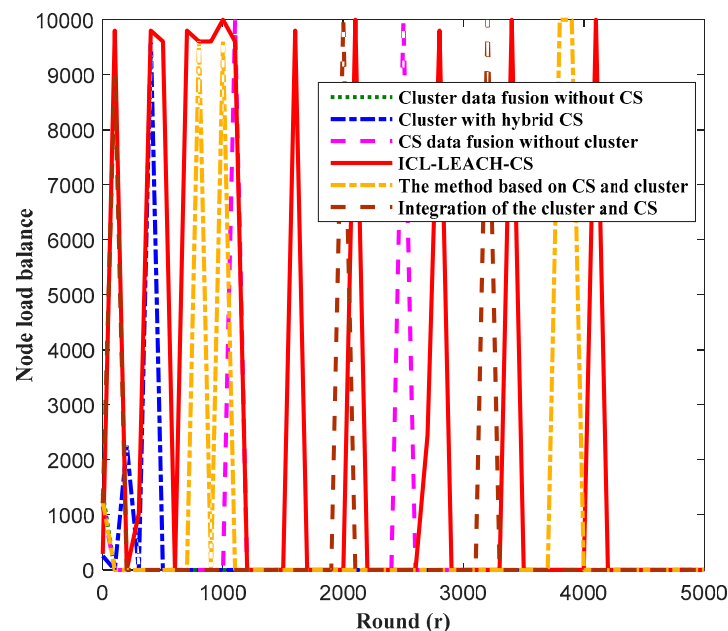


Figure 16. Node load balance vs. round.

In addition, to improve the accuracy and credibility of data fusion in order to get a more accurate representation or estimation of the target, we need to analyze the correct rate of signal reconstruction. We compare the ICL-LEACH-CS algorithm with four other CS-based data fusion algorithms (i.e., the method based on CS and cluster, integration of the cluster and CS, cluster with hybrid CS, and CS data fusion without cluster). Figure 17 reveals the signal reconstruction accuracy as a function of the sparseness level (K/N), where K is the sparsity, and N is the signal length. When the sparse level is 0.31, the reconstruction accuracy of the five algorithms (i.e., the method based on CS and cluster, integration of the cluster and CS, ICL-LEACH-CS, cluster with hybrid CS, and CS data fusion without cluster) is 100%. It can be seen that the CS data fusion without cluster algorithm fails when the sparse level is 0.31, but the other four algorithms (the method based on CS and cluster, integration of the cluster and CS, ICL-LEACH-CS, and cluster with hybrid CS) still function. When the sparsity reached 0.385, the Cluster with hybrid CS algorithm also started to fail. However, ICL-LEACH-CS continues until $K/N = 0.5$, which is almost equal to half of the measured amount. This also shows that the proposed algorithm’s reconstruction accuracy is relatively high.

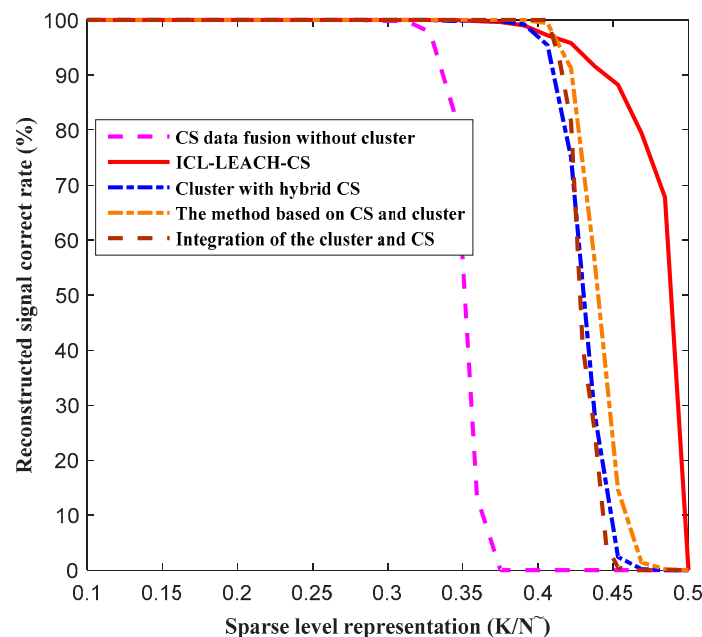


Figure 17. Reconstructed signal correct rate vs. the sparse level representation.

5. Conclusions

In this paper, we studied a data fusion scheme for WSNs. Unlike the traditional clustering data fusion strategy, we considered a combination of clustering and compressed sensing (CS). In this way, we investigated the data fusion mechanism based on the improved LEACH clustering algorithm and CS, which we named the improved clustered LEACH and CS (ICL-LEACH-CS) algorithm. As a new data processing method, CS can greatly reduce the amount of data generated by the sampling process. CS technology is used in the WSN clustering algorithm to compress the data monitored by the node and then transmit it into the network, which can greatly reduce the energy consumption. Since the LEACH algorithm has high randomness, we consider the distance, residual energy, and compression ratio and propose an improved LEACH algorithm called ICL-LEACH. The ICL-LEACH algorithm is used to select the optimal cluster head (CH) node. The cluster member (CM) node sends the data to the CH node. The CH reuses the CS to compress the data and finally sends the compressed data to the sink node. The sink node reconstructs the data and restores the original information. When sparse representation of the signal, we choose the classical Fourier transform (FFT) sparse method. The method has the advantages of good stability and low computational complexity. When designing the measurement matrix, we choose an independent and identically distributed Gaussian random matrix as the observation matrix. We also prove that the sparse basis and the observation base are irrelevant, and the RIP is guaranteed. When designing the signal recovery algorithm, we translate the optimization problem into an optimization problem under the ℓ_1 minimum norm and use the sub-gradient iterative algorithm to get accurate results. In addition, we analyzed the time complexity of the ICL-LEACH-CS and also proved its convergence. The simulation results show that the ICL-LEACH-CS algorithm greatly reduces the energy consumption of a WSN node compared to the traditional clustering algorithm, prolongs the network's lifetime, and balances the network load of the node. These results give further insight into ways to process data fusion in WSNs in the future.

Author Contributions: Conceptualization, Y.S.; methodology, Y.S., Z.L., and X.H.; formal analysis, Y.S.; writing—original draft preparation, Y.S.; writing—review and editing, X.H. and H.J.; supervision, Z.L.

Funding: This work was partially supported by the National Natural Science Foundation of China (No. 61771410), by the Post graduate Innovation Fund Project by South west University of Science and Technology (No. 19ycx0106), by the Artificial Intelligence Key Laboratory of Sichuan Province (No. 2017RYY05, No. 2018RYJ03), and by the Horizontal Project (No. HX2017134, No. HX2018264, No. E10203788).

Acknowledgments: The authors would like to thank the anonymous reviewers for their selfless reviews and valuable comments, which have improved the quality of our original manuscript.

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

Appendix A

Definition 1. For the K sparse signal X , where K is a isnormal and $K = 1, 2 \dots$, if the minimum value of the isometry constant $\delta_K \in (0, 1)$ satisfies Equation (A1)

$$(1 - \delta_K)\|X\|_2^2 \leq \|\Phi_T X\| \leq (1 + \delta_K)\|X\|_2^2 \quad (\text{A1})$$

then the $\tilde{M} \times \tilde{N}$ matrix Φ satisfies K order RIP [6].

The RIP of order $2K$ implies that for any K sparse signal X , if the measurement matrix Φ satisfies the $2K$ order RIP

$$(1 - \delta_{2K})\|X\|_2^2 \leq \|\Phi_T X\| \leq (1 + \delta_{2K})\|X\|_2^2 \quad (\text{A2})$$

then $Y = \Phi X$ has a unique solution.

The RIP ensures that the measurement matrix does not map two different sparse signals into the same set (i.e., it can guarantee one-to-one mapping between the original space and the sparse space).

Appendix B

As discussed before, the measurement vector is denoted as $y_m = \phi_m x_m + \sigma_m$, where $x_m \in X_m$, $\phi_m \in \Phi_m \in \mathbb{R}^{\tilde{M} \times \tilde{N}}$, and σ_m are the Gaussian noise. Let $\tilde{x}_m^*$ denote the reconstruction value of the $t + 1$ update, and the expression is as follows:

$$x_m = (y_m - \sigma_m)\phi_m^T \quad (\text{A3})$$

$$x_m^t = (y_m^t - \sigma_m^t)(\phi_m^t)^T \quad (\text{A4})$$

Additionally, the error between the reconstructed signal and the original signal is $\Delta x = |\tilde{x}_m^* - x_m|$, as shown below:

$$\begin{aligned} \tilde{x}_m^* &= x_m^{t+1} \\ &= x_m^t + \Delta x^t \\ &= (y_m^t - \sigma_m^t)(\phi_m^t)^T + \Delta x^t \end{aligned} \quad (\text{A5})$$

where $\tilde{x}_m^*$ represents the value before the update and Δx^t is the error of the t -th update.

$$\begin{aligned} \Delta x &= |\tilde{x}_m^* - x_m^t| \\ &= \left| \left((y_m^t - \sigma_m^t)\phi_m^T + \Delta x^t \right) - \left((y_m^t - \sigma_m^t)(\phi_m^t)^T \right) \right| \\ &= |\Delta x^t| \end{aligned} \quad (\text{A6})$$

As the number of iterations t increases, the error between the optimal solution obtained by the convex optimization and the original signal becomes smaller and smaller. When $t \rightarrow \infty$, $\Delta x^t \rightarrow 0$, so x_m converges to $\tilde{x}_m^*$.

References

1. Maschi, L.; Pinto, A.; Meneguetto, R.; Baldassin, A. Data summarization in the node by parameters (DSNP): Local data fusion in an IoT environment. *Sensors* **2018**, *18*, 799. [CrossRef] [PubMed]
2. Izadi, D.; Abawajy, J.H.; Ghanavati, S.; Herawan, T. A data fusion method in wireless sensor networks. *Sensors* **2015**, *15*, 2964–2979. [CrossRef] [PubMed]

3. Kobo, H.I.; Abu-Mahfouz, A.M.; Hancke, G.P. A survey on software-defined wireless sensor networks: Challenges and design requirements. *IEEE Access* **2017**, *5*, 1872–1899. [[CrossRef](#)]
4. Cheng, C.T.; Leung, H.; Maupin, P. A delay-aware network structure for wireless sensor networks with in-network data fusion. *IEEE Sens. J.* **2013**, *13*, 1622–1630. [[CrossRef](#)]
5. Donoho D, L. Compressed sensing. *IEEE Trans. Inf. Theory* **2006**, *52*, 1289–1306. [[CrossRef](#)]
6. Candès, E.J.; Romberg, J.; Tao, T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inf. Theory* **2006**, *52*, 489–509. [[CrossRef](#)]
7. Bai, X.; Wang, Z.; Sheng, L.; Wang, Z. Reliable data fusion of hierarchical wireless sensor networks with asynchronous measurement for greenhouse monitoring. *IEEE Trans. Control Syst. Technol.* **2018**, *99*, 1–11. [[CrossRef](#)]
8. Luo, H.; Luo, J.; Liu, Y.; Das, S.K. Adaptive data fusion for energy efficient routing in wireless sensor networks. *IEEE Trans. Comput.* **2006**, *55*, 1286–1299.
9. Venkatesh, V.; Raj, P.; Balakrishnan, P. An energy-efficient fuzzy based data fusion and tree based clustering algorithm for wireless sensor networks. In *The International Symposium on Intelligent Systems Technologies and Applications*; Springer: Cham, Switzerland, 2018; Volume 683, pp. 14–27.
10. Soltani, M.; Hempel, M.; Sharif, H. Data fusion utilization for optimizing large-scale Wireless Sensor Networks. In Proceedings of the 2014 IEEE International Conference on Communications (ICC), Sydney, NSW, Australia, 10–14 June 2014.
11. Li, X.; Liu, W.; Xie, M.; Liu, A.; Zhao, M.; Xiong, N.; Zhao, M.; Dai, W. Differentiated data aggregation routing scheme for energy conserving and delay sensitive wireless sensor networks. *Sensors* **2018**, *18*, 2349. [[CrossRef](#)]
12. Xu, J.; Yang, G.; Chen, Z.; Wang, Q. A survey on the privacy-preserving data aggregation in wireless sensor networks. *China Commun.* **2015**, *12*, 162–180. [[CrossRef](#)]
13. Yao, Y.; Liu, J.; Xiong, N. Privacy-preserving data aggregation in two-tiered wireless sensor networks with mobile nodes. *Sensors* **2014**, *14*, 21174–21194. [[CrossRef](#)] [[PubMed](#)]
14. Vinodha, D.; Anita, E.A.M. A survey on privacy preserving data aggregation in wireless sensor networks. In Proceedings of the 2017 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, 23–24 February 2017.
15. Wang, T.; Tan, J.; Ding, W.; Zhang, Y.; Yang, F.; Song, J.; Han, Z. Inter-Community Detection Scheme for Social Internet of Things: A Compressive Sensing Over Graphs Approach. *IEEE Internet Things J.* **2018**, *5*, 4550–4557. [[CrossRef](#)]
16. Wang, M.; Yang, S.; Liu, Z.; Li, Z. Collaborative Compressive Radar Imaging with Saliency Priors. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 1245–1255. [[CrossRef](#)]
17. Gupta, V.; Kailkhura, B.; Wimalajeewa, T.; Liu, S.; Varshney, P.K. Joint sparsity pattern recovery with 1-bit compressive sensing in distributed sensor networks. *IEEE Trans. Signal Inf. Process. Over Netw.* **2019**, *5*, 15–30.
18. Zayyani, H.; Korki, M.; Marvasti, F. Dictionary Learning for Blind One Bit Compressed Sensing. *IEEE Signal Process. Lett.* **2015**, *23*, 187–191. [[CrossRef](#)]
19. Heinzelman, W.R.; Chandrakasan, A. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Trans Wirel. Commun.* **2002**, *1*, 660–670. [[CrossRef](#)]
20. Wang, D.; Xu, R.; Hu, X.; Su, W. Energy-Efficient Distributed Compressed Sensing Data Aggregation for Cluster-Based Underwater Acoustic Sensor Networks. *Int. J. Distrib. Sens. Netw.* **2016**, *2016*, 1–14. [[CrossRef](#)]
21. Xie, R.; Jia, X. Transmission-Efficient Clustering Method for Wireless Sensor Networks Using Compressive Sensing. *IEEE Trans. Parallel Distrib. Syst.* **2014**, *25*, 806–815.
22. Wang, Q.; Lin, D.; Yang, P.; Zhang, Z. An Energy-Efficient Compressive Sensing-Based Clustering Routing Protocol for WSNs. *IEEE Sens. J.* **2019**, *19*, 3950–3960. [[CrossRef](#)]
23. Zhang, D.G.; Zhang, T.; Zhang, J.; Dong, Y.; Zhang, X.D. A kind of effective data aggregating method based on compressive sensing for wireless sensor network. *EURASIP J. Wirel. Commun. Netw.* **2018**, *159*, 1–15. [[CrossRef](#)]
24. Nguyen, M.T.; Teague, K.A. Compressive Sensing Based Data Gathering in Clustered Wireless Sensor Networks. In Proceedings of the 2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), Marina Del Rey, CA, USA, 26–28 May 2014; pp. 187–192.

25. Ju, Y.; Yan, J.; Xu, H. Performance Optimization Based on Compressive Sensing for Wireless Sensor Networks. *Wirel. Pers. Commun.* **2017**, *95*, 1927–1941.
26. Tian, X. Research on Data Fusion Technology Based on Compressed Sensing. *Int. J. Online Eng. (IJOE)* **2016**, *12*, 13–18. [[CrossRef](#)]
27. Handy, M.J.; Haase, M.; Timmermann, D. Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection. In Proceedings of the 4th International Workshop on Mobile and Wireless Communications Network, Stockholm, Sweden, 9–11 September 2002.
28. Yue, J.; Zhang, W.; Xiao, W.; Tang, D.; Tang, J. A Novel Cluster-Based Data Fusion Algorithm for Wireless Sensor Networks. In Proceedings of the 2011 7th International Conference on Wireless Communications, Networking and Mobile Computing, Wuhan, China, 23–25 September 2011.
29. Li, X.; Tao, X.; Mao, G. Unbalanced Expander based Compressive Data Gathering in Clustered Wireless Sensor Networks. *IEEE Access* **2017**, *5*, 7553–7566. [[CrossRef](#)]
30. Wohwe Sambo, D.; Yenke, B.O.; Förster, A.; Dayang, P. Optimized clustering algorithms for large wireless sensor networks: A review. *Sensors* **2019**, *19*, 322. [[CrossRef](#)] [[PubMed](#)]
31. Ran, G.; Zhang, H.; Shulan, G. Improving on LEACH Protocol of Wireless Sensor using Fuzzy Logic. *J. Inf. Comput. Sci.* **2010**, *3*, 767–775.
32. Liu, J.L.; Ravishankar, C.V. LEACH-GA: Genetic Algorithm-Based Energy-Efficient Adaptive Clustering Protocol for Wireless Sensor Networks. *Int. J. Mach. Learn. Comput.* **2011**, *1*, 79–85. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).