Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 16(5): 4526–4545.
DOI: 10.3934/mbe.2019226
Received: 11 March 2019
Accepted: 09 May 2019
Published: 22 May 2019

Research article

Reliable data transmission in wireless sensor networks with data decomposition and ensemble recovery

Fengyong Li, Gang Zhou*and Jingsheng Lei

College of Computer Science and Technology, Shanghai University of Electric Power, Shanghai, P.R.China

* **Correspondence:** Email: z15555152938@163.com.

Abstract: Wireless sensor networks (WSNs) are usually used to helps many basic scientific works to gather and observe environmental data, whose completeness and accuracy are the key to ensuring the success of scientific works. However, due to a lot of noise, collision and unreliable data link, data loss and damage in WSNs are rather common. Although some existing works, e.g. interpolation methods or prediction methods, can recover original data to some extent, they maybe provide an unsatisfactory accuracy when the missing data becomes large. To address this problem, this paper proposes a new reliable data transmission scheme in WSNs by using data decomposition and ensemble recovery mechanism. Firstly, the original data are collected by sensor nodes and then are expanded and split into multiple data shares by using multi-ary Vandermonde matrix. Subsequently, these data shares are transmitted respectively to source node via the sensor networks, which is made up of a large number of sensor nodes. Since each share contains data redundancy, the source node can reconstruct the original data even if some data shares are damaged or lost during delivery. Finally, extensive simulation experiments show that the proposed scheme outperforms significantly existing solutions in terms of recovery accuracy and robustness.

Keywords: wireless sensor networks; reliable transmission; data loss; matrix decomposition; ensemble recovery

1. Introduction

Wireless Sensor Networks (WSNs) is a multi-hop wireless network that is formed by self-organizing the sensor nodes of large-scale deployment. By WSNs, the data information of the perceived object in the monitoring area is collected and transmitted by a coordinated manner. Since WSNs has more advantages, e.g. strong environment adaptiveness, portable and energy efficient, they are widely used to collect information from the physical world and produce large-scale sensor data sets. Obviously, the

completeness and accuracy of these awareness data sets significantly affect the reliability of scientific results. If these data are lost or jumping during transmission, it will inevitably lead to the unreliable or error results. Therefore, the completeness and accuracy for scientific data are so important in decision-making. Nevertheless, in actual data collection scenario, data loss is so common. The reasons can be summarized as follows: (1) wireless channel instability and noise interference, (2) mutual interference between channels caused by tree or clustered topology, (3) congestion caused by data bursts in high-density deployments or emergencies, (4) unexpected failure due to node damage or battery problem. The mentioned cases maybe cause serious errors in receiving end and finally lead to invalidation of some scientific research and engineering calculations.

Although the importance of missing data in wireless sensor networks is very prominent, the research on this problem is still relatively rare. Some researchers have tried to reconstruct the missing data by using some typical interpolation algorithms [1], which always appear in the field of database [2], multimedia [2], and signal processing [3]. For example, as a classical local interpolation method, k-Nearest-Neighbor (kNN) [4] is usually used to estimate the missing data according to the nearest k neighbors around the missing data position. This scheme can achieve a good estimation performance due to high correlation between adjacent data, and is thus used in low-fidelity estimation cases. Delaunay Triangulation (DT) [5] is another typical global interpolation method. This method considers each collected data as vertices, which are connected into triangles according to the process of gradually reducing the global error, and the missing values are finally inserted into the data set. Multi-channel Singular Spectrum Analysis (MSSA) [6] is a nonparametric adaptive method, which belongs to the category of principal component analysis. This method uses embedded self-covariance matrix to insert missing values. Compressive Sensing (CS) method [7, 8] was proposed in 2006, which is an advanced data recovery algorithm. If the data set is sparse, CS method can efficiently estimate the whole data set by using very little known data. Therefore, this method has attracted much research interest. On the basis of CS scheme, Kong et al. proposed an improved algorithm [9], named by Environmental Space Time Improved Compressive Sensing (ESTI-CS), which embeds customized features into the baseline CS to deal with the specific data loss patterns and then uses a multi-attribute assistant (MAA) component to perform data reconstruction. Chen et al. proposed a novel data reconstruction scheme via temporal stability guided matrix completion [10]. They formulate the data reconstruction problem as a matrix completion with structural noise and further reduce the reconstruction error by introducing a constraint about short-term stability to the matrix completion problem.

Although the existing schemes can work for the data recovery and reconstruction, they can not completely solve the data loss in the wireless sensor network due to the following reasons:

- The data loss mechanism in wireless sensor networks has some special characteristics, which often do not meet the assumptions of classical interpolation algorithms.
- Existing schemes always reconstruct missing data by interpolation or prediction, which makes that some predicted data are not exactly consistent with the original data. Therefore, they cannot be applied in some cases that needs high data accuracy.
- Existing schemes can recover original data only when the missing data is small. In other words, they maybe provide an unsatisfactory accuracy when the missing data becomes large.

Overall, existing methods always provide the estimated value for the loss data. This mechanism is usually efficient to small-scale insensitive environmental data, but for large-scale and high accuracy

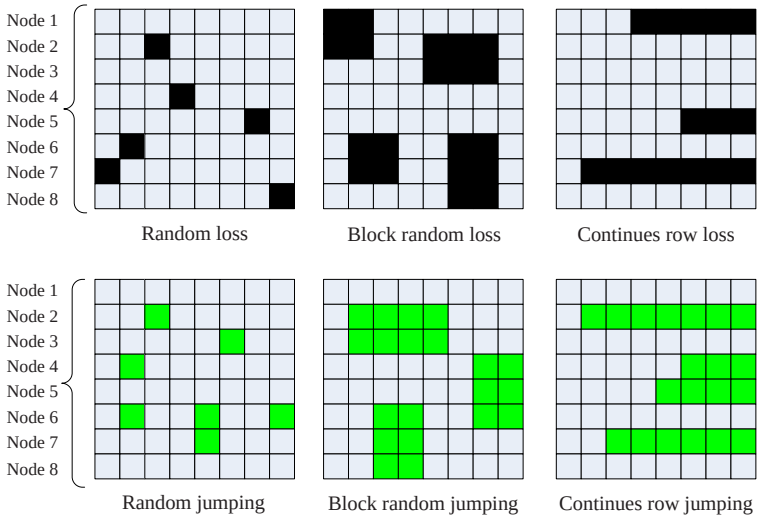

Figure 1. The data loss and jumping pattern in wireless sensor networks. Each block represents the transmission data of one sensor node. The top subfigures present the data loss pattern and the black blocks stand for loss data, while the bottom subfigures present the data jumping pattern and the green blocks stand for the jumping data.

data, it may not work well anymore. Facing the aforementioned problems, this paper designs a data decomposition and ensemble recovery mechanism, and tries to fundamentally solve the data loss and jumping in the data transmission for wireless sensor networks. Comparing with the previous works, we make the following novel contributions.

- We propose a reliable data transmission scheme by designing data decomposition and ensemble recovery mechanism. We claim that proposed scheme can solve the data loss problem in WSNs, and guarantee the correctness of original data even if some data are lost or damaged in delivery.
- We use matrix decomposition to design data expansion mechanism, and then split the original data into multiple data shares, which are transmitted through multiple sensor nodes. Since each data share contains some data redundancy, the loss of part of the shares does not affect the recovery of original data.
- Ensemble recovery mechanism is designed to reconstruct the missing data. This mechanism can not only recover the missing data, but can revise the incorrect data if they occurs jumping in transmission.
- Comprehensive simulation experiments are performed by the WSNs including multiple sensor nodes. The experimental results demonstrate that proposed scheme can reconstruct the missing data perfectly, and significantly superior than the existing interpolation schemes.

The rest of this paper is organized as follows. Section 2 provides the details of proposed scheme by constructing the loss model and introducing the procedure of data decomposition and ensemble reconstruction. Subsequently, a series of simulation experiments are performed to evaluate the performance of proposed scheme. The experimental results and corresponding discussions are presented in Sections 3. Finally, Section 4 concludes the paper.

2. Data unreliable transmission pattern in wireless sensor networks

In this section, we discuss the data loss pattern in WSNs. For data unreliable transmission problem, we always intuitively think that the data is randomly lost. However, for WSNs, this problem becomes rather particular. For example, when a sensor node encounters failure, it will continuously lose data, while other nodes may not generate data loss. In general, in terms of the nature of WSNs, we can summarize two typical data loss patterns as follows.

Data loss pattern. Since WSNs include multiple sensor nodes and each node has special function and characters, data loss pattern usually contains random loss, block random loss and continues row loss. Random loss is always caused by the noise and collision in WSNs. The missing data is evenly distributed in the data matrix. Block random loss may result from large-scale data congestion caused by unexpected events. This case usually happens in high density sensor nodes. Continues row loss presents the case that the whole data is lost from a certain location to the end of this row. This pattern may caused by battery exhausted or accidentally damaged for sensor node. The subfigures on the top of Figure 1 show the cases of data loss pattern.

Data jumping pattern. Data jumping pattern is similar to data loss pattern and also includes three case, random jumping, block random jumping and continues row jumping. In fact, data jumping pattern explains the case that when data is transmitted, single digit may be changed to another one due to signal interference. For example, 1 is changed to 0, or vice versa. The subfigures on the bottom of Figure 1 give the cases of data jumping pattern.

3. Proposed reliable data transmission scheme

3.1. The framework of proposed scheme

Proposed data transmission scheme is comprised of three parts: data decomposition, data delivery and data ensemble recovery. In the first part, aiming at each sensor node, the original perceived data can be split into multiple data shares by matrix decomposition mechanism. These data share are sent to the the next sensor node in sequence. In the second part, each data share is delivered into complex sensor networks. They may counter noise and channel interference or sensor node fault so that some data are lost or damaged in delivery. In the third part, according to received data shares, we design ensemble mechanism to recover the original perceived data even if received data is lost a lot or contains many digital jumping. The framework of our proposed scheme is shown in Figure 2.

3.2. Perceived data decomposition

In WSNs, the environmental data are perceived by sensor node and then sent to the networks in sequence. Then, multiple wireless node transmit the data by relaying, and finally deliver them to the source node . At the source node, the receiver (computer or processing center) reconstruct the original data according to a fixed order. Unfortunately, these perceived data may be attacked/damaged during transmission, such as the channel interference, network noise or sensor node fault. In this case, it is unreasonable to assume that the source node can receive the information completely and accurately. Once the data are lost, it may cause serious problems due to the incomplete of perceived data. Although researchers have tried many interpolation method, the incomplete and incorrect problem of perceived

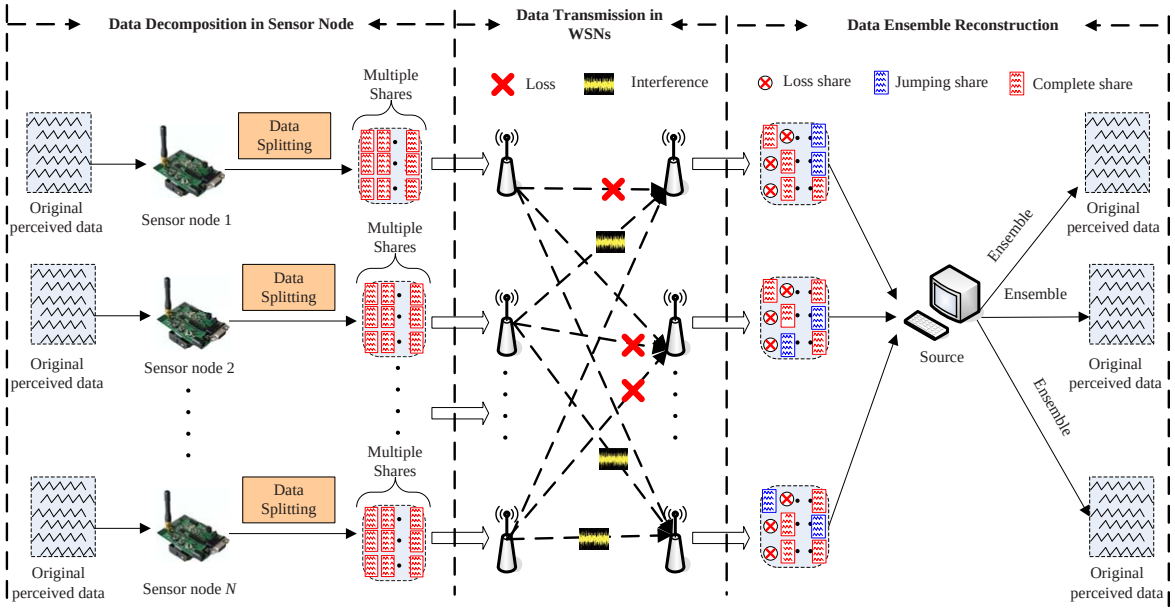

Figure 2. The framework of proposed data transmission scheme.

data are still difficult to solve.

To improve the robustness of perceived data transmission in WSNs, in this section, we try to use matrix decomposition mechanism [11, 12] to preprocess the perceived original data. Specifically, we are inspired by data sharing. The original perceived data are expanded and divided into multiple data shares. According to the idea of data sharing, each share only contains a small portion of valid data, partial share loss do not affect the recovery for original perceived data. The corresponding details are provided as follows.

First, we assume that perceived data is always a decimal number or digital string. To divide the original data, they are transmitted firstly to a binary stream, and then are presented as q-ary symbol system, where q is an odd prime. We can use a simple procedure to process this presentation. For example, the binary stream are segmented into multiple pieces and each of them includes L_1 bits. According to the following equation, we can convert these L_1 bits to L_2 q-ary digits .

$$L_1 = \lfloor L_2 \cdot \log_2 q \rfloor . \tag{3.1}$$

Assume that $L_1 = 4$, $L_2 = 2$. If we convert the original data into 5-ary notational system, we can use the following simple sample to explain this procedure.

$$\begin{matrix} \text{2-ary} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix} \end{matrix} \Rightarrow \begin{matrix} \text{5-ary} \\ \begin{bmatrix} 23 \\ 11 \\ 14 \end{bmatrix} \end{matrix} \tag{3.2}$$

The relationship between L_1 and L_2 can be represented by the following equation.

$$r = 1 - \frac{L_1}{L_2 \cdot \log_2 q} \tag{3.3}$$

Obviously, if L_1 and L_2 are very large, we can determine that the parameter r is close to 0.

Subsequently, we integrate all q-ary digits as a digital matrix $\mathbf{D}$. The detailed data decomposition procedure can be explained by the following steps.

Step 1 : Divide $\mathbf{D}$ into K vectors, $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, \cdots, \mathbf{d}_K\}$, each of them can be represented as follows.

$$\mathbf{d}_k = \{\mathbf{d}_{k,1}, \mathbf{d}_{k,2}, \cdots, \mathbf{d}_{k,m}\} \tag{3.4}$$

where m represents that each vector contains the number of q-ary digits and $k \in [1, K]$.

Step 2 : Select $a_1, a_2, \cdots, a_n$ as a set of indices and use them to construct a q-ary Vandermonde matrix $\mathbf{A}$ with the size of $m \times n$.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ a_1 & a_2 & \cdots & a_n \\ a_1^2 & a_2^2 & \cdots & a_n^2 \\ \vdots & \vdots & \cdots & \vdots \\ a_1^{m-1} & a_2^{m-1} & \cdots & a_n^{m-1} \end{bmatrix} \bmod q \tag{3.5}$$

where $a_1, a_2, \cdots, a_n \in [0, q-1]$ are different with each other. With the core of Vandermonde matrix, m, n, and q must satisfy $m \leq n \leq q$.

Step 3 : According to the Step 1 and Step 2, each digital block $\mathbf{d}_k$ can be divided into n shares, which are denoted as a q-ary digital vector $\mathbf{t}_k$.

$$\mathbf{t}_k = \begin{bmatrix} d_{k,1} & d_{k,2} & d_{k,3} & \cdots & d_{k,m} \end{bmatrix} \cdot \mathbf{A} \tag{3.6}$$

where $\mathbf{t}_k = [t_{k,1} \ t_{k,2} \ t_{k,3} \ \cdots \ t_{k,n}]$ includes n q-ary digits, and the symbol "$\cdot$" is the multiplication operator in q-ary notational system.

We can use a simple example to explain data decomposition procedure. Assume that $q = 7$, $n = 6$, $m = 3$, the original perceived data are three 7-ary digits [2 1 4], and the indices of Vandermonde matrix are fixed as $[a_1 a_2 \cdots a_n]$ = [5 3 1 0 2 4]. According the Equation (3.5), the Vandermonde matrix can be built as follows.

$$\mathbf{A} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 3 & 1 & 0 & 2 & 4 \\ 4 & 2 & 1 & 0 & 4 & 2 \end{bmatrix} \tag{3.7}$$

We use the Equation (3.6) to calculate the expanded data vector $\mathbf{t}_k$ = [2 6 0 2 6 0], which subsequently is sent into the WSNs by the sensor node.

$$\begin{bmatrix} 2 & 1 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 5 & 3 & 1 & 0 & 2 & 4 \\ 4 & 2 & 1 & 0 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 6 & 0 & 2 & 6 & 0 \end{bmatrix} \bmod q \tag{3.8}$$

3.3. Data ensemble recovery

According to the data decomposition procedure, the expanded data will be sent into the WSNs. Unfortunately, since the wireless sensor nodes always face the harsh environments. If they are delivered through insecure network channel, it might encounter data loss or jumping due to diverse network faults. Following the formula of data decomposition, the source node can reconstruct the original perceived data by an ensemble mechanism only if it can receive enough expanded data.

3.3.1. Data recovery for loss pattern

Assume that the source node only receive correctly a part of expanded data $\mathbf{t}'_k = [t_{k,1}\ t_{k,2}\ t_{k,3}\ \cdots\ t_{k,n'}]$, $m \le n' \le n$, and the other data is lost. In that way, as long as the loss number for original data vector $\mathbf{t}_k$ is not more than $n - m$, the source node can reconstruct original perceived data by the following equation.

$$\begin{bmatrix} d_{k,1} & d_{k,2} & d_{k,3} & \cdots & d_{k,m} \end{bmatrix} = \begin{bmatrix} t'_{k,1} & t'_{k,2} & t'_{k,3} & \cdots & t'_{k,m} \end{bmatrix} \cdot (\mathbf{A}')^{-1} \tag{3.9}$$

where $t'_{k,1}, t'_{k,2}, t'_{k,3}, \cdots, t'_{k,m}$ are m received digits selecting randomly from the remaining digits (Here, we suppose the remaining digits do not contain the jumping case). In addition, $\mathbf{A}'$ is a $m \times m$ Vandermonde matrix built by the indices $a'_1, a'_2, \cdots, a'_m$ that correspond to one-to-one with the $t'_{k,1}, t'_{k,2}, t'_{k,3}, \cdots, t'_{k,m}$. $\mathbf{A}'^{-1}$ is the inversion matrix of $\mathbf{A}'$ in q-ary notational system and we can find the derivation process of the inversion matrix in [12].

Obviously, the case for data loss can be also explained by a simple example. Following the actual example shown in Section 3.2. Assume that the received expanded data are [2 6 2 0] (the complete expanded data [**2 6** 0 **2** 3 **0**]). We randomly select 3 digits [2 6 2] and their corresponding indices is [5 3 0] (the complete indices [**5 3** 1 **0** 2 4]). Accordingly, we can build easily the Vandermonde matrix $\mathbf{A}'$ by Equation (3.5) and also calculate the inversion matrix easily based on the derivation in [12].

$$\mathbf{A}' = \begin{bmatrix} 1 & 1 & 1 \\ 5 & 3 & 0 \\ 4 & 2 & 0 \end{bmatrix} \bmod q \Rightarrow (\mathbf{A}')^{-1} = \begin{bmatrix} 0 & 6 & 5 \\ 0 & 2 & 1 \\ 1 & 6 & 1 \end{bmatrix} \bmod q \tag{3.10}$$

The original perceived data can be calculated easily by Equation (3.11).

$$\begin{bmatrix} 2 & 6 & 2 \end{bmatrix} \cdot \begin{bmatrix} 0 & 6 & 5 \\ 0 & 2 & 1 \\ 1 & 6 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 4 \end{bmatrix} \bmod q \tag{3.11}$$

3.3.2. Data recovery for jumping pattern

Similarly, we also assume that the source node receive a part of expanded data $\mathbf{t'}_k = [t_{k,1}\ t_{k,2}\ t_{k,3}\ \cdots\ t_{k,n'}]$, $m \le n' \le n$. The received data may suffer some jumping (more or less), but we can not determine which data are changed. Although the jumping number for $\mathbf{t'}_k$ is not determined, the source node can also recover original perceived data by ensemble mechanism, which is explained detailed in **Algorithm 1**. Actually, the data jumping case that we discuss here includes the data loss case.

Algorithm 1: Ensemble Recovery for Data Jumping

Input: $\mathbf{t'}_k = [t_{k,1}\ t_{k,2}\ t_{k,3}\ \cdots\ t_{k,n'}]$, the corresponding indices $a'_1, a'_2, \cdots, a'_{n'}$ for $\mathbf{t'}_k$, multi-ary parameter q, ensemble rounds R.

Output: Original data vector $\mathbf{d}_k = \{d_{k,1}, d_{k,2}, \cdots, d_{k,m}\}$.

for $i \leftarrow 1$ **to** R **do**
  Randomly select m digits from $\mathbf{t'}_k$ as $\mathbf{T'}_i = \{t'_{k,1}, t'_{k,2}, t'_{k,3}, \cdots, t'_{k,m}\}$, and then denote their corresponding indices as $\{a'_1, a'_2, \cdots, a'_m\}$.
  Using the indices $\{a'_1, a'_2, \cdots, a'_m\}$ to build Vandermonde matrix $\mathbf{A'}$.
  Solve the inverse matrix $\mathbf{A'}^{-1}$ for $\mathbf{A'}$.
  $\mathbf{d}_i = \mathbf{T'}_i \times \mathbf{A'}^{-1} \bmod q$.
end
$\mathbf{d}_k = \text{MajorityVoting}(\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3, \cdots, \mathbf{d}_R)$.

In order to understand the ensemble mechanism easily, we also use the actual example proposed in Section 3.3.1. According to the complete expanded data [2 **6** 0 2 6 0], we assume that the last digit is lost and the second digit is jumping in transmission (take '6' to '4' as an example). Thus, the received data are [2 **4** 0 2 6]. Since the source node does not know which digits are jumping, the ensemble mechanism with majority voting is used to decide the correct original data. An actual ensemble example is shown in Figure 3.

We would like to remind readers that if we only consider the case of data loss in WSNs, we can provide an effective solution by the description in Section 3.3.1. However, if the received data contains both cases, data loss and data jumping, then we must use the ensemble reconstruction mechanism. In practice, we always use ensemble mechanism to handle all situations because we can't confirm whether the received data contains data jumping.

3.4. Algorithm analysis

According to the data decomposition mechanism, we can easily expand m q-ary digits to n q-ary digits. In other words, n q-ary digits contain $n - m$ redundancy digits and the redundancy rate, named as R_e in this paper, can be calculated easily by Equation (3.12). Obviously, two parameters m and n have an important impact for redundancy rate. If the data lost rate is more than R_e, the original data will be hard to recover. If the data lost rate is less than R_e (assume that the number of received data are n'), we can randomly select m digits from n' to recover the original data.

$$R_e = 1 - \frac{m}{n}. \tag{3.12}$$

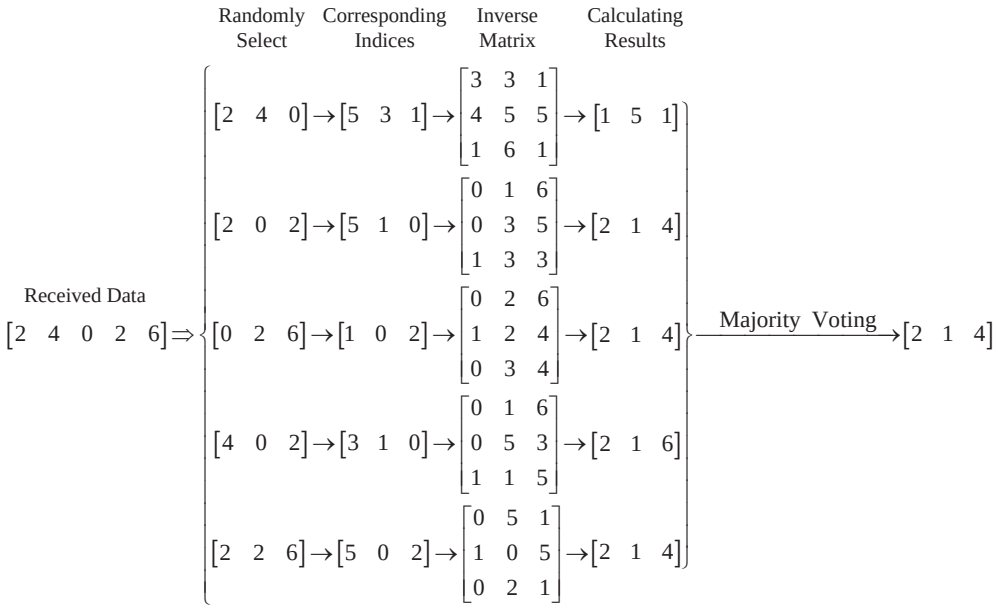

Figure 3. An example for data ensemble recovery processing. In this example, we use five rounds ensemble in 7-ary notational system ($R = 5$ in **Algorithm 1**). The complete expanded digits are [2 6 0 2 6 0] and the received digits are [2 4 0 2 6]. Since $m = 3$ and $n = 6$, each experiment can randomly select 3 digits from the received digits.

In addition, we would like to stress that the proposed ensemble reconstruction mechanism will be used more generally to recover the original perceived data, because we do not know whether the received data contain the jumping digits even if there is not any data loss. Nevertheless, we also remind that ensemble reconstruction mechanism only make an accurate decision with greater probability and can not thus guarantee that the original data is recovered perfectly. Similar to data loss pattern, if the data jumping rate is more than R_e, ensemble mechanism does not work.

In fact, we can calculate data recovery probability theoretically. Assume that we transmit n digits through WSNs, and in the receiving end, p_1 digits are lost and p_2 digits are jumping. The recovery probability T_1 can be calculated as following when the ensemble mechanism is not used.

$$T_1 = \frac{C^m_{n-p_1-p_2}}{C^m_{n-p_1}} \tag{3.13}$$

When the ensemble mechanism is used, the recovery probability T_2 is

$$T_2 = 1 - \left[\left(1 - \frac{C^m_{n-p_1-p_2}}{C^m_{n-p_1}}\right)^R + R \cdot \frac{C^m_{n-p_1-p_2}}{C^m_{n-p_1}} \cdot \left(1 - \frac{C^m_{n-p_1-p_2}}{C^m_{n-p_1}}\right)^{R-1}\right] \tag{3.14}$$

where R is the ensemble rounds (corresponding to the parameter R in **Algorithm 1**). In all the ensemble results, the result with the maximum number of occurrences will be decided as the correct original data.

Table 1. The relationship between parameters (m, n) and redundancy rate R_e.

m	n	q	R_e
5	7	11	28.57%
5	11	13	54.55%
5	31	37	83.87%
5	101	103	95.05%
5	991	997	99.50%

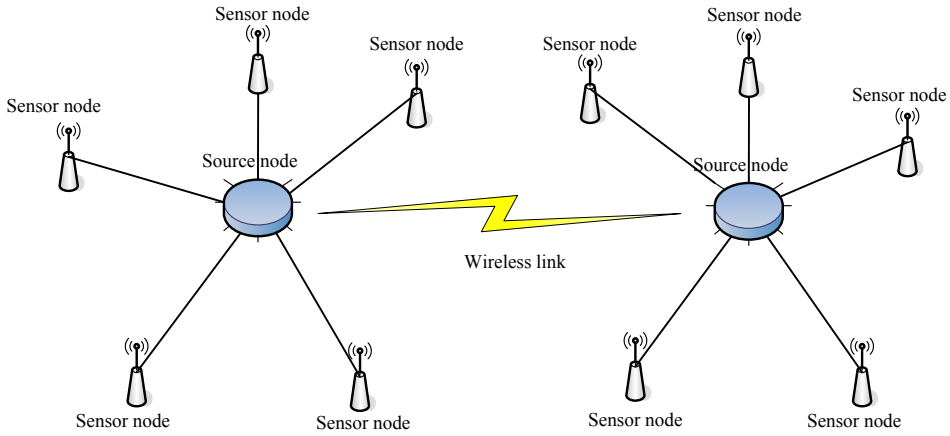

Figure 4. The topology of sensors networks in our simulation experiments.

4. Simulation results and discussion

In this section, we validate the proposed scheme by simulating a wireless sensor network, which include 30 sensor nodes and one source node. We assume that the original data sent by each node is already decomposition data[*]. These data are delivered by multiple sensor nodes to source node and some of them may be lost in this procedure, our goal is to reconstruct original data.

To simulate the proposed scheme, we distribute 30 wireless sensor nodes in a $100m \times 100m$ area. The 30 wireless sensor nodes and source node are intergraded to form a wireless network with star topology[†], which is shown in Figure 4. We design a series of experiments by randomly select sensor node as sending end. The selected node sends the decomposition data to source node through multiple nodes' delivery. Some data may be lost in this transmission, then the source node will reconstruct the original data by proposed algorithm if the loss rate is not more than its limitation.

We repeat each experiment 100 times to obtain an average result, which is considered as the overall performance evaluation. Each time, we select a different sensor node as the sending end, which maybe send different original data.

[*]In fact, to ease comparison, we can set a fixed decomposition matrix in all sensor nodes.

[†]In this paper, the actual network protocol in the wireless sensor networks is not considered. To ease understand, we use a simple self-organizing protocol and secure scheme [15] to simulate wireless network scenario.

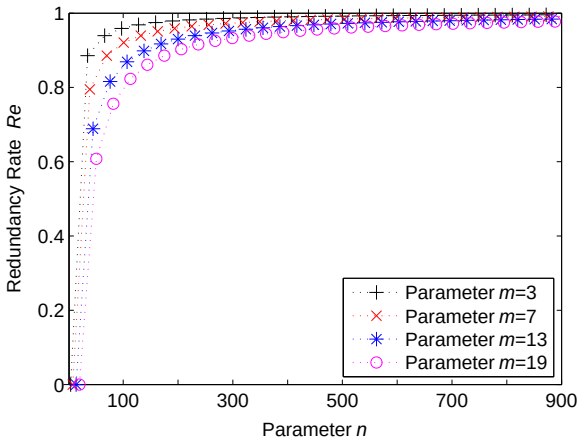

Figure 5. The relationship between redundancy rate R_e and two parameters (m, n). We test four different values, $m = 3$, $m = 7$, $m = 13$, $m = 19$ with the condition $m \leq n \leq q$.

4.1. Reliability analysis for proposed scheme

In this section, we analyze the reliability of proposed data transmission scheme. According to the idea of proposed scheme, the Vandermonde matrix is used to expand m original data to n shares, that is also to say, n expanded data carry m original data. Actually, it is similar to the data sharing mechanism. In this mechanism, each expanded data contains $\frac{n-m}{n}$ redundancy data and the redundancy rate R_e can be also calculated by Equation (3.12).

Apparently, the redundancy rate R_e depends on the parameters m and n and can be up very high if there is an extreme gap between m and n. Table 1 shows the relationship between redundancy rate R_e and parameters m and n. By fixing parameter m, the redundancy rate R_e will gradually raise with n increasing. This conclusion has been reflected theoretically by Figure 5.

4.2. Performance testing for proposed scheme

4.2.1. Testing for data loss pattern

In this section, we test the recovery performance of proposed scheme for data loss pattern. Since data completeness is very easy to be checked by the source node (we suppose that the decomposition matrix is known by source node.), when source node receives all the data shares, it can check quickly whether the lost data exceeds the redundancy rate R_e. Table 2 provides the original data recoverability for different expanded parameters m and n. In this table, we set $m = 5$ and $q = 37$. The data loss rates are fixed to five levels, which are 10%, 30%, 50%, 70%, 80%. As can be seen from the table, when data loss is not more than the redundancy rate R_e, the original data can be reconstructed perfectly by source node. On the contrary, the original data can not be recovered once the received data rate is less than m/n. Actually, this conclusion can be also validated theoretically by the reliability analysis in Section 4.1.

In addition, we also test the data recoverability only for data loss by comparing proposed scheme with the k-NearestNeighbor (kNN) scheme, which is also illustrated as classical traditional reconstruction method for data loss pattern. For any loss data, kNN scheme uses the k nearest neighbors to

estimate it, and always replace the loss data with the majority of the k nearest neighbors. Note that the direct comparison is rather hard, because most of traditional data reconstruction schemes only provide a estimated value for the lost data, that is also to say, there is a slight gap between reconstruction value and actual value for kNN scheme, while proposed scheme can definitely give an exact value if the loss rate is not more than R_e. Therefore, we need to build a fair comparison between proposed scheme and kNN scheme by setting a recovery error α in our following experiment. When the absolute of recovery error falls into a certain range $(0,\ \alpha)$, the reconstruction data is considered to be valid. Thus, the data recoverability (DR) can be calculated as follows.

$$\mathrm{DR} = \frac{\text{Number of correct data}}{\text{Total number of data}} \tag{4.1}$$

To gain more insight, we imitate four seasons to simulate the real sensor network, and set the sensor nodes to perceive the environment temperature, Spring ($10 - 22$°C), Summer ($22 - 37$°C), Autumn ($10 - 22$°C), and Winter ($0 - 10$°C). For kNN scheme, the parameter k is set to $k = 3$ and five distance measures are used to give the comparison results. Moreover, the recovery error parameter α is fixed as 0.5 and 1.0, respectively. This indicates that the error ranges are set to $[-0.5, 0.5]$ and $[-1, 1]$. Each experiment, we repeat 100 times to give the average DR value. The experimental results are shown in Figure 6 and Figure 7. In these figures, the x-axis represents data lost rate, while the y-axis denotes the data recoverability DR. It is easy to observe that for proposed scheme, when the data loss rate is lower than the theoretical redundancy rate R_e, the overall DR values can reach 100%. However, when the data loss rate is more than the theoretical redundancy rate $R_e = 73.68\%$ ($m = 5$, $n = 19$, $q = 37$), proposed scheme will not work well. In addition, for kNN method, the overall DR values are tending towards decreasing slowly with an increasing data loss rate. This is because there is a data estimation processing for kNN scheme, if the data loss rate becomes high, the valid data might be rather sparse so that the gap between reconstruction value and actual value becomes bigger and bigger, because it is very difficult to find k nearest neighbors with similar value.

Moreover, we also test the data recoverability for kNN scheme with different error ranges $[\alpha, \alpha] \in \{[-0.5, 0.5], [-1.0, 1.0], [-2.0, 2.0], [-3.0, 3.0]\}$. In this experiment, kNN scheme uses five distance measures, Euclidean, Manhattan, Chebyshev, Angle cosine and Hamming, to give the comparison results, which are shown in Table 3. We can see that when the recovery error range $[-\alpha, \alpha]$ becomes big, the data recoverability is slightly stronger. In fact, this phenomenon is explained easily. When α has a large range, more estimation data may be involved in *valid data*, leading to a higher data recovery rate. Also, we can also observe that proposed scheme consistently provide a perfect recoverability as long as the data loss rate is lower than R_e, but, when the data loss rate is more than theoretical redundancy rate R_e, e.g. 80%, proposed scheme does not work.

4.2.2. Testing for data jumping pattern

In this section, we discuss the data jumping pattern for unreliable transmission in WSNs. Essentially, the data jumping pattern should consider the combination of data loss and data jumping, because the network transmission always include these two cases, and the source node in WSNs is hard to identify the data jumping, but is easy to detect the data loss.

We show the advantages of proposed by simulating a sensor network and test data recoverability in the context of ensemble reconstruction ($R > 1$) and single reconstruction (R=1). We use the 30 sensor

Table 2. Data recoverability for different redundancy rate R_e. Data loss rates are fixed to five levels, which are 10%, 30%, 50%, 70%, 80%, and the parameters m and q are fixed as 5 and 37, respectively.

n	R_e	Data loss rate				
		10%	30%	50%	70%	80%
9	44.44%	Yes	Yes	No	No	No
14	64.29%	Yes	Yes	Yes	No	No
19	73.68%	Yes	Yes	Yes	Yes	No
24	79.17%	Yes	Yes	Yes	Yes	No
29	82.76%	Yes	Yes	Yes	Yes	Yes

Table 3. Data recoverability for proposed scheme and kNN scheme using five measure methods. Four error ranges $[\alpha, \alpha]$, [-0.5, 0.5], [-1.0, 1.0], [-2.0, 2.0], [-3.0, 3.0], are tested in this experiment. Data loss rates are fixed to five levels, which are 10%, 30%, 50%, 70%, 80%, and the parameters m and q are fixed as 5 and 37 (corresponding to $R_e = 73.68\%$), respectively.

Scheme	Measure	$[-\alpha, \alpha]$	Data loss rate				
			10%	30%	50%	70%	80%
kNN	Euclidean	[-0.5, 0.5]	0.878	0.756	0.556	0.400	0.322
		[-1.0, 1.0]	0.967	0.878	0.767	0.789	0.711
		[-2.0, 2.0]	0.978	0.967	0.944	0.944	0.944
		[-3.0, 3.0]	1.000	1.000	1.000	1.000	1.000
	Manhattan	[-0.5, 0.5]	0.900	0.756	0.622	0.444	0.411
		[-1.0, 1.0]	0.956	0.900	0.789	0.778	0.744
		[-2.0, 2.0]	0.978	0.967	0.944	0.944	0.944
		[-3.0, 3.0]	1.000	1.000	1.000	1.000	1.000
	Chebyshev	[-0.5, 0.5]	0.911	0.733	0.600	0.467	0.400
		[-1.0, 1.0]	0.967	0.933	0.811	0.800	0.722
		[-2.0, 2.0]	0.978	0.967	0.944	0.944	0.944
		[-3.0, 3.0]	1.000	1.000	1.000	1.000	1.000
	Angle cosine	[-0.5, 0.5]	0.878	0.733	0.611	0.489	0.422
		[-1.0, 1.0]	0.944	0.900	0.722	0.711	0.656
		[-2.0, 2.0]	1.000	0.956	0.956	0.922	0.900
		[-3.0, 3.0]	1.000	0.989	0.978	0.967	0.978
	Hamming	[-0.5, 0.5]	0.867	0.744	0.600	0.389	0.311
		[-1.0, 1.0]	0.911	0.811	0.611	0.589	0.578
		[-2.0, 2.0]	0.956	0.800	0.722	0.700	0.622
		[-3.0, 3.0]	0.978	0.911	0.856	0.800	0.778
Proposed	-	-	1.000	1.000	1.000	1.000	0

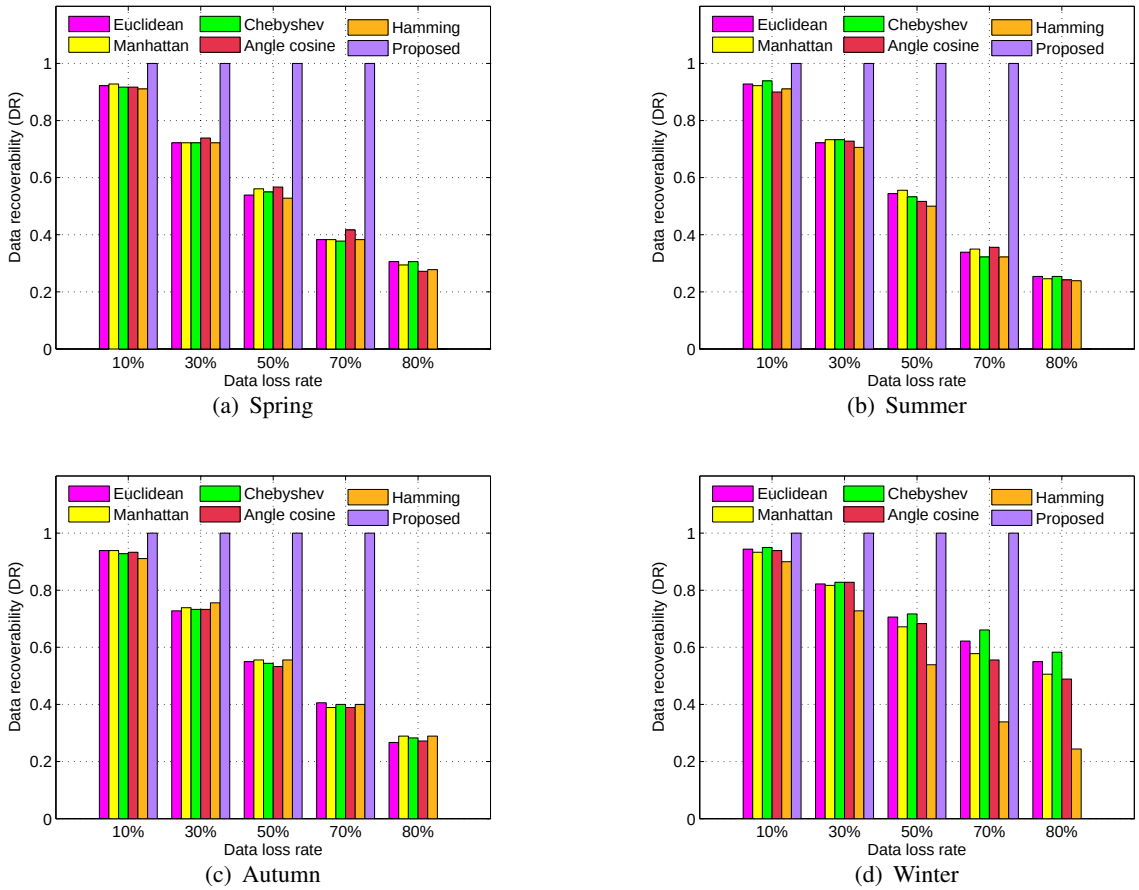

Figure 6. Correlation between data recoverability (DR) and different data lost rate (%) by comparing kNN with five distance measures (Euclidean, Manhattan, Chebyshev, Angle cosine and Hamming) and proposed scheme. The recovery error is $\alpha = 0.5$. For kNN method, the parameter k is set to $k = 3$.

nodes and one source node to build the simulation scenario. The decomposition matrix is calculated through fixing three parameters $m = 7$, $n = 30$, and $q = 37$, and five jumping rates, 10%, 20%, 30%, 40%, and 50%, are considered in this experiment. In order to build a fair comparison, we fix data loss rate as 10% for each experiment, and give the comparison experimental results by using ensemble recovery and single recovery. The experiments are repeated 100 time.

Table 4 shows the corresponding comparison results. As can be seen in this table, the overall data recoverability for the case using ensemble recovery is always higher than the case using single recovery, no matter what the data jumping rate is. We can explain this phenomenon as follows. For source node, when the data loss rate is less than theoretical redundancy rate R_e, it is believed that the original data can be reconstructed perfectly. Actually, the source node can not identify the jumping data, e.g. decimal digit 6 ('110' for binary) changing to decimal digit 4 ('100' for binary). If the source node use single recovery to reconstruct data, it may select a data share including jumping digits. Finally, it results in an error recovery. On the contrary, ensemble mechanism can largely avoid this

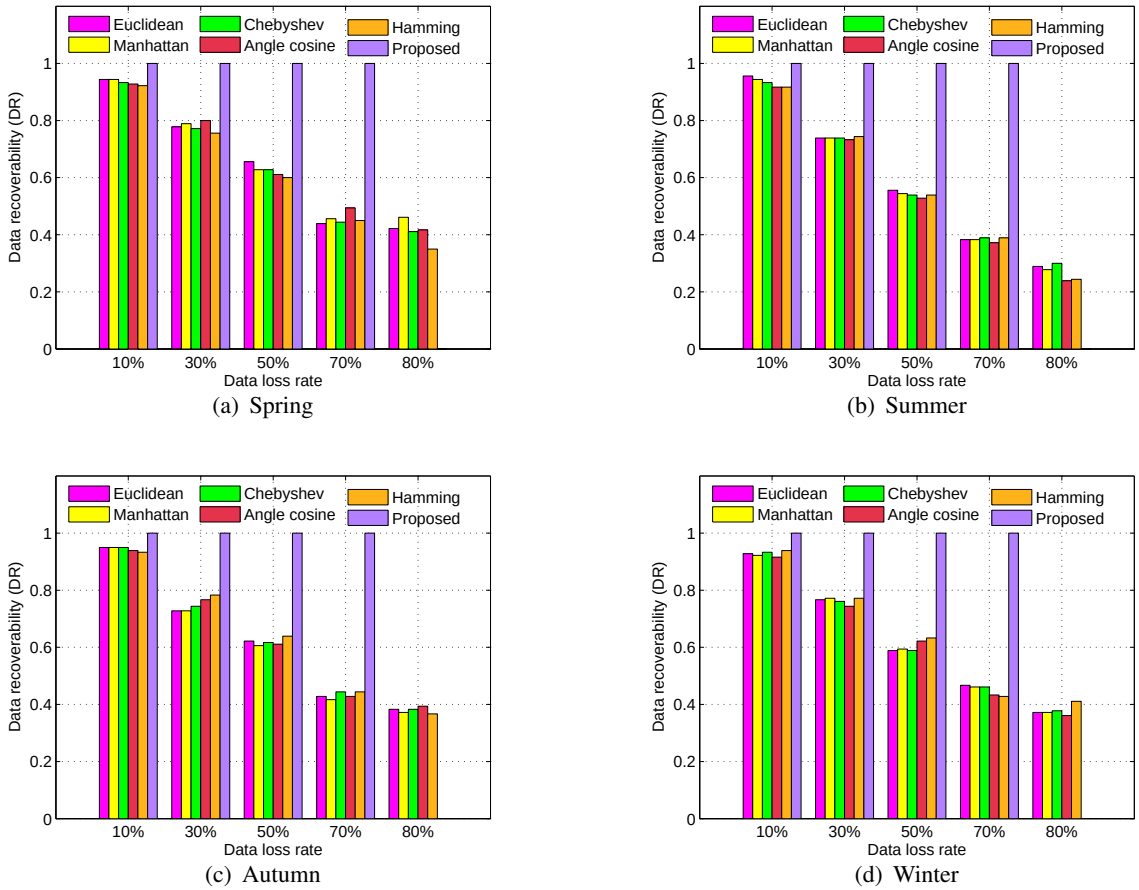

Figure 7. Correlation between data recoverability (DR) and different data lost rate (%) by comparing kNN with five distance measures (Euclidean, Manhattan, Chebyshev, Angle cosine and Hamming) and proposed scheme. The recovery error is $\alpha = 1.0$. For kNN method, the parameter k is set to $k = 3$.

problem. As long as the jumping rate is less than theoretical redundancy rate R_e, the recovery results calculating from all received data shares contain at least two identical results, which is likely to be the original data.

Notably, we need to stress that the jumping rate must be less than, not equal to, theoretical redundancy rate R_e. This is because if the jumping rate is equal to the redundancy rate R_e, the recovery results calculating from all received data shares may be different from each other. This makes that the source node can not decide which result is the original data, leading to a wrong reconstruction result.

4.3. Comparison with the state of the arts

In this subsection, we compare the proposed ensemble scheme with several existing interpolation methods, 'Zero interpolation' [10], 'Slinear interpolation' [13], and 'Quadratic interpolation' [14]. Proposed scheme first uses data decomposition to expand the original data to multiple data shares and then deliver them by wireless sensor networks. Even though the source node do not receive all data

Table 4. Data recoverability for single recovery and ensemble recovery. The data loss rate is set to 10% and the jumping rates are respectively 10%, 20%, 30%, 40%, and 50%.

Scheme	Round R	Data jumping rate				
		10%	20%	30%	40%	50%
Single Recovery	$R = 1$	46%	16%	5%	2%	0%
Ensemble Recovery	$R = 5$	100%	99%	95%	78%	0%
	$R = 50$	100%	100%	100%	87%	51%
	$R = 100$	100%	100%	100%	98%	99%
	$R = 1000$	100%	100%	100%	100%	99%
	$R = 5000$	100%	100%	100%	100%	100%

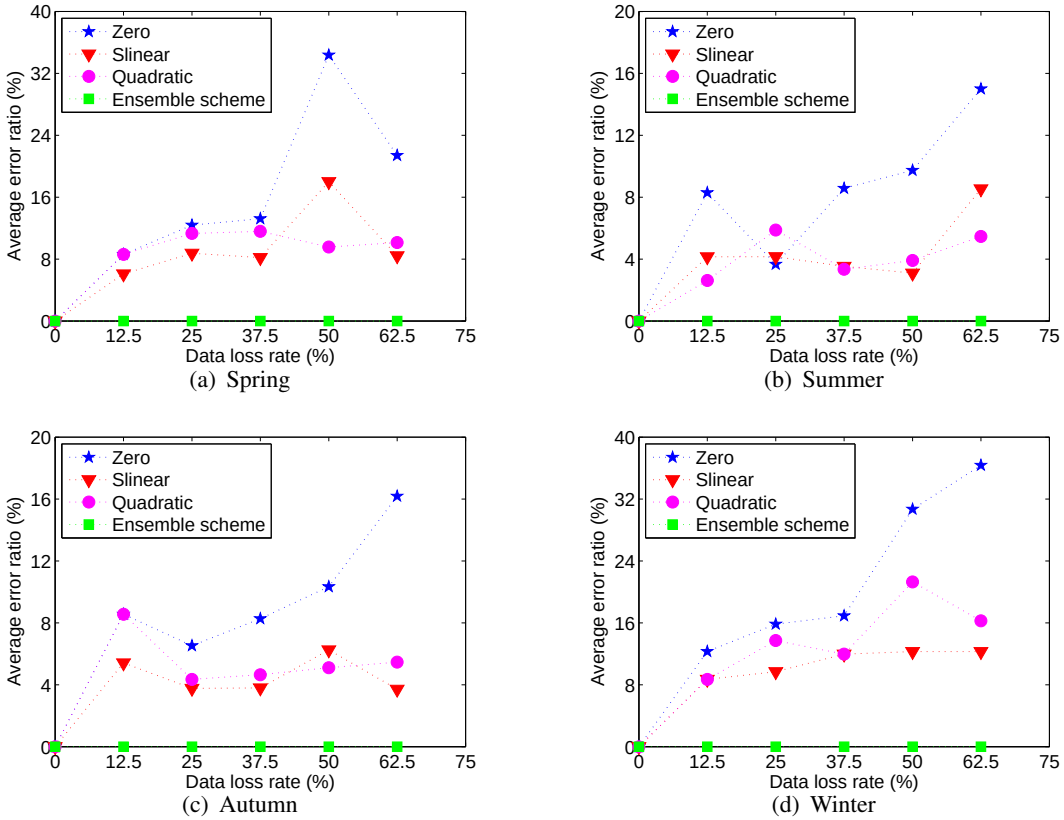

Figure 8. The variation tendency for average reconstruction error rate when only data loss pattern is considered. In this test, the ensemble recovery mechanism is used and the ensemble rounds is set to $R = 101$.

shares, it can still reconstruct original data by using ensemble recovery mechanism. Note that the direct comparison is also difficult, because existing typical interpolation methods usually estimate the original data so that the exact values are hard to be obtained, whereas our proposed scheme definitely gets the exact original data as long as data loss rate is lower than theoretical redundancy rate R_e. According

to above consideration, we need to build a fair comparison between the proposed scheme and several interpolation methods as follows.

All simulation experiments are implemented in the same wireless sensor network, which contains 30 sensor nodes and one source node. For every data reconstruction scheme, we divide the same original data (also named as original valid data) and set different data loss rates. The data recoverability performance can be measured by counting how much original valid data the recipient can finally receive. The experiments are repeated 100 times. In addition,

- For the proposed scheme, we decompose the original data to multiple data shares. All shares are integrated and sent by the sensor nodes randomly. In this experiment, the ensemble rounds are set to $R = 101$ and the decomposition parameters are fixed as $m = 7$, $n = 30$, and $q = 37$, respectively. Thus, the original data will be expanded to $\frac{n}{m} = 4.3$ times. The Equation (4.1) is used to calculate the overall recovery accuracy rate, which represents the data recoverability of proposed scheme.
- For the other interpolation schemes, original valid data are directly transmitted over sensor network. Since the estimation values always have some errors comparison with the actual values, we therefore set the error ratio (ER) to show the difference between actual value $x(i)$ and estimation value $\hat{x}(i)$, which is calculated by Equation (4.2).

$$ER = \frac{\sqrt{\sum_{i,j} (x(i,j) - \hat{x}(i,j))^2}}{\sqrt{\sum_{i,j} (x(i,j))^2}} \tag{4.2}$$

We carry out a serial of experiments to compare the proposed scheme and existing works. The average recovery error ratio is used to measure the performance of different methods. It indicates the proportion of error times for the total number of experimental times. Figure 8 shows the results of average error ratio for several schemes. Since proposed scheme either obtains the exact original data, or gets the complete wrong original data, in our experiments the average error ratio will be set directly to 100% once the data damaged rate (including data loss and data jumping) is more than theoretical redundancy rate R_e.

We can observe that proposed scheme achieves low average recovery error ratio with high data loss rate, e.g. Figure 8. In fact, this conclusion has been explained in detail by Section 3.4. Also, we note that proposed ensemble scheme has significantly effective for the case of combination data loss and data jumping, e.g. Figure 9. When we fix data loss rate to 12.5%, average recovery error ratio for ensemble recovery is even approximate zero when data jumping rate is up to 20%, the effect, however, does not work well for single recovery. This demonstrates that proposed ensemble scheme can efficiently improve the accuracy of data recovery. We also explain this interesting phenomenon as follows. The interpolation methods are usually very hard to provide the exact value for each original data. This makes that the recovery data may be decided as wrong value once the recovery error α is out of the range, and finally leads to a high average recovery error ratio. Moreover, for existing interpolation methods, we can also observe that zero interpolation method gives an inferior performance than other several schemes. This is mainly because zero interpolation always employs the nearest neighbors to estimate the original data. If data is continuously lost, The estimation value that provided by zero interpolation method may deviate significantly from the exact value, leading to an inferior performance.

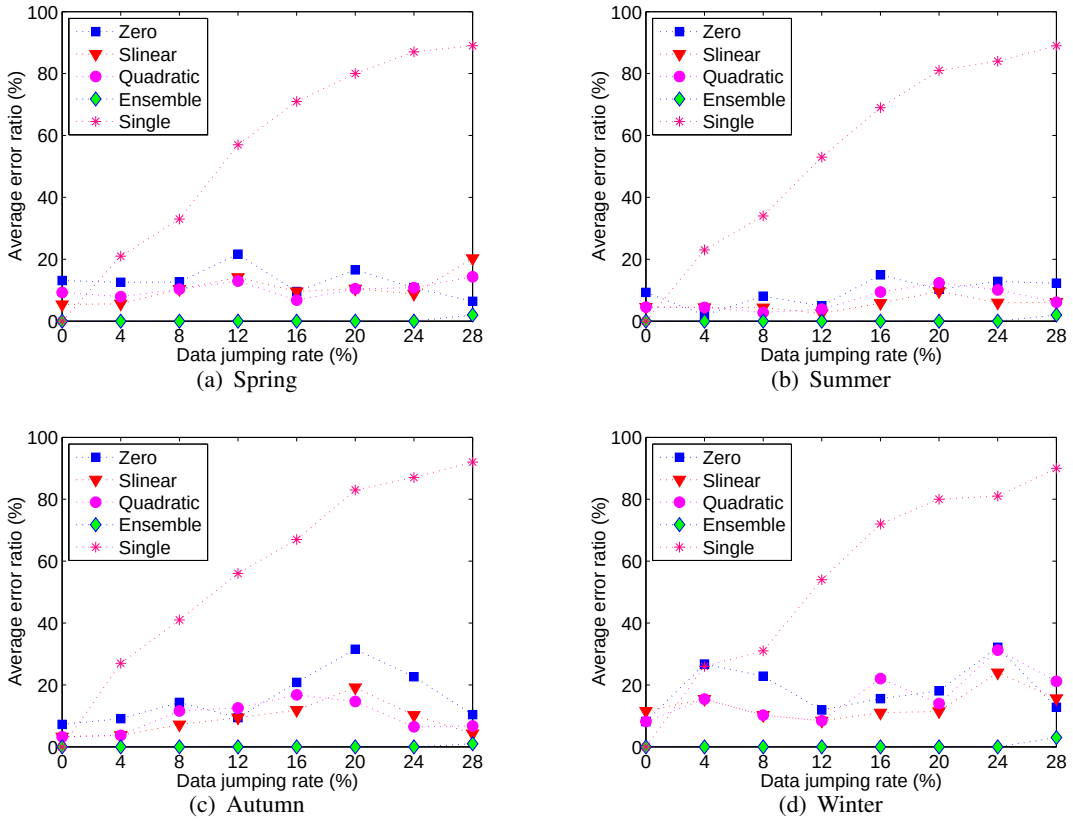

Figure 9. The variation tendency for average reconstruction error rate when data loss pattern and data jumping pattern are both considered. In this test, data loss rate is fixed to 10%. The ensemble recovery mechanism is used and the ensemble rounds is set to $R = 101$.

5. Conclusions

In this paper, we addressed the reliable data transmission problem in WSNs and proposed a new scheme based on data decomposition and ensemble recovery mechanism, which is significantly different from the traditional interpolation schemes. We design matrix decomposition mechanism to split the original data into multiple data shares, and then transmit them through wireless networks, which including multiple sensor nodes. In order to reconstruct the original data completely, ensemble recovery mechanism is designed to ensure the correct recovery for missing data. This mechanism can not only recover the missing data, but can revise the incorrect data if they occurs jumping in transmission. We compared our scheme with several existing interpolation schemes. The results show proposed scheme has better performance and herein shows a valuable attempt for data recovery in wireless sensor networks.

In addition, we should note that the proposed scheme can circumvent the problem of data loss in network transmission. However, its disadvantage are also obvious due to the following two aspects: (1) Since the original data are expanded and divided into multiple shares by data decomposition mechanism, the sensor nodes need to spend more time sending these data. It may cause energy waste. As we known, the energy is very important for individual sensor node. (2) Unlike the interpolation scheme,

proposed scheme requires the individual sensor node to pre-process the original data, that is, calculate the expanded data by decomposition matrix. This increases the computational requirements for individual sensor node. Overall, proposed method may be more suitable for the high-precision data recovery.

In the future, we plan to carry our work forward in two directions. First, we should further optimize the data decomposition mechanism and find a fast computation method. Second, we should try to combine the estimation method and data decomposition mechanism. This will be considered as part of the future effort.

Acknowledgements

This work was supported by Natural Science Foundation of China under Grants (61602295, U1736120, 61672337) and Natural Science Foundation of Shanghai (16ZR1413100).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. H. Shen, X. Li, Q. Cheng, et al., Missing information reconstruction of remote sensing data: A technical review, *IEEE Geosc. Rem. Sen. M.*, **3** (2015), 61–85.
2. M. Chen, S. Mao and Y. Liu, Big data: A survey, *Mobile Netw. Appl.*, **19** (2014), 171–209.
3. A. Sandryhaila and J. Moura, Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure, *IEEE Signal Proc. Mag.*, **31** (2014), 80–90.
4. T. Cover and P. Hart, Nearest neighbor pattern classification, *IEEE T. Inform. Theory*, **13** (1967), 21–27.
5. L. Kong, D. Jiang and M. Wu, Optimizing the spatio-temporal distribution of cyber-physical systems for environment abstraction, 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS), Genoa, Italy, June 21–25, (2010), 179–188.
6. H. Zhu, Y. Zhu, M. Li, et al., SEER: Metropolitan-scale traffic perception based on lossy sensory data, in Proceedings of IEEE International Conference on Computer Communications (INFOCOM), Rio de Janeiro, Brazil, April 19-25, (2009), 217–225.
7. E. Candes, J. Romberg and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, *IEEE T. Inform. Theory*, **52** (2006), 489–509.
8. D. Donoho, Compressed sensing, *IEEE T. Inform. Theory*, **52** (2006), 1289–1306.
9. L. Kong, M. Xia, X. Liu, et al., Data loss and reconstruction in wireless sensor networks, *IEEE T. Parall. Distr.*, **25** (2014), 2818–2828.
10. Z. Chen, L. Chen, G. Hu, et al., Data reconstruction in wireless sensor networks from incomplete and erroneous observations, *IEEE Access*, **6** (2018), 45493–45503.

11. F. Li, K. Wu, X. Zhang, et al., Robust batch steganography in social networks with non-uniform payload and data decomposition, *IEEE Access*, **6** (2018), 29912–29925.
12. X. Zhang, Matrix analysis and applications, Beijing, Tsinghua University Press, (2004), 161–166.
13. A. Sekey, A computer simulation study of real-zero interpolation, *IEEE T. Audio Electroacoustics*, **18** (1970), 43–54.
14. N. Dodgson, Quadratic interpolation for image resampling, *IEEE T. Image Process.*, **6** (1997), 1322–1326.
15. M. Wen, K. Lu, J. Lei, et al., BDO-SD: An efficient scheme for big data outsourcing with secure deduplication, 2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Hong Kong, China, April 26-May 1, (2015), 214–219.

AIMS Press