Hindawi
Complexity
Volume 2020, Article ID 6467891, 17 pages
https://doi.org/10.1155/2020/6467891

Research Article

Efficient Data Collection Method in Sensor Networks

Keyan Cao, [1,2] Haoli Liu, [1] Yefan Liu, [1] Gongjie Meng, [1] Si Ji, [3] and Gui Li [1]

[1] *Shenyang Jianzhu University, Liaoning, Shenyang 110168, China*
[2] *Liaoning Province Big Data Management and Analysis Laboratory of Urban Construction, Shenyang 110168, China*
[3] *Northeastern University, Liaoning, Shenyang 110819, China*

Correspondence should be addressed to Keyan Cao; caokeyan@sjzu.edu.cn

Received 22 January 2020; Accepted 27 February 2020; Published 10 April 2020

Guest Editor: Xuyun Zhang

Wireless sensor networks are widely used in many fields, such as medical and health care, military monitoring, target tracking, and people's life, because of their advantages of convenient deployment, low cost, and good concealment. However, due to the low battery capacity of sensor nodes and environmental changes, the energy consumption of nodes is serious and the accuracy of data collection is low. In the data collection method of multiple random paths, due to the uneven geographical distribution between nodes and the influence of the environment, it is easy to cause the communication between nodes to be blocked and the construction of random paths to fail. This paper proposes an efficient data collection algorithm for this problem. The algorithm is improved on the basis of the random node selection algorithm. This method can effectively avoid the failure of random path node selection and improve the node selection of random path in wireless sensor networks. Then, the sensor network in the dynamic environment is analyzed based on the static environment. An efficient data collection algorithm based on the position prediction of extreme learning machines is proposed. This method uses extreme learning machine methods to perform trajectory prediction for nodes in a dynamic environment.

1. Introduction

In recent years, with the development of wireless communication technology, sensor technology, and embedded technology, wireless sensor networks (WSNs) have been widely concerned [1]. Wireless sensor networks can rely on sensor nodes to sense a wide range of physical or environmental conditions, which are widely used in medical, military, scientific observation, emergency monitoring, and commercial applications. An important function of wireless sensor networks is to complete data collection. Sensor nodes collect sensor data and transmit it to Sink nodes through multihop wireless communication. Effective data collection is the key to various applications in wireless sensor networks, such as battlefield monitoring, habit monitoring, infrastructure monitoring, and environmental monitoring.

In the current research and development process, Shannon sampling theorem is generally used as the main data collection method for wireless sensor network signals. However, because it needs to collect a large amount of raw data and perform data transmission, it causes node energy consumption and data redundancy. As a new technology, compressed sensing uses low-speed sampling to compress and project the collected raw data and then uses nonlinear reconstruction algorithm to reconstruct the compressed signal. Compressed sensing avoids collecting a large amount of raw data, which has become the development direction of extensive research.

In wireless sensor network data collection, the traditional compressed sensing method uses the dense random matrix [2] as the observation matrix to compress and collect the data. However, the dense measurement matrix brings dense observation problems; that is, each projection value needs all nodes in the network to participate in the operation [3]. Therefore, the collection of the projection value requires great communication overhead, and the amount of calculation is huge when the signal is reconstructed. The proposed sparse random matrix collects and measures the data in the wireless sensor network [4] and collects the measurement data of some nodes in the network to construct the projection value, which reduces the communication cost between nodes and the amount of calculation when constructing the projection value. It plays a certain role in promoting the development of compressed sensing.

In the process of constructing random path based on sparse random matrix, the random nodes are initialized as the initial nodes of random paths, and then the random walk strategy is used to collect the node data. After each random path reaches the specified length, the projection of the collected data can be completed. Then, the collected data is projected to the Sink node, as shown in Figure 1. Finally, the project data is decompressed and reconstructed at the Sink node. At the same time, the observation matrix is generated when the random path is constructed; that is, the sparse random matrix is constructed. The observation matrix fills the parameter values in the matrix according to the different walk nodes of the random path, which are generally 0 or 1. The number of nonzero elements in each row of the observation matrix is consistent with the length of the corresponding random path. After reaching the specified length of random path, data reconstruction can be completed at Sink node. However, due to the uncertainty of the external environment, during the construction of the random path, the current nodes do not reach the neighbor nodes, resulting in the failure of random path construction and observation matrix construction. Therefore, it is particularly important to propose a more effective algorithm.

Based on the abovementioned problems, this paper proposes an efficient data collection method in the static and mobile environments of wireless sensor networks, which can effectively avoid the problems in the process of random path construction. The main contributions of this paper are as follows:

(1) We propose an efficient data collection algorithm in static environment. By statistical analysis of the current node's neighbor nodes, select the appropriate threshold to filter the neighbor nodes that meet the threshold, select the appropriate set of neighbor nodes, and select the next hop node and improve the traditional random selection algorithm in static environment.

(2) We propose an efficient data collection algorithm based on ELM position prediction. By using the method of limit learning machine to predict the geographical location of nodes at the next time, select the appropriate node as the next hop node at the current time according to the predicted location, and improve the construction length of the random path.

(3) The construction of random path in static network environment and mobile network environment is simulated by designing the prototype system of efficient data collection method in wireless sensor network. The comparison between the traditional data collection algorithm and the efficient data collection algorithm is achieved, and the correctness and effectiveness of the algorithm are verified under multiple tests, and the robustness and accuracy of the system are verified.

2. Related Work

This section introduces the existing research results from two aspects: wireless sensor networks and compressed sensing.

2.1. Wireless Sensor Network. Wireless sensor network is a new development direction in the field of communication, and it also converges the integration and development of traditional disciplines and emerging disciplines. Wireless sensor network originated in the United States in the 1990s. In the following years (more than 10 years), wireless sensor network technology has been widely concerned by industry, academia, and government. It has also become the focus of competing development in military, environmental, and many other fields. The US Business Week has selected ten emerging technologies that have a great impact on the future, of which wireless sensor network is the first. As soon as the wireless sensor network technology is released, it has attracted the extensive attention of military departments, academia, and industry in many countries in the world and has become a research hotspot. The American Computer Society (ACM) specializes in the creation of Transaction on Sensor Network, which is used to research the most advanced wireless sensor network results. The application demand of wireless sensor networks is to achieve the best monitoring of the target under the limited energy supply. Therefore, the wireless sensor network has undergone multiple designs and optimizations in this respect.

In recent years, wireless sensor network technology has made rapid development. Rebai et al. proposed a sensor deployment optimization method for wireless sensor network coverage and connectivity [5], Amine et al. proposed an efficient and secure weighted clustering algorithm for mobile wireless sensor network [6], and Diallo et al. proposed a distributed database management technology in wireless sensor network [7]. Bhuiyan et al. studied the mobile target tracking based on local area prediction in wireless sensor networks [8]. Iyer et al. proposed STCP sensor transmission control protocol [9]. In order to ensure the data security of wireless sensor networks, Ren et al. proposed an end-to-end security framework for position sensing, which ensures the security of sensor data [10]. Lin et al. proposed an adaptive transmission power control algorithm for wireless sensor networks [11]. All these scientific researches have promoted the development of wireless sensor networks.

In China, in view of the development of wireless sensor network in the world, China Computer Federation Technical Committee on Sensor Network was established in October 2006, which has contributed to the research and development of wireless sensor network in China. In recent years, domestic research institutions have made important contributions to the development of wireless sensor networks. In [12], an improved DV-Hop algorithm was proposed to realize the accurate localization of sensor nodes. Reference [13] summarized the privacy protection technology of wireless sensor networks. In [14], the OSNC method was

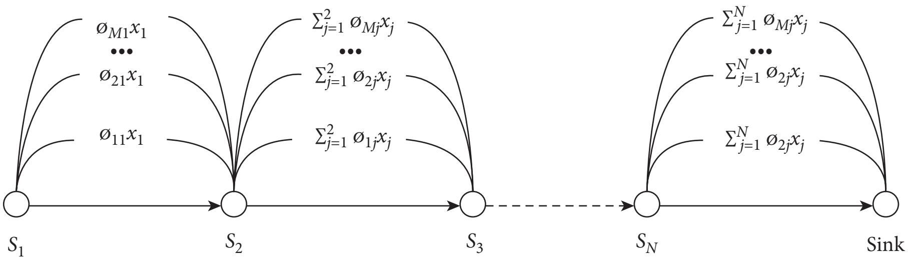

$\varnothing_{M1}x_1$
$\varnothing_{21}x_1$
$\varnothing_{11}x_1$
$\sum_{j=1}^{2} \varnothing_{Mj}x_j$
$\sum_{j=1}^{2} \varnothing_{2j}x_j$
$\sum_{j=1}^{2} \varnothing_{1j}x_j$
$\sum_{j=1}^{N} \varnothing_{Mj}x_j$
$\sum_{j=1}^{N} \varnothing_{2j}x_j$
$\sum_{j=1}^{N} \varnothing_{2j}x_j$
S_1
S_2
S_3
S_N
Sink

Figure 1: Data gathering process based on Compressed Sensing.

proposed to reduce the overload around the aggregation node. Now wireless sensor networks have been widely used in military, environment, agriculture, health care, intelligent transportation, and other fields. These will vigorously promote the good and rapid development of wireless network industry [15].

2.2. Compressed Sensing. Compressed sensing originated from Professor Mistretta [16] of the University of Wisconsin and others' questions: we can reconstruct the original image with a small amount of MRI sampling data. In order to solve this problem, Professor Mistretta used the classical image reconstruction theory to perform simulation experiments. The results show that the resolution is low and the edge is fuzzy. At the same time, Professor Candes's research team realized the accurate reconstruction of the original image from the limited adopted data by using the penalty function and proved that, from the Fourier transform coefficients of the image, random selection of no less than $2k$ coefficients can accurately and uniquely reconstruct the original signal, where k is the number of nonzero Fourier coefficients [17].

Donoho et al. [18–21] proposed a theoretical framework of compressed sensing based on the ideas of sparse decomposition and signal recovery. On this basis, Donoho formally proposed the concept of "compressed sensing" [18]. The core idea is to first acquire the nonadaptive linear projection of the signal and then reconstruct the original signal from the measured values according to the corresponding reconstruction algorithm. References [17, 18, 21, 22] have made in-depth research and analysis on accurate signal reconstruction, signal stability maintenance, and optimal sparse decomposition in incomplete measurement and have achieved good development. Compressed sensing has attracted the attention of many scholars and experts, such as Waheed and others, who have conducted extensive research on compressed sensing of network data. In 2008, Jarvis Haupt proposed a specific network compression method in network distribution, which is considered as the prototype of distributed compressed sensing. Scholar D. Baron et al. proposed a new theory of distributed compressed sensing, which has been widely used.

In recent years, a new compressed sensing theory has been born. For sparse signals or compressible signals, this method can achieve data compressed acquisition while acquiring signal, and its frequency is far lower than Nyquist sampling frequency [23]; its advantage is that it can reduce sampling data, save storage space, and contain enough signal information. In the process of data reconstruction, enough data points can be recovered according to the compressed data. Compressed sensing combines traditional data collection with data compression, but it does not need complex data encoding algorithm. Compressed sensing has become a new research direction in the field of signal processing [24].

At home and abroad, the theoretical and practical application of compressed sensing has become a hot research direction in recent years. In [25], an algorithm of MRI image reconstruction based on joint regularization and compressed sensing was proposed for traditional compressed sensing MRI image reconstruction. In [26], an adaptive compressed sensing algorithm based on energy balance was proposed to solve the problem of limited processing capacity and energy shortage of traditional wireless sensor networks. In [27], an energy-constrained Bayesian compressed sensing algorithm was proposed. Reference [28] analyzed the performance and delay analysis of wireless sensor networks. In [29], a random walk path algorithm based on nonuniform sampling was proposed to compress and collect data. In [30], a data model of random projection estimation algorithm was proposed to achieve less compression measurement. In [31], EDAL data collection protocol was proposed. In [32], a sequential sliding window processing framework was proposed. With the continuous deepening of the theory of compressed sensing, the research on compressed sensing has made great progress in various fields. In [33], a novel distributed compression estimation scheme for sparse signals and systems based on the theory of compressed sensing was proposed. In [34], a compressed sensing method based on Gaussian mixture model was proposed. In [35], a CSPR method for dynamic wireless sensor network path reconstruction based on compressed sensing was proposed. In [36], compressed sensing was applied to image restoration.

3. Efficient Data Collection Method in Static Sensor Network

3.1. Problem Description. This section first gives the definition of undirected graphs and the process of constructing random paths based on undirected graphs, then gives the observation matrix constructed from random paths, and finally gives the method of node selection in the process of constructing observation matrices. The symbols used in this chapter and subsequent chapters and their interpretations are shown in Table 1.

Table 1: Symbols and meanings.

Symbol	Interpretation
Φ	Random observation matrix
x	Raw data
y	Compressed data
Ψ	Sparse basis matrix
$\theta, \widehat{\theta}$	Sparse coefficient
k	Sparse value
A, Θ	Perception matrix
m	Number of paths
n	Number of nodes
μ	Correlation coefficient
t	Length of random path
W_u	Path W starting from u
$\deg(v)$	Degree of node v
$\mathrm{pre}(v)$	Previous node of node v
$\lvert V\rvert$	Vertex set
$\lvert E\rvert$	Edge set
$G(V, E)$	Undirected graph with V and E as parameters
R, r	Node communication radius

Definition 1 (undirected graph). A graph $G = (V, E)$ with no direction in an edge is called an undirected graph if and only if the following conditions are satisfied:

(1) Vertex set V is a nonempty set

(2) Edge set E is a set of unordered two-tuples composed of elements in V

Definition 2 (neighbor node). For node u, if node v exists, node u can communicate with node v within the communication radius, that is, satisfy $r(u, v) \leq R$ (R is the communication radius). Then, call node v a neighbor of node u.

Definition 3 (random path). In a wireless sensor network, if there is a path consisting of any consecutive neighbor nodes and the path length $t \geq o(n/k)$ is satisfied, the path is called a random path.

For the undirected graph $G = (V, E)$, let the vertex set V be the set of nodes in the wireless sensor network; that is, $V = n$; construct a Boolean observation matrix Φ by completing m random paths in the undirected graph $G = (V, E)$, where the wireless sensor network data collection is shown in Figure 2 below.

For example, during the data collection process in Figure 2, first, randomly select node v as the initial node of the path; second, node v selects neighbor node l as the next hop node of the random path within the communication range and then randomly selects neighbor nodes p and q to be added to the path; then, perceptual data is aggregated and projected on node q. Finally, the node q sends the projection data to the Sink node through the shortest routing method to analyze and recover the data.

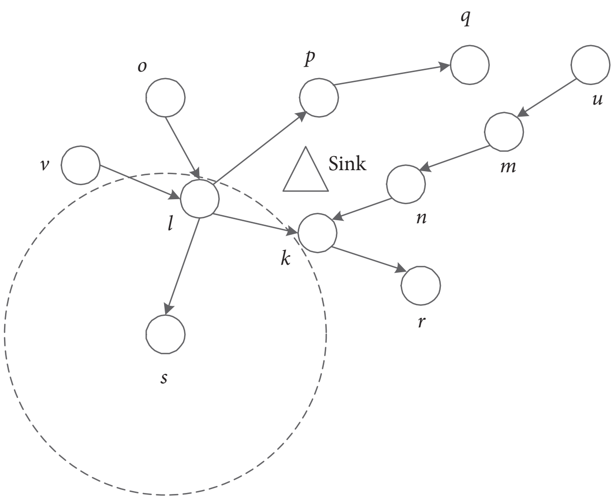


Figure 2: Process of data gathering with random walk.

In Figure 2, if the random path randomly selects the next hop node when it reaches the node, because the node's neighbor node set is only the node and has already arrived, the path does not satisfy the length requirements of the random path, so the observation matrix construction fails and the collection of perception data is incomplete. In order to avoid this situation, during the construction of each random path, the traditional node random selection algorithm is analyzed and an efficient data collection algorithm is proposed. During the construction of the random path, a neighbor node set is constructed for the current node, each node in the neighbor node set is counted separately for the number of neighbor nodes, and then the total number of neighbor nodes for all nodes in the neighbor node set is calculated; then set a reasonable threshold to filter the nodes in the neighbor node set, and, finally, select appropriate neighbor nodes for random selection.

3.2. Property Analysis. Because the selection of the current node in each random path may affect the selection of the next node, the nature of the wireless sensor network is analyzed first, and then the problems existing in the current network are analyzed according to the properties of the sensor network. The shortcomings of the method are improved, and new solutions are proposed.

Definition 4 (reachable nodes). In wireless sensor networks, if node u can reach node v through n ($n \geq 1$) consecutive neighbor nodes, then v is called reachable node of u.

Definition 5 (reachable path). In a wireless sensor network, if there is a path formed by n ($n \geq 2$) consecutive neighbor nodes starting with node u and ending with node v, then it is called a reachable path and is denoted as R (u, v).

Inference 1. The path formed between two reachable nodes is called the reachable path.

Definition 6 (reachable path length). If there is a reachable path R (u, v) starting with node u and ending with node v, then the number of connections between nodes u and v is called the reachable path length, which is denoted by $L<u, v>$.

In Figure 3, node v is a neighbor node of node u, node x is a neighbor node of node v, node u can reach node x

through node v, and node x is the reachable node of node u. The path $u \longrightarrow v \longrightarrow x$ is a reachable path $R(u, x)$, and the reachable path length is 2, which is $L<u, v> = 2$.

Theorem 1. *If node b is a neighbor of node a, then b must be a reachable node of a.*

Proof. Node b is a neighbor of node a. A reachable path $R(a, b)$ can be formed between node b and node a by Definition 5. That is, node a can reach neighbor node b, which meets Definition 4; node b is called reachable node of node a. From Figure 3 and Definition 4, we know that x is a reachable node of u, but x is not a neighbor node of u. □

Theorem 2. *If there is a random path, this path must be a reachable path.*

Proof. There is a random path. The random path length $t \geq o(n/k)$ is obtained by Definition 3. If k and n are known, then $n/k > 2$ exists. By Definition 5, the reachable path length requirement is met. Therefore, the random path length meets the reachable path length requirement; that is, the random path must be reachable path. □

Definition 7 (P function set). For node a's neighbor node set $_aT$, node *0*'s neighbor node set is T_b and satisfies node $b \in T_a$; then the function $p(a, b) = T_a U\ Tb - \{a\}$ is called the P function set of node a to neighbor node b.

Definition 7 clearly indicates the set of nodes within the communication range of the current node and any of its neighboring nodes and provides the necessary set of nodes for the next efficient data collection algorithm. As shown in Figure 4, assuming that the initial node of the path in this network is b, and within its communication range, its neighbor node set is, $T_b = \{a\ c\ y\}$, and then each node in the T_b set is calculated separately and their P function sets select the construction selection domain for the next hop node. After obtaining the P function set of each node, a reasonable threshold can be set to filter the appropriate neighbor nodes for the next hop node selection.

For a, $T_a = \{b\}$, $P(b, a) = T_b$ U $T_a - \{b\} = \{a, c, y\}$;

For c, $T_c = \{b, p, y, s, d\}$, $P(b, c) = T_b$ U $T_c - \{b\} = \{c, p, y, s, d, a\}$;

For y, $T_y = \{b, c, d, s, t, x\}$, $P(b, y) = T_b$ U $T_y - \{b\} = \{a, c, d, y, s, t, x\}$.

After obtaining the function set of each node, a reasonable threshold can be set to screen suitable neighbor nodes for the next hop node selection.

3.3. Algorithm Description. As can be seen in Figure 2, for the current node on any reachable path, if the next node is randomly selected, it may be happen that the neighbor node set of the next node may be empty or the number of neighbor nodes is few, resulting in the failure of random path construction and incomplete data collection. According to Definition 7, on the premise of the current node, a reasonable threshold is set by calculating the number of P function sets of the current node and each neighbor node, and, on this basis, a reasonable selection of the next node is carried out, so that the random path reaches a reasonable length, and the projection of perceived data is completed.

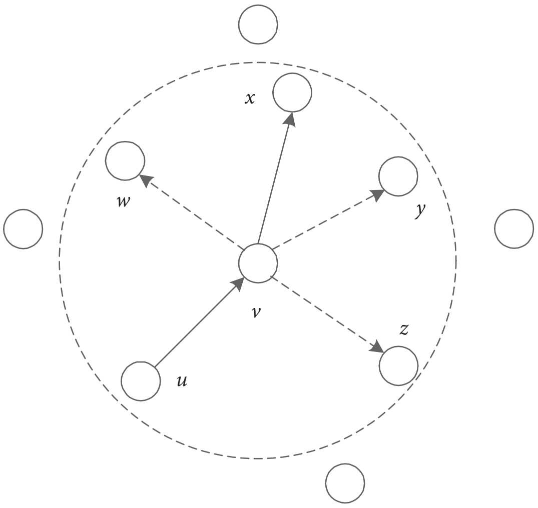


FIGURE 3: Select of next node.

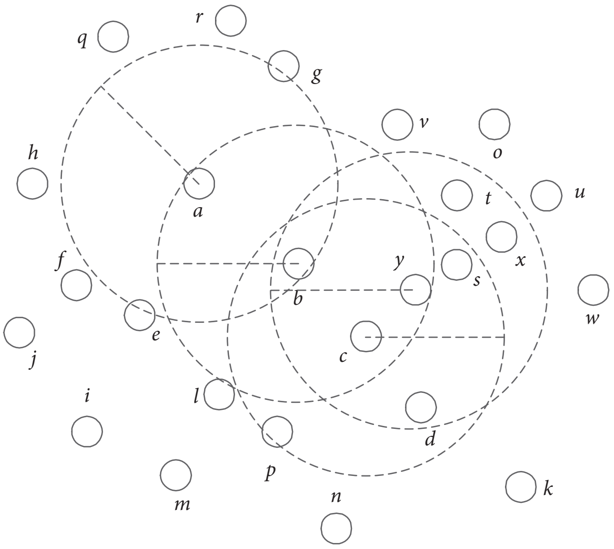


FIGURE 4: P function set.

Based on the current nodes, this section compares the traditional Nodes Random Selection Algorithm in Static WSNs (NRAS) and the proposed Efficient Data Collection Algorithm in Static WSNs (EDCAS) and makes reasonable descriptions of the two algorithms, respectively.

In data collection of wireless sensor networks, firstly, m nodes are randomly selected as initial nodes of m random paths. In the process of data collection of each random path, appropriate nodes are selected through different node selection methods to acquire data. Finally, the acquired data are gathered at the end of the path to form projection values of perceptual data, which are sent to Sink nodes for data reconstruction.

Definition 8 (Neighbor node dataset to be selected). To connect any node of the reachable path in the wireless sensor network, the neighbor node set within the letter range is called the neighbor node dataset to be selected and is expressed as SNS. Firstly, select the initial nodes. Sink node selects M initial nodes from N nodes through its own calculation starting node, each starting node serves as the starting point of each random path, and the specific process is shown in Algorithm 1 below.

As shown in Algorithm 1, firstly, Sink nodes in the wireless sensor network are randomly generated n element vectors by rand$(1 * n)$ function, and the sequence of nodes is 1, 2, ... (step 1); secondly, $1 * n$ elements are sorted by sort(n) function (step 2); then, a node k is randomly selected from the nodes through the randSelect(p, n) function (step 4). If the currently selected node already exists in the index set S, the initial node is recalculated until there are no duplicate nodes (step 5 to 7); finally, it is added to the initial node index set S (step 8) and returned to the set (step 10).

In the node selection algorithm of wireless sensor networks, first, we analyze the traditional random node selection algorithm, and, on this basis, an efficient data collection algorithm is proposed.

As shown in Algorithm 2, the current node is v. Firstly, the neighbor node set *SNS* (step 1) of the v node to be selected is obtained through the function getSNS(v); secondly, the neighbor node u (step 2) is randomly selected from the SNS through the function randSelect(SNS). If the node has already existed on the path, it means that the node has already been visited. Then the next hop node (steps 3–5) is randomly selected from SNS again, and then the data collected by the random path in the previous step and the data collected by the node u are superimposed at the node u (step 6), the current random path length is added by one (step 7), and the current node is marked as have been accessed (step 8); finally, return to the next hop node u (step 9).

Algorithm 2 takes Figure 5 as an example. Assuming that the current initial node is c and the collected data is $x\ c\ (t)$, the neighbor nodes in the communication range of c have five nodes as b, p, y, s, and d; then the neighbor node dataset SNS $= \{b, p, y, s, d\}$; then node d is selected randomly from SNS; the next node of c is d; that is to say, the data collected by node d is $x_d(t+1) = x_c(t) + x_d(t)$. The reachable nodes in the communication range of node d are c, y, and k, namely, the neighbor node dataset SNS $= \{c, y, k\}$. If node d randomly selects the next hop node as k, then the data collected by node k is $x_k(t+1) = x_d(t) + x_k(t)$ (t is the length of the current random path construction step).

In Figure 5, due to the distribution edge of node k, the *SNS* set of node k is only d, and node d has already been accessed. If the current path length does not meet $t \geq o(n/k)$, then random path data collection fails and projection of perceived data cannot be completed.

Based on the traditional random node selection algorithm, this section proposes an efficient data collection algorithm in a static environment, with specific implementation as Algorithm 3. Firstly, the neighbor node dataset SNS (step 1) is selected for the current node, the number of unlabeled nodes in the current SNS set is recorded, and the total number of nodes in the P function set (steps 2-3) is initialized; then, the P function set of nodes is calculated for each unlabeled node in the *SNS*, respectively, and the node data of the P function set is added to the total number of nodes totle (steps 4–9). Finally, calculate the average threshold and use in the next hop node selection (step 10). For each unlabeled node in SNS, the length of the P function set is calculated separately. If the length is longer than or equal to the average threshold, the current node is added to the cache neighbor node set *tempSNS* (steps 11–15). Then, the function *randSelect*(*tempSNS*) is used to randomly select node u from the *tempSNS* set as the next hop node (step 16). Finally, add the data collected by the current node to the path, add one to the current path length (steps 17-18), mark the current node (step 19), and return to the next hop node u (step 20).

Take Figure 6 as an example to describe the efficient data collection algorithm in detail. Assuming that the current initial node is b, firstly obtain the neighbor node dataset SNS $= \{a, c, y\}$ of node b to be selected, the number of unlabeled nodes in the dataset is length $= 3$, and totle is initialized to 0: then the p function set is calculated by *getPSet*(p) function for each value in the *SNS* set, respectively.

For node a, $p\ (b, a) = \{c, y, a\}$, $p\ (b, a)$ length $= 3$, totle $= 3$;

For node c, $p\ (b, c) = \{p, y, s, d, a, c\}$, $p\ (b, c)$ length $= 6$, totle $= 3 + 6 = 9$;

For node y, $p\ (b, y) = (a, c, d, s, t, x, y)$, $p\ (b, y)$ length $= 7$, totle $= 9 + 7 = 16$.

Then, calculate the average threshold value average $=$ totle/length $= 16/3$, and then each unlabeled node in *SNS* is traversed again. For each node, the P function set is calculated separately. If the length of the P function set is longer than or equal to the average threshold, the node is added to the cache neighbor node set *tempSNS*.

For node a, $P\ (b, a)$ length $= 3 <$ average, do not add to *tempSNS*;

For node c, $P\ (b, c)$ length $= 6 >$ average, add to *tempSNS*;

For node y, $P\ (b, y)$ length $= 7 >$ average, add to *tempSNS*.

In the set *tempSNS* $= \{c, y\}$, node c is randomly selected from *tempSNS* as the next hop node, and the node data is superimposed; then the current random path length is increased by 1, and node c is marked as accessed at the same time.

Efficient data collection algorithm can avoid the failure of node selection in the process of data collection for random paths to some extent. As shown in Figure 6, Algorithm 3 filters the neighbor node a in the set *SNS*. From the figure, it can be observed that if node a is selected as the next hop node of the current node, then since the dataset of the neighbor nodes to be selected of node a is SNS $= \{b\}$, and node b has already accessed the tag, it may cause node a to fail to continue selecting nodes for

```
Function: Select the initial nodes of $m$ random paths
Input: Amount of node $n$
  Amount of random path $m$
Output: Index set $S$
Algorithm Description:
rand($1 * n$);    //Randomly generate $1 * n$ element vectors
sort($n$);    //Sort the $1 * n$ elements
foreach $p \le m$
    $k = randSelect(p, n)$;    //Randomly choose node $k$ form $n$ nodes
    while $k$ exists in $S$ do
        $k = randSelect(p, n)$;
    end while
    $S$.add($k$);    //Add node $k$ to the initial index set $S$
end foreach
return $S$;    //Return to set $S$
```

Algorithm 1: The initial node selection in wireless sensor networks.

```
Function: Randomly select the next node in the random path
Input: Current node v
Output: Next node u
Algorithm Description:
SNS = getSNS($v$);    //Obtain the neighbor node set SNS
$u = randSelect$(SNS);    //Randomly select node $u$ from SNS
while $u$ exists in $Route$ do
$u = randSelect$(SNS);
end while
$x_u(t) = x_v(t-1) + x_u(t-1)$;    //Superimposing node's data
$Route.length = Route.length + 1$;    //Increase 1 to the length of random path
$u.flag = 1$;    //Mark $u$ as accessed
return $u$;    //Return to u
```

Algorithm 2: Random select node.

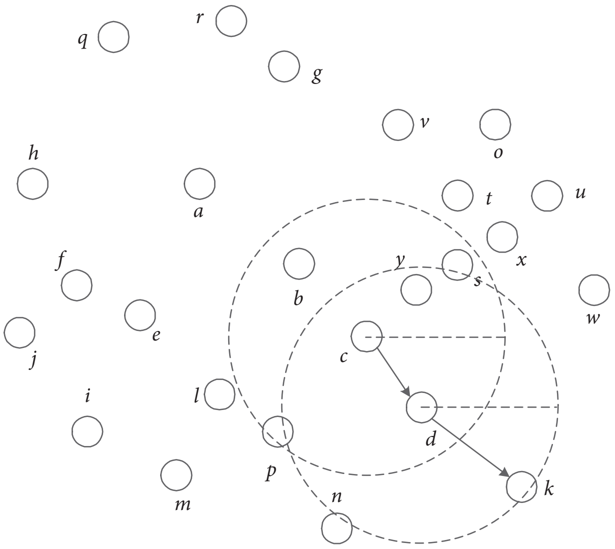


Figure 5: Example of random select node.

this path. If the random path length does not reach the required length at node a, the sensor sensing data collection will fail.

4. Efficient Data Collection Method in Mobile Sensor Network

4.1. Problem Description. This section will introduce the change of wireless sensor node position in the mobile network environment and the method of node selection in random path data collection. In a static environment, a node can communicate with nodes within a communication radius. However, in a mobile network environment, the location of the node may change due to changes in the external environment. The nodes that can communicate can lose contact due to changes in distance, and then the construction of random paths has an impact.

As shown in Figure 7, it is assumed that three random paths are required, with each including five sensor nodes. Figure 7(a) shows the node position at time $t = t_1$. First, three initial nodes a, f, and m are selected randomly. Then, within the communication radius of each node, the node randomly selects the neighbor nodes to form $a \longrightarrow b \longrightarrow c \longrightarrow d \longrightarrow e$, $f \longrightarrow g \longrightarrow i \longrightarrow j \longrightarrow k$, $m \longrightarrow n \longrightarrow j \longrightarrow o \longrightarrow p$, three routes for data collection. Among them, the solid line indicates the route selected by the nodes of each random path, and the dashed line indicates the position change of the

```
Function: Randomly select the next node in the random path
Input: Current node v
Output: Next node u
Algorithm Description:
SNS = getSNS(v);        //Obtain the neighbor node set SNS
length = SNS.length;        //Obtain the length of unmarked node in SNS
totle = 0;
foreach node p in SNS
    if P not exists in Route then
        P(v, p) = getPSet(p);
        totle = totle + P(v, p).length;
    end if
end foreach
average = totle/length;
foreach node k in SNS
    if k not exists in Route and getPSet(k).length ≥ average then
        tempSNS.add(k);
    end if
end foreach
u = randSelect(tempSNS);        //Randomly select node u from tempSNS
$x_u(t) = x_v(t-1) + x_u(t-1)$;        //Superimposing node's data
Route.length = Route.length + 1;        //Increase 1 to the length of random path
u.flag = 1;        //Mark u as accessed
return u;        //Return to u
```

Algorithm 3: Efficient data gathering algorithm.

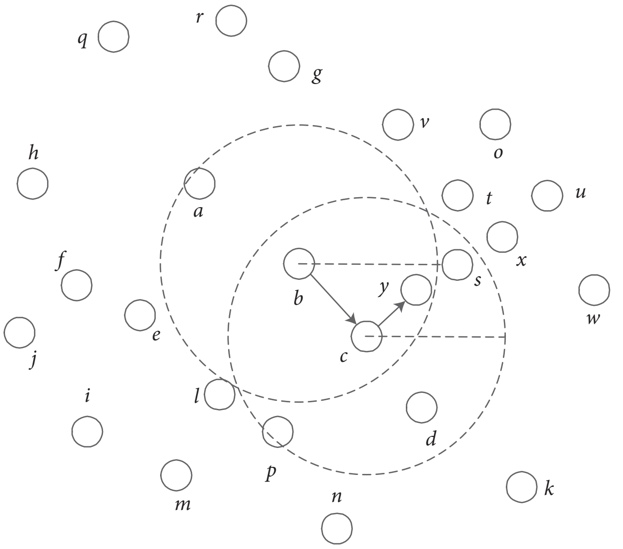
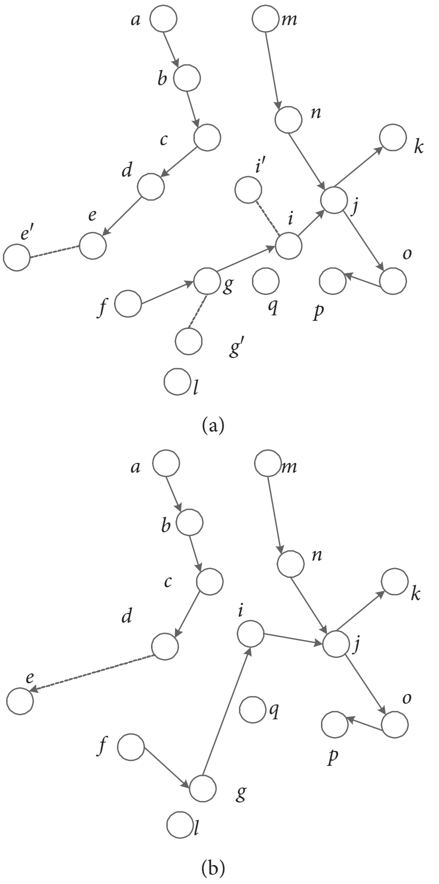


Figure 6: Example of random select node.

sensor node at the next moment; for example, i represents the position of node i at $t = t_2 (t_2 = t_1 + 1)$. The node position at time $t = t_2$ is shown in Figure 7(b), where the solid line indicates the random path selection after the node position changes, and the dashed line indicates that the connection of the random path is disconnected due to the node position change. The distance of d exceeds the communication radius, and communication cannot be performed, so the path $a \longrightarrow b \longrightarrow c \longrightarrow d \longrightarrow e$ becomes $a \longrightarrow b \longrightarrow c \longrightarrow d$: the positions of nodes g, i, and l have changed, but nodes f and g



(a)

(b)

Figure 7: Sensor node dynamic network. (a) $t = t_1$. (b) $t = t_2$.

are still within the communication radius, and communication can still be performed. Node g and node l can also communicate. Node g and node i have a larger communication distance and cannot communicate. The neighbor node dataset of node g is $\{f, l, q\}$. The neighbor node dataset of node l is $\{g\}$, so the path $f \longrightarrow g \longrightarrow i \longrightarrow j \longrightarrow k$ cannot communicate with node i when it reaches node g. If node g randomly selects node l as the next node on the path, similarly, after reaching l, because the neighbor dataset is only g cannot select the next hop node, the path becomes $f \longrightarrow g \longrightarrow l$. Because the path length does not meet the requirements, the random path data collection fails.

In the abovementioned example, if the path $f \longrightarrow g \longrightarrow i \longrightarrow j \longrightarrow k$ can predict the position of node g when it reaches node f and predict the next node based on the size of the neighbor node dataset node of g, it will work well. The selection of node paths avoids the failure of the construction of random paths.

Therefore, it is an important part to be able to perform position prediction on nodes on a random path. This chapter uses extreme learning machine methods to predict node locations. Extreme learning machine is a simple and effective single hidden layer feedforward neural network learning algorithm. The extreme learning machine only needs to set the number of nodes in the hidden layer of the network, and the algorithm does not need to adjust the input weight of the network and the offset of the hidden elements to generate the optimal solution. The extreme learning machine has the advantages of less manual intervention, fast learning speed, etc., and it is not easy to generate a local optimal solution during the learning process, so it is widely used in text classification, image recognition, biomedicine, and other fields.

4.2. Property Analysis.

In the mobile network environment, the construction of random paths may fail due to changes in node locations. Therefore, this section first analyzes the properties based on this and then proposes new solutions.

Definition 10 (prenode and postnode). In a wireless sensor network, if there is a path that connects node a directly to node b, then node a is called the prenode of node b and node b is called the postnode of node a.

Theorem 3. *If node a is a predecessor (or postnode) of node b, then node a must be a neighbor of node b.*

Proof. If node a is the predecessor (or postnode) of node b, there must be $r\ (a, b) \leq R$ (R is the communication radius); then the definition can be found from the neighbor nodes. □

Definition 11 (historical location). In the mobile network environment, the existence node ν is recorded as L_t^v at time t; then if there are times t_1 and t_2 $(t_1 \leq t_2)$, then L_{t1}^v is called the historical position of L_{t2}^v.

Definition 12 (historical trajectory and future trajectory). In a mobile network environment, if there is a set of positions of node u in continuous unit time as $N = \{L_{t1}^u, \cdots, L_i^u, \ldots, L_{tn}^u\}$ (i$\in[t1, t_n]$), then it is called a set $N_1 = \{L_{t1}^u, \cdots, L_i^u\}$ which is set $N_2 = \{L_i^u, \ldots, L_{tn}^u\}$'s historical trajectory; N_2 is N_1's future trajectory.

Definition 13 (historical path and future path). In a mobile network environment, if there is a reachable path $R(u, v)$ and a node $k (k \in [u, v])$ exists, the reachable path is divided into two parts $R(u, k)$ and $R(k, v)$; then the path $R(u, k)$ is called the historical path of $R(k, v)$, and the path $R(k, v)$ is the future path of $R(u, k)$.

As shown in Figure 7(a), for the random path $a \longrightarrow b \longrightarrow c \longrightarrow d \longrightarrow e$, node c is called the predecessor of d, and d is the postnode of c. Node c and node d are neighbor nodes to each other. Nodes d and h are neighbors to each other in the communication range, but because there is no path connectivity between the two nodes, they cannot be called prenodes or postnodes: in the random path $f \longrightarrow g \longrightarrow i \longrightarrow j \longrightarrow k$, node g is called the historical position of node g', reachable path $f \longrightarrow g \longrightarrow i$ is called the historical path of reachable path $i \longrightarrow j \longrightarrow k$, and reachable path $i \longrightarrow j \longrightarrow k$ is called reachable future path of $f \longrightarrow g \longrightarrow i$ future path.

4.3. Algorithm Description.

Based on the analysis of Nodes Random Selection Algorithm in Mobile WSNs (NRSAM) in traditional mobile wireless sensor networks, this section proposes a university data collection algorithm (ELM) based on extreme learning machine position prediction in a mobile wireless sensor network-based Efficient Data Collection Algorithm in Mobile WSNs (EEDCM) and dynamically selects the next hop node, so that the most avoidable path can reach the corresponding length, thereby achieving the purpose of data collection.

As shown in Figure 8, first, the Sink node randomly selects m different nodes from the n nodes of the sensor network as the initial nodes (shaded nodes in the figure) through its own operation and then sends them to the selected m nodes by broadcasting and starts the construction of the random path. The process is shown in Algorithm 4.

This section first analyzes the node random selection algorithm in the traditional mobile network environment. The node random algorithm in the mobile environment is shown in Algorithm 5. The process of the node random selection algorithm in the mobile network is illustrated in Figure 9, where the black arrow indicates the next hop data collection node of the node. As shown in Figure (a), the initial distribution of network nodes at time $t = t_1$, node a is the initial node, and its neighbor node set is $\{b, d, g, l\}$, and then node b is randomly selected as the path. For one-hop node, at time $t = t_2$, the node position changes, and the set of neighbor nodes of node b is $\{a, d, p, l, o\}$, and then node p is randomly selected as the next hop node. At time $t = t_3$, the positions of nodes g and q and other nodes move, the set of neighbor nodes of node p is $\{b\}$, and node b has been visited

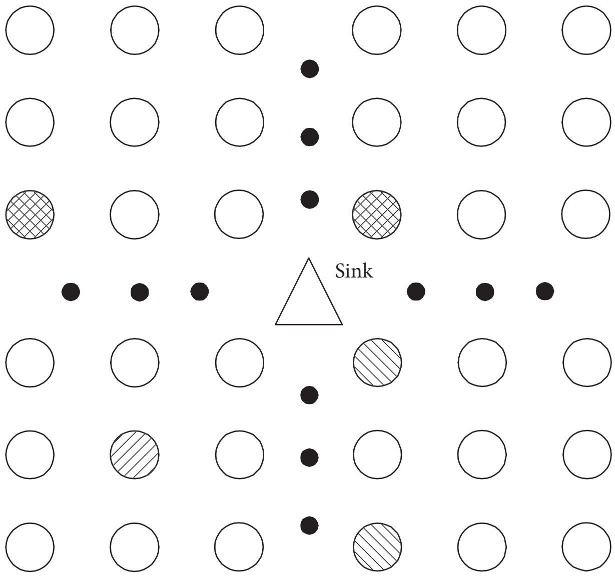


Figure 8: The initial node selection in wireless sensor networks.

```
**Function**: According to learning machine training parameters
**Input**: The number of hidden nodes $L$, the node's historical trajectory vector $F$, and the node's future trajectory vector $T$.
**Output**: ELM three parameters $w$, $b$, $\beta$.
**Algorithm Description**:
for $i = 1$ to $L$ do
Randomly generate hidden layer nodes $w_i$, $b_i$, $i = 1, \ldots, L$
Compute hidden layer node output matrix $H$ according to $F$;
Calculate $\beta = H'T$;
Return <$w$, $b$, $\beta$>;
```

Algorithm 4: Parameter training.

```
**Function**: Prediction of node position in sensor network.
**Input**: Number of sensor nodes $N$, number of hidden layer nodes $L$, node historical trajectory vector $F$, node parameters $w$, $b$, $\beta$
**Output**: Test result $S$
**Algorithm Description**:
Calculate the hidden layer output matrix $H$ by $F$, $w$, $b$
Obtaining test results $S = H * \beta$
Return $S$;
```

Algorithm 5: Detection function.

and cannot be repeatedly selected. Therefore, node P cannot select the next hop node for random selection, resulting in random travel. The path cannot meet the minimum length requirement and the construction fails. At time $t = t_4$, the node position changes, and node P selects neighbor node g as the next hop node.

In the traditional random path data collection process, the node's location changes due to changes in the mobile network environment and the random path construction fails. Therefore, this section proposes an efficient data collection algorithm based on ELM location prediction in mobile networks. In the case of the current node, the network node is predicted and the appropriate value is selected to select the next hop node to ensure the construction length of the random path.

During the node selection process of the random path, ELM is used to predict the position of each node in the sensor network. First, the historical trajectory of each node

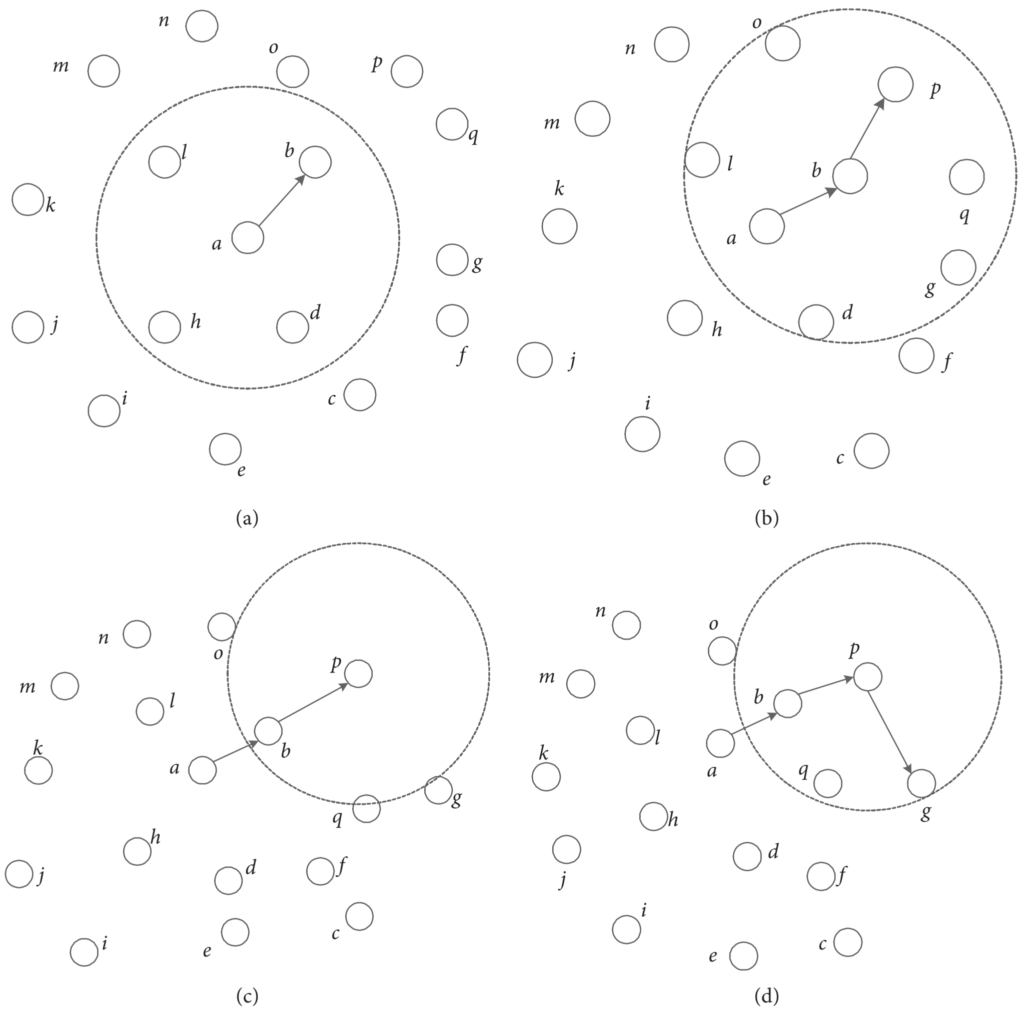


FIGURE 9: Example for nodes random selection algorithm in mobile networks. (a) $t = t_1$, (b) $t = t_2$, (c) $t = t_3$, and (d) $t = t_4$.

is sent to the Sink node, and the node is parameterly trained according to the trajectory of the node. After the parameters, they are sent to the nodes in the network through Sink broadcast, and then, according to the training parameters, the nodes in the network can obtain the predicted trajectory.

The algorithm for ELM training parameters is shown in Algorithm 4. First, the historical trajectory vector F and the future trajectory vector T of the nodes and the number of hidden layer nodes L are obtained in a dynamic environment. Then the hidden layer node parameters are w_i and b_i (lines 1-2). Then calculate the hidden layer node output matrix H according to F (line 3). Then calculate the parameter β (line 4) according to the formula $\beta = H'T$. Finally, return the parameters.

After training on parameters of the extreme learning machine, it can predict the future position of each node in the wireless sensor network, as shown in Algorithm 5. First, use the node history trajectory F and the training node parameters to calculate the hidden layer output matrix H (line 1) by (1); then use $S = H * \beta$ to obtain the detection result (line 2) as shown in Equation (2), and return the prediction result (line 3):

$$H\left(W_1, \ldots, W_L, b_1, \ldots, b_L, \ldots, X_1, \ldots, X_L\right) = \begin{bmatrix} g\left(W_1 \cdot X_1 + b_1\right) & \ldots & g\left(W_L \cdot X_1 + b_L\right) \\ M & \ldots & M \\ g\left(W_1 \cdot X_N + b_1\right) & \ldots & g\left(W_L \cdot X_N + b_L\right) \end{bmatrix}, \tag{1}$$

$$\widehat{\beta} = H'T. \tag{2}$$

Next, analyze the efficient data collection algorithm in the mobile sensor network environment, as shown in Algorithm 6. First, for the current node v, the prediction position of the current node (line 1) is obtained by the extreme learning machine prediction function *getELMLocation*(), and the prediction process is shown in Algorithm 5. Then perform ELM node position prediction for all other nodes in the wireless sensor network, add new predicted nodes to the node position set *tempN*(line 34), and obtain node v to be selected from the node predicted position set *tempN*.

Node set *SNS*(line 6) and parameter information are initialized (lines 7-8), and then a P function set is calculated for each unlabeled node in the *SNS* set. The

```
Function: Select the next hop node of the random path in the mobile network environment
Input: Current node $v$ on random path, node set $N$
Output: Next hop $u$ of random path
Algorithm Description:
(1) $v' = v.getELMLocation()$;    //Get the current node predicted position $v'$ by ELM
(2) foreach node $n$ in N
(3)     $n' = n.getELMLocation()$;    //Get the predicted position of the nodes in the sensor network through ELM
(4)     $tempN.add(n')$;    //Add new node to new node position combined with $tempN$
(5) end foreach
(6) $SNS = getSNS(v', tempN)$;
(7) $length = SNS.length$;    //Get the length of unlabeled nodes in the $SNS$
(8) $totle = 0$;
(9) foreach node $p$ in $SNS$
(10)     if $p$ not exits Route then
(11)         $P(v', p) = getPSet(p, tempN)$;
(12)         $totle = totle + P(v', p).length$;
(13)     end if
(14) end foreach
(15) $average = totle/length$;
(16) foreach node $k$ in $SNS$
(17)     if $getPSet(k, tempN).length \geq average$ then
(18)         $tempSNS.add(k)$;
(19)     end if
(20) end foreach
(21) $u = randS\,elect(tempSNS)$;    //Randomly select node $u$ from $tempSNS$
(22) while $u$ exists in Route or $R(v, u) > r$ do
(23)     $u = randSelect(tempSNS)$;
(24) end while
(25) $x_u(t) = x_v(t-1) + x_u(t-1)$;    //Node data overlay
(26) $Route.length = Route.length + 1$;    //Current path length plus 1
(27) $u.flag = 1$;    //Current node $u$ is marked as passed
(28) return $u$;    //Returns the next hop node $u$
```

Algorithm 6: ELM-based efficient data gathering algorithm.

reference nodes in the calculation are nodes in the node prediction position set *tempN*, and the length of the P function set of the unlabeled nodes is accumulated (lines 9–14), and the average length is obtained as the selection threshold (line 15). Then, for each unlabeled node in the set *SNS*, determine whether the length of the node P function set is greater than or equal to the average threshold, and if it is greater, add it to the *tempSNS* set as the node to be selected (lines 16–20). Next, randomly select nodes from *tempSNS*. If the selected node μ and the current node v are not in the communication range or the selected node μ has collected data in the current path, then the nodes in the *tempSNS* set are randomly selected again until the appropriate node (lines 22–24); finally superimpose the node data (line 25), add one to the current path length (line 26), and mark the selected node to return its node information (lines 27-28).

In terms of analysis of the time complexity of all the algorithms, the number of executions of basic operations in the algorithm is a measure of the time complexity of the algorithm. For Algorithm 6, the size of it is N; then the number of loops is determined. The time complexity $T(n) = O$ ($f(n)$, fastest-growing term/term factor). The time complexity of the algorithm can be determined as $O(N)$.

5. Experiments

In this section, the simulation experiments are performed under the variable parameter variables such as signal sparsity and communication radius. The random selection method of nodes in static environment and the efficient data collection algorithm are implemented. Finally, the performance of the two algorithms is compared and analyzed. During the experiments, random paths are selected by using the synthetic data and the sensor node data from IntelLab laboratory.

5.1. Experimental Environment

5.1. Experimental Environment. The experimental environment in this chapter is as follows.

(1) Operating system: win7 64 bit operating system

(2) Memory: 8 GB RAM

(3) Processor: Intel (R) Core (TM) i5

(4) Algorithm language: C++

The data used in this simulation experiment are artificial data and IntelLab laboratory data:

(1) In order to verify the effectiveness of the efficient data collection algorithm, 256 nodes' location data information is synthesized and a 100 × 100 sensor

network environment is constructed in this experiment. Because of the randomness of sensor network layout in the actual wireless sensor network environment, 256 nodes' location data are randomly generated, and then the program removes the duplicate nodes and generates new location nodes.

(2) The data used in the simulation experiment is the node location data of IntelLab laboratory from February to April 2004. There are 54 sensor nodes, which are distributed in 35×45 network environment to verify the effectiveness of the efficient data collection algorithm.

Table 2 lists the parameter value change range and default value in the experiment. In the simulation experiments, the main parameters are signal sparsity k, the number of sensor nodes n, and the node communication radius r. The main parameters are the length changes that can be reached by m random paths in the range of multiple parameters. During the experiment, when one of the parameter variables changes, the other two reference variables are set as default values.

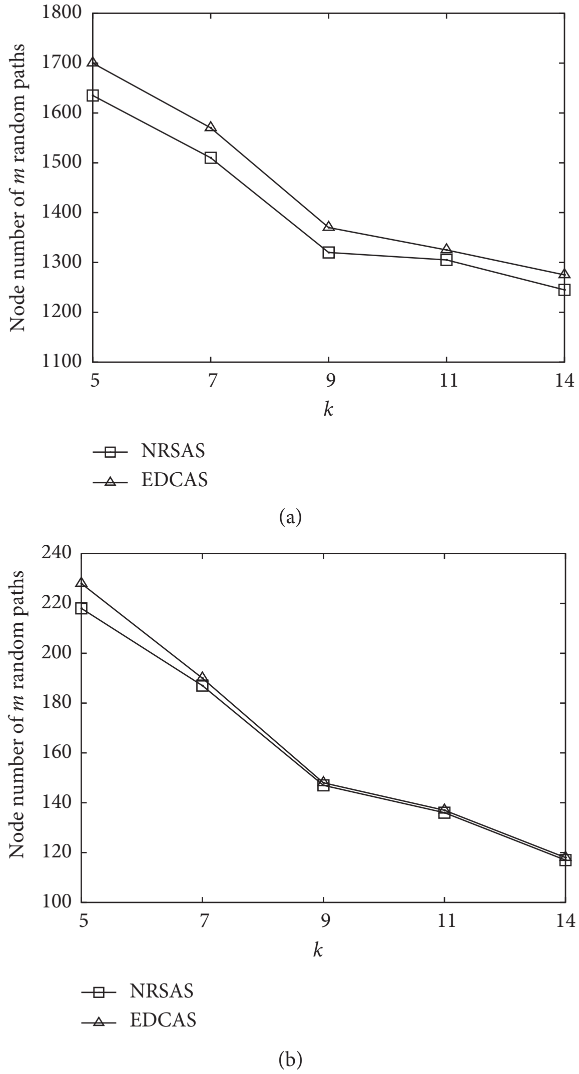
When considering the effect of the number of nodes n on the performance of the two algorithms, this section only compares the two algorithms by using synthetic data.

5.2. Performance of Efficient Data Collection Methods in Static Sensor Networks. Figure 10 shows the effect of different signal sparsity on the total number of random walk path nodes. As the signal sparseness changes, both the random path length t and the number of random paths m change. Under different signal sparsity, each test case is statistically summed 15 times and the average node length is calculated. It can be observed from the figure that, as the signal sparsity increases, the total length of actual nodes of m random paths decreases. Then it is observed that, under the same signal sparsity, the efficient data collection algorithm has better performance than the node random selection algorithm, and its performance advantage is more obvious when the signal sparsity is smaller.

Figure 11 shows the performance comparison between the two algorithms in the case of different nodes in the network. It can be observed from the figure that, with the increase of the total number of nodes in the network, the total number of actual random path nodes obtained by random selection algorithm and efficient data collection algorithm shows an increasing trend, because, with the increase of the value, the number of random walk paths and the length of random walk paths are increasing. Compared with the same node in the network, the high-efficiency data collection algorithm can get more random path nodes than the random selection algorithm, and, with the increase of the total number of nodes, the performance of the high-efficiency data collection algorithm is more obvious.

Table 2: Range and default values of parameters.

Parameter	Default	Optional parameter value
K	7	5, 9, 11, 14
Number of nodes n	256	169, 400
Communication radius r	8	4, 6, 10, 12



(a)

(b)

Figure 10: The effect of the sparsity on the random path. (a) Synthetic dataset. (b) IntelLab dataset.

Figure 12 shows a comparison of the performance of the two algorithms under different communication radius, with other parameters remaining at their default values. Since the total number of nodes and the sparsity of signal remain unchanged, the length of random walk path remains unchanged in theory. It can be seen from the figure that, with the increase of the communication radius of the sensor nodes, the total number of nodes on the random path shows an increasing trend. When the communication radius reaches a certain value, the total number of nodes basically tends to be the same. The reason is that, as the communication range of the nodes becomes larger, the dataset of the neighbor nodes becomes larger, and the number of the next hop nodes that can be selected increases, the construction of the random walk

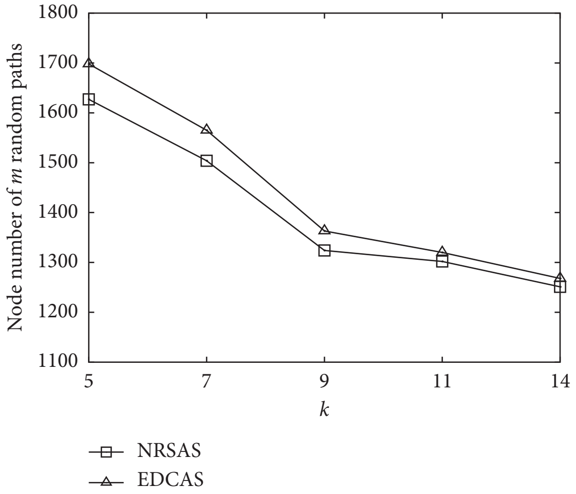
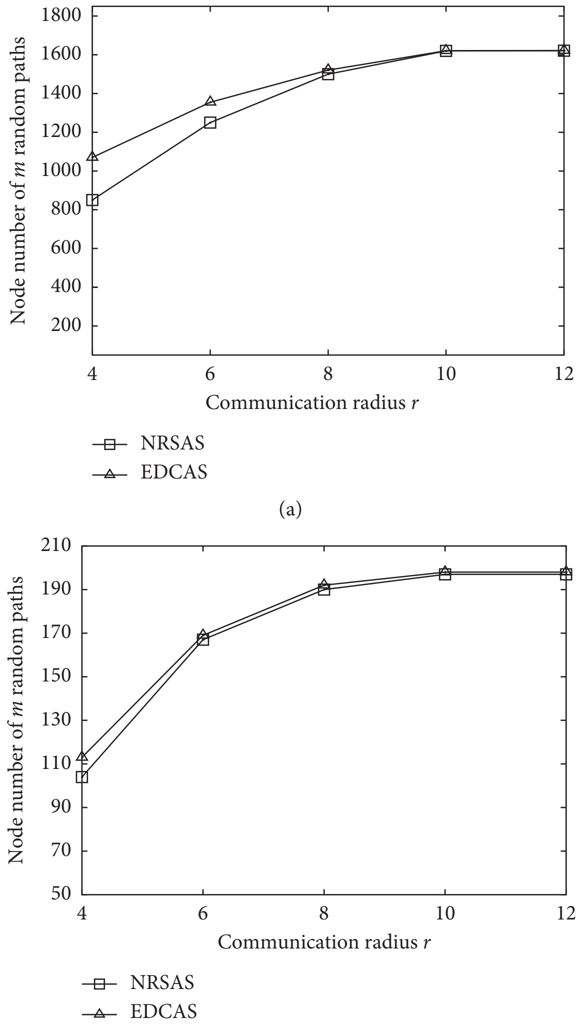


Figure 11: The effect of the node number on the random path.

path can be completed. Then it is observed that, under the same communication radius, the efficient data collection algorithm can show better performance than the node random selection algorithm.

5.3. Performance of Efficient Data Collection Methods in Mobile Sensor Networks. As shown in Figure 13, the effect of signal sparsity on the total number of random path nodes in the mobile network environment is demonstrated, in which the parameter variables such as communication radius and total number of nodes keep the default values. It can be seen from the figure that, with the increase of signal sparsity k degree, the total number of nodes in random selection algorithm and efficient data collection algorithm based on ELM position prediction is decreasing. Then compare the performance of the two algorithms under the same signal sparsity. Through analysis, we can see that, under the same signal sparsity, the efficient data collection algorithm based on ELM position prediction can get more nodes than the random path collection algorithm. More data is collected, so the former has better performance advantages. With the increase of signal sparsity, the gap between the two algorithms gradually narrows, because, with the increase of signal sparsity, the required length of random path gradually reduces. When the value is larger, the performance advantage of efficient data collection algorithm is more obvious.

As shown in Figure 14, the effect of the change of the total number of different nodes n in the network on the total number of m random path nodes is demonstrated, and other parameters keep the default values. It can be seen from the figure that, with the change of the total number of nodes n, the total number of m random path nodes in both algorithms shows an upward trend, because, with the increase of the total number of nodes, the required length t and the number of random paths m increase. Then it is observed that, under the same node number n, the efficient data collection algorithm based on ELM position prediction has better performance than the node random path collection algorithm. Also, as n increases, this performance advantage becomes more obvious.

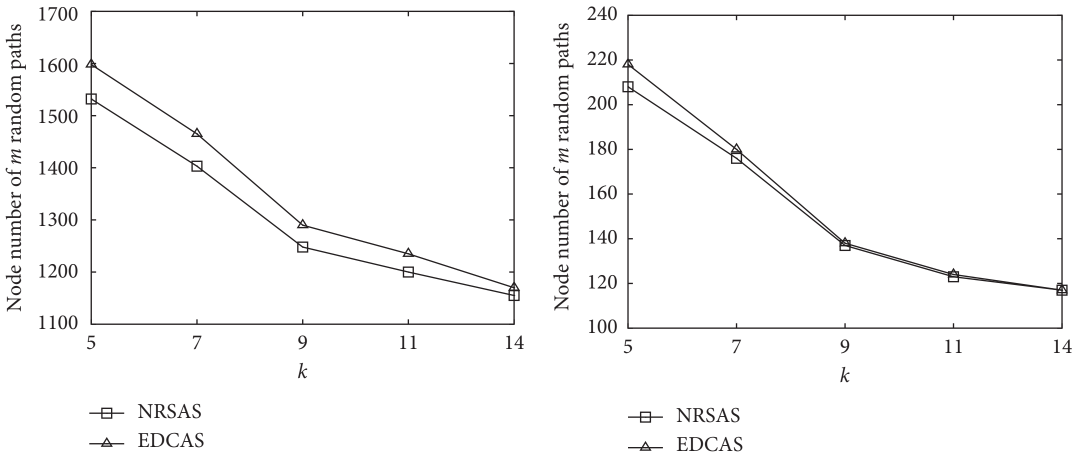


(a)

(b)

Figure 12: The effect of the communication radius on the random path. (a) Synthetic dataset. (b) IntelLab dataset.

As shown in Figure 15, the effect of the change of communication radius on the total number of random path nodes is demonstrated. The other parameters remain at their default values. It can be seen from the figure that, with the increase of the communication radius of the nodes, the total number of nodes that can be obtained by random paths in the two algorithms is increasing. The number of selected neighbor nodes to be selected in the data collection node increases, so the number of selectable next hop nodes increases, which provides more possibilities for the current node to select the next hop node. Then it is observed that, in the same communication radius range, the proposed efficient data collection algorithm based on ELM has better performance than the random selection algorithm in a certain communication radius range. It shows that the efficient data collection algorithm based on ELM position prediction has better performance than the random selection algorithm.



(a)

(b)

Figure 13: The effect of the sparsity on the random path. (a) Synthetic dataset. (b) IntelLab dataset.

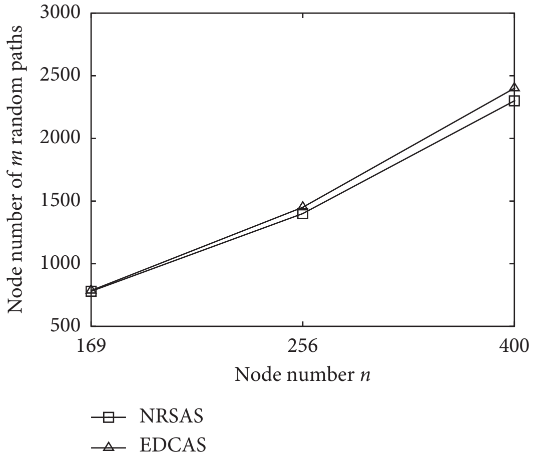
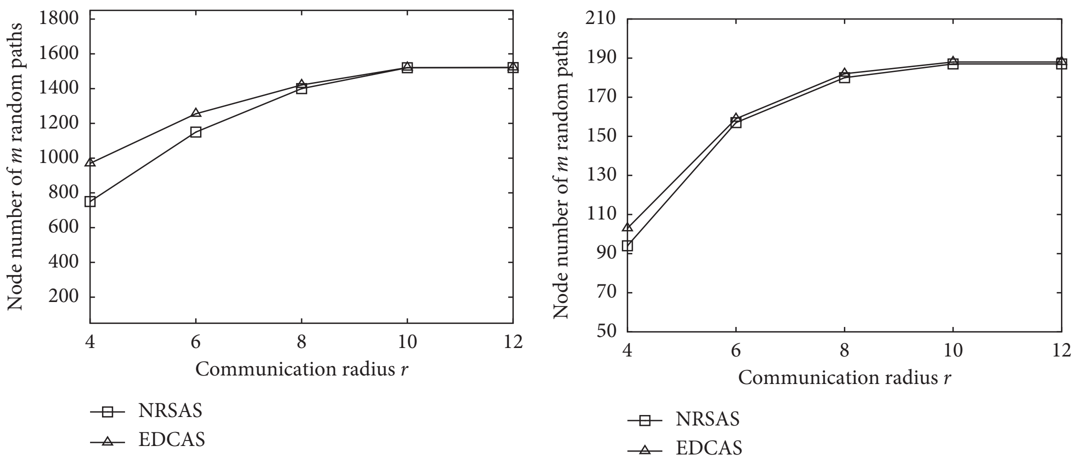


Figure 14: The effect of the node number on the random path.



(a)

(b)

Figure 15: The effect of the communication radius on the random path. (a) Synthetic dataset. (b) IntelLab dataset.

6. Conclusion

In this paper, the properties of compressed sensing in wireless sensor networks are analyzed, and the random walk path selection strategy is studied in depth. This paper proposes an efficient data collection algorithm in a static environment for the uncertain environment of the wireless sensor network and the problems that occur during the data collection process of a complex layout. Then, it is extended to propose an efficient data collection algorithm based on ELM location prediction in mobile network environment and makes a lot of simulation experiments to verify the correctness of the algorithm. Finally, the prototype system is designed to verify the correctness and effectiveness of the algorithm.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (no. 61602323), National Postdoctoral Foundation of China (no. 2016M591455), Youth Seedling Foundation of Liaoning Province (no. lnqn201913), and Natural Science Funds of Liaoning Province (nos. 2019MS264 and 20180550019).

References

[1] J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.

[2] J. Luo, L. Xiang, and C. Rosenberg, “Does compressed sensing improve the throughput of wireless sensor networks,” in *Proceedings of the 2010 IEEE International Conference on Communications*, pp. 1–6, Cape Town, South Africa, May 2010.

[3] S. Lee, S. Pattem, M. Sathiamoorthy, B. Krishnamachari, and A. Ortega, “Spatially-localized compressed sensing and routing in multi-hop sensor networks,” *GeoSensor Networks*, vol. 9, pp. 11–20, 2009.

[4] W. Wang, M. Garofalakis, and K. Ramchandran, “Distributed sparse random projections for refinable approximation,” in *Proceedings of the 6th international Conference on Information Processing in Sensor Networks*, pp. 331–339, ACM, Cambridge, MA, USA, April 2007.

[5] M. Rebai, M. Le berre, H. Snoussi, F. Hnaien, and L. Khoukhi, “Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks,” *Computers & Operations Research*, vol. 59, no. 3, pp. 11–21, 2015.

[6] D. Amine, B. Nassreddine, and K. Bouabdellah, “Energy efficient and safe weighted clustering algorithm for mobile wireless sensor networks,” *Procedia Computer Science*, vol. 34, pp. 63–70, 2014.

[7] O. Diallo, J. J. P. C. Rodrigues, M. Sene, and J. Lloret, “Distributed database management techniques for wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 2, pp. 604–620, 2015.

[8] M. Z. A. Bhuiyan, G. Wang, and A. V. Vasilakos, “Local area prediction-based mobile target tracking in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 64, no. 7, pp. 1968–1982, 2015.

[9] Y. G. Iyer, S. Gandham, and S. Venkatesan, “STCP: a generic transport layer protocol for wireless sensor networks,” in *Proceedings of the International Conference on Computer Communications and Networks, 2005. ICCCN 2005*, pp. 449–454, San Diego, CA, USA, October 2015.

[10] K. Ren, W. Lou, and Y. Zhang, “LEDS: providing location-aware end-to-end data security in wireless sensor networks,” *IEEE Transactions on Mobile Computing*, vol. 7, no. 5, pp. 585–598, 2008.

[11] S. Lin, F. Miao, J. Zhang et al., “ATPC,” *Acm Transactions on Sensor Networks*, vol. 12, no. 1, pp. 1–31, 2016.

[12] X. X. Ma and L. Peng, “Improved DV-Hop positioning algorithm for wireless sensor networks,” *Computer Engineering and Applications*, vol. 51, no. 21, pp. 97–101, 2015.

[13] H. Peng, H. Chen, X. Y. Zhang et al., “Location privacy protection technology in wireless sensor networks,” *Journal of Software*, vol. 26, no. 3, pp. 617–639, 2015.

[14] X. Z. Tian and S. Yang, “Optimization strategy for the survival time of bottleneck areas in wireless sensor networks based on network coding,” *Chinese Journal of Computers*, vol. 39, no. 5, pp. 1039–1050, 2016.

[15] Y. Q. Zhou and D. Z. Tan, “Application and research status of wireless sensor networks,” *Sensor World*, vol. 5, pp. 35–40, 2009.

[16] S. L. Marple, *Digital Spectral Analysis: With Applications*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1987.

[17] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, 2006.

[18] D. L. Donoho, “Compressed sensing,” *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[19] Y. Tsaig and D. L. Donoho, “Extensions of compressed sensing,” *Signal Processing*, vol. 86, no. 3, pp. 549–571, 2006.

[20] E. J. Candès, “Compressive sampling,” *Proceedings of the International Congress of Mathematicians*, European Mathematical Society, vol. 3, pp. 1433–1452, , Zürich, Switzerland, 2006.

[21] E. J. Candes and J. Romberg, “Quantitative robust uncertainty principles and optimally sparse decompositions,” *Foundations of Computational Mathematics*, vol. 6, no. 2, pp. 227–254, 2006.

[22] E. J. Candès, J. K. Romberg, and T. Tao, “Stable signal recovery from incomplete and inaccurate measurements,” *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.

[23] R. Q. Twiss, “Nyquist’s and Thevenin’s theorems generalized for nonreciprocal linear networks,” *Journal of Applied Physics*, vol. 26, no. 5, pp. 599–602, 1955.

[24] E. J. Candes and M. B. Wakin, “An introduction to compressive sampling,” *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[25] Y. Wang, Q. S. Lian, and K. Li, “MRI image reconstruction based on joint regularization and compressed sensing,” *Optical Technique*, vol. 36, no. 3, pp. 350–355, 2010.

[26] L. Tang, Z. Zhou, L. Shi, H.-P. Yao, and J. Zhang, “Energy balance based WSN compressive sensing algorithm,” *Journal of Electronics & Information Technology*, vol. 33, no. 8, pp. 1919–1923, 2011.

[27] C. H. Zhao and Y. L. Xu, “Energy-constrained Bayesian compressed sensing detection algorithm,” *Journal on Communications*, vol. 10, pp. 1–6, 2012.

[28] H. Zheng, S. Xiao, X. Wang, X. Tian, and M. Guizani, “Capacity and delay analysis for data gathering with compressive sensing in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 12, no. 2, pp. 917–927, 2013.

[29] H. Zheng, F. Yang, X. Tian, X. Gan, X. Wang, and S. Xiao, “Data gathering with compressive sensing in wireless sensor networks: a random walk based approach,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 1, pp. 35–44, 2015.

[30] X.-Y. Liu, Y. Zhu, L. Kong et al., “Compressive data collection for wireless sensor networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 8, pp. 2188–2197, 2015.

[31] Y. Yao, Q. Cao, and A. V. Vasilakos, “EDAL: an energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks,” *IEEE/ACM Transactions on Networking (TON)*, vol. 23, no. 3, pp. 810–823, 2015.

[32] M. Leinonen, M. Codreanu, and M. Juntti, “Sequential compressed sensing with progressive signal reconstruction in wireless sensor networks,” *IEEE Transactions on Wireless Communications*, vol. 14, no. 3, pp. 1622–1635, 2015.

[33] S. Xu, R. C. de Lamare, and H. V. Poor, “Distributed compressed estimation based on compressive sensing,” *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1311–1315, 2015.

[34] J. Yang, X. Liao, X. Yuan et al., “Compressive sensing by learning a Gaussian mixture model from measurements,” *IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society*, vol. 24, no. 1, p. 106, 2015.

[35] Z. Liu, Z. Li, M. Li et al., “Path reconstruction in dynamic wireless sensor networks using compressive sensing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 4, pp. 1948–1960, 2016.

[36] N. Eslahi and A. Aghagolzadeh, “Compressive sensing image restoration using adaptive curvelet thresholding and nonlocal sparse regularization,” *IEEE Transactions on Image Processing*, vol. 25, no. 7, pp. 3126–3140, 2016.