2019

V-Matrix-Based Scalable Data Aggregation Scheme in WSN

Xindi Wang
Dongguan University of Technology

Qingfeng Zhou
Dongguan University of Technology

Jun Tong
University of Wollongong, jtong@uow.edu.au

V-Matrix-Based Scalable Data Aggregation Scheme in WSN

Abstract

Data aggregation is one of the most important functions provided by wireless sensor networks (WSNs). Among a variety of data aggregation schemes, the coding-based approaches (such as Compressive sensing (CS) and other similar programs) can significantly reduce traffic quantity by encoding the raw sensed data using weight vectors. The critical feature to design a coding-based data aggregation protocol is to construct a weight/measurement matrix for the application scenario. After that, the sink node assigns the column of the matrix, which is treated as the weight vector during the encoding process, to each sensor node respectively. However, for a dynamic scenario where the number of sensor nodes changes frequently, the existing approaches have to reconfigure the network by regenerating the measurement matrix and allocating the new weight vectors for all the existing nodes, which causes a considerable energy consumption and affects the regular monitoring tasks. To solve this problem, we propose a Vandermonde matrix-based scalable data aggregation protocol (VSDA), which preserves the advantages of coding-based schemes and addresses the issues mentioned above. In VSDA, as new nodes join into the scaled-up network, the original weight vectors owned by the original nodes do not need to regenerate the weight vectors entirely but add some new entries by itself at all. It outperforms the existing schemes by saving the energy in network scaling-up. Besides, we propose a concise hardware framework to quantify the data encoding process of VSDA, which provides a performance analysis process that is closer to practical application. The numeric tests validate the performance of VSDA compared with the existing schemes in several aspects, such as, the number of transmissions, energy consumption, and storage space showing the outperformance of VSDA scheme.

Disciplines

Engineering | Science and Technology Studies

Publication Details

X. Wang, Q. Zhou & J. Tong, "V-Matrix-Based Scalable Data Aggregation Scheme in WSN," IEEE Access, vol. 7, pp. 56081-56094, 2019.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 10, 2019, accepted April 23, 2019, date of publication May 1, 2019, date of current version May 9, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2913396

V-Matrix-Based Scalable Data Aggregation Scheme in WSN

XINDI WANG[1], QINGFENG ZHOU[1], AND JUN TONG[2]
[1]School of Electric Engineering and Intelligentization, Dongguan University of Technology, Dongguan 523808, China
[2]School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

Corresponding author: Qingfeng Zhou (enqfzhou@ieee.org)

This work was supported in part by the Science and Technology Planning Project of Guangdong Province under Grant 2016B010108002, in part by the NSFC under Grant 61471156, and in part by the Starting Fund of Dongguan University of Technology under Grant DGUT-G200906-49.

ABSTRACT Data aggregation is one of the most important functions provided by wireless sensor networks (WSNs). Among a variety of data aggregation schemes, the coding-based approaches (such as *Compressive sensing* (CS) and other similar programs) can significantly reduce traffic quantity by encoding the raw sensed data using weight vectors. The critical feature to design a coding-based data aggregation protocol is to construct a weight/measurement matrix for the application scenario. After that, the sink node assigns the column of the matrix, which is treated as the weight vector during the encoding process, to each sensor node respectively. However, for a dynamic scenario where the number of sensor nodes changes frequently, the existing approaches have to reconfigure the network by regenerating the measurement matrix and allocating the new weight vectors for all the existing nodes, which causes a considerable energy consumption and affects the regular monitoring tasks. To solve this problem, we propose a *Vandermonde* matrix-based scalable data aggregation protocol *(VSDA)*, which preserves the advantages of coding-based schemes and addresses the issues mentioned above. In *VSDA*, as new nodes join into the scaled-up network, the original weight vectors owned by the original nodes do not need to regenerate the weight vectors entirely but add some new entries by itself at all. It outperforms the existing schemes by saving the energy in network scaling-up. Besides, we propose a concise hardware framework to quantify the data encoding process of *VSDA*, which provides a performance analysis process that is closer to practical application. The numeric tests validate the performance of *VSDA* compared with the existing schemes in several aspects, such as, the number of transmissions, energy consumption, and storage space showing the outperformance of *VSDA* scheme.

INDEX TERMS Wireless sensor network, data aggregation, vandermonde matrix, measurement matrix.

I. INTRODUCTION

Data aggregation is one of the most important functions provided by the wireless sensor network (WSN) [1], [2], which gathers the sensor readings from sensor nodes to data collection sites (sink nodes) by multi-hop routing. Since sensor nodes usually have limited computing capability and power reserve, the primary goal of data aggregation processes is to collect data at required accuracy with the lower power consumption.

Raw data aggregation (*RDA*) is a conventional methodology applied in WSN. In *RDA*, each node needs to transmit its raw data and also relays the received data to sink over multiple-hop without any data processing. As wireless transmission is the major contributor to power consumption in WSN, reducing the redundant transmission volume during data aggregation is a vital problem. Thus, data splicing (*DS*), which splice the payload of several packets together, is often used in some practical applications to reduce the amount of transmitted data. The drawback of *DS* is that the protocol design is relatively complicated and there are many spaces occupied by control overheads. These issues of *DS* are also occurred in the traditional data compression schemes [3], [4] and distribute source coding techniques [5], [6], which make them inefficient for practical applications.

In recent years, *Compressive Sensing* (CS) [7], [8] provides a new approach for data aggregation in WSNs. In the existing CS-based data gathering schemes [10]–[18], the sensed readings of all N sensor nodes in the network is modeled as a data vector $\mathbf{x} = (x_1, x_2, \cdots, x_N)^T$ in which x_j

The associate editor coordinating the review of this manuscript and approving it for publication was Emanuele Lattanzi.

is denoted as the acquired datum of node j. The data gathering process is represented as

$$\boldsymbol{y} = \sum_{j=1}^{N} x_j \boldsymbol{\phi}_j = \boldsymbol{\Phi x}, \tag{1}$$

where each raw datum x_j is encoded by a unique weight vector $\boldsymbol{\phi}_j$. Therefore, it can also be considered that the measurement matrix $\boldsymbol{\Phi}$ is composed of the weight vectors corresponding to each node. It has been proved that if the data vector $\boldsymbol{x}$ meets the k-sparse property and the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ meets the *RIP* (Restricted Isometry Property [9]) condition, then the sink node can recover the original data vector $\boldsymbol{x}$ from $\boldsymbol{y} = (y_1, \cdots, y_M)^T$ with a very high accuracy. Based on *RIP*, in a general CS-based data aggregation scheme, the length of weight vector, namely the row dimension of measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$, is related to the total number of nodes in the network (N). So, as the total amount of nodes in the network increases, the length of the weight vector also is increased synchronously.

The existing CS-based studies usually focus on constructing the measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ based on the existing network scenario and meeting *RIP* condition simultaneously. In these studies, it has been shown that the CS-based scheme can achieve better performance under the cluster topology [10]–[12], [17], chain topology [14], and random topology [18], [20]. However, there still exists some common drawbacks in the CS-based data aggregation schemes. As inherited from CS algorithm, their reconstructed data naturally have decoding error due to the underdetermined feature of the measurement matrix. This issue poses a significant limitation to practical applications. Next, since CS only focuses on the overall information of the network, it naturally ignores the correlation between local links. Therefore, there is greater redundancy in CS-based data aggregation scheme design under some specific topologies. For example, the data transmission in each branch of a tree-like topology is usually independent of each other, so the data aggregation scheme in each path can be designed independently, which is also the main research scenario in this paper. Although there have been many studies using the CS-based data aggregation scheme on the tree-like topology scenario [17], [19], it seems that these schemes are only applicable to some straightforward tree structures or requiring an extra schedule design to assist, which lack of universality in a general application scenario. Obviously, the CS-based scheme may not obtain a more efficient performance under this topology, which has been shown in [21].

To improve the performance of data aggregation process in a tree topology, a topology-aware data aggregation scheme *TADA* [21] is proposed recently. *TADA* is an efficient data aggregation scheme which encodes the raw readings with weight vectors as in CS while its measurement matrix in *TADA* consists of several orthogonal vector sets based on the topology information in the network. Meanwhile, the weight vector adopted in *TADA* owns a smaller size, which results in fewer transmissions than the CS-based schemes. Also, the orthogonality between weight vectors on a shared path guarantees a high reconstruction accuracy of raw data. Therefore, *TADA* scheme takes full advantage of the characteristics of the tree-like topology and gets better performance than CS-based scheme.

In general, both *TADA* and CS-based schemes are designed based on a static scene in which the number of nodes or the network topology is fixed. However, the fact is that the WSN is a dynamic network and it is common to add some new nodes into the existing network to expand the monitoring area. Therefore, here we consider a more realistic application scenario where some new nodes prepare to add in the dynamic network one after another. In this case, since the dimensions of the existing measurement matrix in CS are designed based on the total number of nodes in the existing scene, they have to regenerate a new measurement matrix with a larger size and meet the CS properties at the same time. After that, sink has to re-allocate the vectors from the newly generated matrix to all the nodes in the current network. During this period, the regular monitoring task has to be interrupted and additional energy will be consumed in the weight vector re-allocation process. Thus, it can be seen that the dynamic change of the number of nodes has a significant influence on the CS-based scheme. Although the scheme *TADA* can address some cases happened in a scaled-up scenario, it tends to suffer the same problem in the case where the newly added node joins in the longest path of the network. In summary, the existing data aggregation schemes tend to be inefficient in a scaled-up practical scenario.

A. CONTRIBUTIONS OF THIS ARTICLE

Based on the above analysis, to solve the problem caused by network scaling up, we propose a *Vandermonde* matrix-based scalable data aggregation scheme (*VSDA*). The main contributions of this paper are described as follows.

- We generate an easy-to-expand structure of the measurement matrix, which can easily expand to a larger size based on the previous matrix.
- We analyze different scaled-up scenarios in detail and present the corresponding expansion strategy to each case.
- We propose a concise hardware framework to quantify the actual data aggregation process and validate the performance of proposed scheme under this implementation framework.

The rest of the article is organized as follows. Section II presents the data aggregation process of the *VSDA* scheme and puts forward the problem formulation of a scaled-up scenario. Section III propose a quantification framework to exhibit the actual encoding process of *VSDA* scheme. In Section IV, the proposed *VSDA* scheme is evaluated against the traditional CS-based schemes and *TADA* in several aspects, such as the amount of transmissions, energy consumption, and storage space. Section V concludes the paper and suggests some future works.

TABLE 1. Notations table.

Notations	Meaning
N	The total number of nodes in the network.
M	The dimension of the weight vector.
L_i	The length of path i.
L_{max}	The length of the longest path in the network.
x_i	The raw datum acquired by node i.
$\boldsymbol{x}_i$	The raw datum of nodes on path i, whose dimension is L_i .
$\boldsymbol{y}_i$	The aggregated data from path i.
$\boldsymbol{\Phi}$	The measurement matrix with $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ for the whole network.
$\boldsymbol{\Phi}^{M\times L_i}$	The measurement matrix with $\boldsymbol{\Phi}^{M\times L_i} \in \mathbb{R}^{M\times L_i}$ for path i.
$\boldsymbol{\phi}_i$	The weight vector with $\boldsymbol{\phi}_i \in \mathbb{R}^M$ for node i.
$\mathcal{D}$	The coding set with $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}$.
s	The base element of coding set $\mathcal{D}$.
x_{i_2}	The quantification of raw data x_i with B_x bits.
$\boldsymbol{\phi}_{i_2}$	The quantification of weight vector $\boldsymbol{\phi}_i$ with B_ϕ bits.
B_x	The resolution of the raw data x.
N_e	The resolution of each entry in the encoded data.
B_ϕ	The resolution of each encoded entry of weight vector.

In addition, we summarize the notations used in this paper in Table. 2.

TABLE 2. The feasible values of *s*.

Decimal	Binary
1.00048828125	B 1.00000000001
1.0009765625	B 1.00000000010
1.00146484375	B 1.00000000011
...	...
1.1328125	B 1.00100010000

II. DESIGN OF VSDA SCHEME

This section introduces the data aggregation process of the *VSDA* scheme and puts forward the problem formulation of the scaled-up scenario. Then we divide the scaled-up scenarios into two cases and propose corresponding update strategy to each case.

A. BASIC DATA AGGREGATION PROCESS

Consider a WSN with N sensor nodes and a sink node. The raw data of all sensor nodes can be formulated as a vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_N)^T$, in which x_i is the acquired datum by the i-th node. As mentioned in Section. I, the coding-based data aggregation process can be represented as Eq. (1) in which the design of measurement matrix $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ is the key point. Among existing schemes, the CS-based scheme integrally generates the matrix $\boldsymbol{\Phi}$ following some distribution, such as *Gaussian* and *Bernoulli* distribution. Also, the column dimension N is usually equal to the total number of nodes in the network. In this way, each node will get a weight vector from $\boldsymbol{\Phi}$ without repeating. In *VSDA*, we construct $\boldsymbol{\Phi}$ based on network topology in *VSDA* as *TADA* does. To do this, at the initialization phase of *VSDA*, the sink node generates the network topology based on *BMST* algorithm [21] and collects following information: the amount of nodes and the corresponding nodes'ID on each path. Next, the sink node generates a *coding set* $\mathcal{D} = \{\boldsymbol{\phi}_1\ \boldsymbol{\phi}_2 \ldots \boldsymbol{\phi}_{L_{max}}\} \in \mathbb{R}^{M\times L_{max}}$ $(M \geq L_{max})$, in which L_{max} is the length of the longest path. After that, for any path i, the sink node allocates the L_i (the length of path i) column vectors of $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}$ to nodes on the path i. In this way, the measurement matrix in *VSDA* or *TADA* consists of weight vectors assigned to each node, and it is stored in the sink node. After the vector allocation process, each sensor node encodes its raw datum using the assigned weight vector and the aggregated data of path i is produced by accumulating all encoded data on this path. The details of above processes are elaborated in [21].

B. PROBLEM FORMULATION OF A SCALED-UP SCENARIO

Based on above basic scenario, we study the data aggregation process under the scaled-up scenario here.

Assume that the raw data acquired by nodes on path i are denoted as a data vector $\boldsymbol{x}_i \in \mathbb{R}^{L_i}$ and the corresponding data aggregation process is given by

$$\boldsymbol{y}_i = \boldsymbol{\Phi}^{M\times L_i}\boldsymbol{x}_i, \tag{2}$$

where L_i is the number of nodes on path i and the matrix $\boldsymbol{\Phi}^{M\times L_i} \in \mathbb{R}^{M\times L_i}$ consists of L_i weight vectors which belong to nodes on the path i.

Assume there is a new node ready to join the path i, then the data aggregation process of this expanded path is given by

$$\boldsymbol{y}'_i = \boldsymbol{\Phi}'\boldsymbol{x}'_i, \tag{3}$$

where $\boldsymbol{x}'_i = [\boldsymbol{x}_i\ x'_1]^T$, x'_1 is the raw datum sampled by the new node. $\boldsymbol{\Phi}'$ is the measurement matrix applied in the scaled-up scenario and it has to meet a larger column dimension (L_i+1) as the total number of nodes increases.

Obviously, if we can simply pass a new weight vector $\boldsymbol{\phi}_1{}'$ to the newly added node and the existing configuration keeps unchanged, then the new matrix $\boldsymbol{\Phi}'$ can be directly constructed by combining the existing matrix $\boldsymbol{\Phi}^{M\times L_i}$ and the vector $\boldsymbol{\phi}_1{}'$ together, the process Eq. (3) can be further represented by

$$\boldsymbol{y}'_i = \left[\boldsymbol{\Phi}^{M\times L_i}\ \boldsymbol{\phi}_1{}'\right]\begin{bmatrix}\boldsymbol{x}_i\\ x'\end{bmatrix}, \tag{4}$$

which is an ideal situation that there is no need to change the configuration of previous network.

After performing the data gathering process in this path, the sink node receives the aggregated data $\boldsymbol{y}'_i$ and reconstructs the whole data $\boldsymbol{x}'_i$ by solving above equation. In this case,

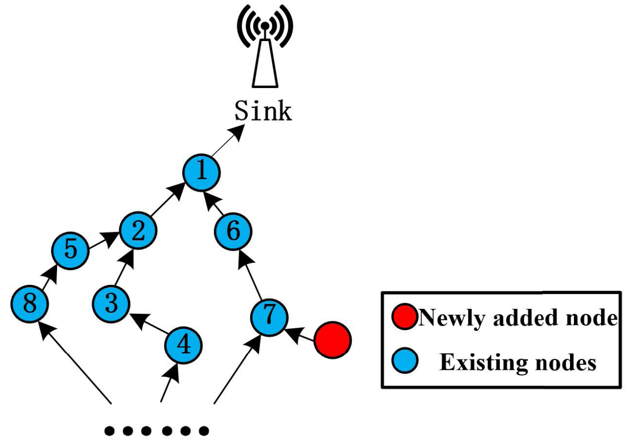

FIGURE 1. Illustration of the network scaling example.

if the columns of $\boldsymbol{\Phi}^{M\times(L_i+1)}$ are independent with each other, the raw data vector $[\boldsymbol{x}_i'\ x_1']^T$ can be recovered exactly.

Therefore, if we use the same construction method as Eq. (4) to generate the matrix applied in the scaled-up scenario, as the number of newly added nodes increases, the data aggregation process on this path can be further given by

$$\boldsymbol{y}_i' = [\boldsymbol{\Phi}^{M\times L_i}\,\boldsymbol{\phi}_1{}'\cdots\boldsymbol{\phi}_n{}'] \begin{bmatrix} \boldsymbol{x}_i \\ x_1' \\ \vdots \\ x_n' \end{bmatrix}, \tag{5}$$

where n is the total number of newly added nodes.

In this case, generating the new measurement matrix $\boldsymbol{\Phi}^{M\times(L_i+n)} = [\boldsymbol{\Phi}^{M\times L_i}\,\boldsymbol{\phi}_1{}'\cdots\boldsymbol{\phi}_n{}']$ as Eq. (5) has to face a serious issue that the total number of nodes may be larger than the length of weight vector, namely $L_i + n > M$, thus the columns of $\boldsymbol{\Phi}^{M\times(L_i+n)}$ cannot meet the independence property. As a result, it is impossible to obtain the raw data by solving Eq. (5) directly. At this point, the easiest way to deal with this situation is to generate a matrix $\boldsymbol{\Phi}^{M'\times(L_i+n)}$ of a larger row dimension which satisfies the condition $M' \geq L_i + n$ and the columns are independent with each other.

For the existing studies such as *TADA* and CS-based schemes, they have to entirely re-generate a new matrix $\boldsymbol{\Phi}^{M'\times(L_i+n)}$ which certainly affects the original configurations (the assigned weight vectors) of all the original nodes. It requires additional energy cost to reconfigure the whole network and also interrupts the monitoring task.

To address this issue, we plan to generate a scalable structure of matrix $\boldsymbol{\Phi}^{M\times L_i}$ on path i, which can be expanded based on existing elements without re-generating a new one and guarantees the independence property between each column. Since the matrix $\boldsymbol{\Phi}^{M\times L_i}$ consists of the assigned weight vectors, which are picked from *coding set*. Therefore, to design a *coding set* $\mathcal{D}$ with scalable structure is the key issue.

To clearly explain our idea, a graphical description of the network scaling is given as follows.

In Fig. 1, suppose currently there are 14 existing nodes (the blue points) in the scenario. In this case, the CS-based studies only focus on generating a feasible measurement matrix $\boldsymbol{\Phi}$, such as $\boldsymbol{\Phi} \in \mathbb{R}^{5\times 14}$ for the existing scenario. Then, the sink node assigns and delivers the column vector $\boldsymbol{\phi}_i, i \in \{1, 2, \ldots, 14\}$ of $\boldsymbol{\Phi}$ to these 14 nodes to encode their raw data. The corresponding data aggregation process is given in Eq. (1).

At a certain moment, if a new node (the red point) joins the network, in which all column vectors of the matrix $\boldsymbol{\Phi} \in \mathbb{R}^{5\times 14}$ have been exhausted. If a CS-based approach is considered, one may design a new measurement matrix $\boldsymbol{\Phi}' \in \mathbb{R}^{6\times 15}$ by using the traditional CS-based approaches. This, however, can lead to high energy consumption because 14 new weight vectors from the updated matrix $\boldsymbol{\Phi}'$ should be transmitted to the prior 14 sensor nodes.

C. DESIGN CODING SET

As described above, if the previous weight vectors of existing sensor nodes can be reused in the scaled-up scenario, we just need to update the previous matrix $\boldsymbol{\Phi}^{M\times L_{max}}$ by adding some new entries, which avoids re-configuring the whole network. Thus, we plan to construct the new matrix $\boldsymbol{\Phi}^{(M+m)\times(L_i+n)}$ as Fig. 2 shows. Meanwhile, the columns of the new matrix have to meet the condition that they are independent of each other.

To accomplish this target, *VSDA* designs the coding set $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}(M \geq L_{max})$ with a scalable structure in this subsection.

Firstly, we generate a *Vandermonde matrix* $\boldsymbol{G} \in \mathbb{R}^{M\times n}(M \leq n)$ with structure as follows.

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^n \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^n)^2 \\ & & \vdots & & \\ s^{M-1} & (s^2)^{M-1} & (s^3)^{M-1} & \cdots & (s^n)^{M-1} \end{bmatrix}. \tag{6}$$

As proven in [22], any M columns of $\boldsymbol{G}$ are linearly independent when $s, s^2, \ldots, s^n$ are all distinct. Apparently, it is straightforward to establish that any L_{max} ($L_{max} \leq M$) columns of $\boldsymbol{G}$ are linearly independent too. Thus we pick out the first L_{max} columns of $\boldsymbol{G}$ to compose the coding set $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}$, in which L_{max} denotes the maximum path length. The coding set is given by

$$\mathcal{D} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ s & s^2 & s^3 & \cdots & s^{L_{max}} \\ s^2 & (s^2)^2 & (s^3)^2 & \cdots & (s^{L_{max}})^2 \\ & & \vdots & & \\ s^{M-1} & (s^2)^{M-1} & (s^3)^{M-1} & \cdots & (s^{L_{max}})^{M-1} \end{bmatrix}. \tag{7}$$

D. UPDATE STRATEGY DESIGNED IN SCALED-UP SCENARIO

In this subsection, we discuss the impact caused by the new node joins in the network and the corresponding upgrade strategy in the scaled-up scenario. Assume the new node joins the i-th path which already has L_i nodes in the existing scenario. We further divide the scaled-up scenario into two

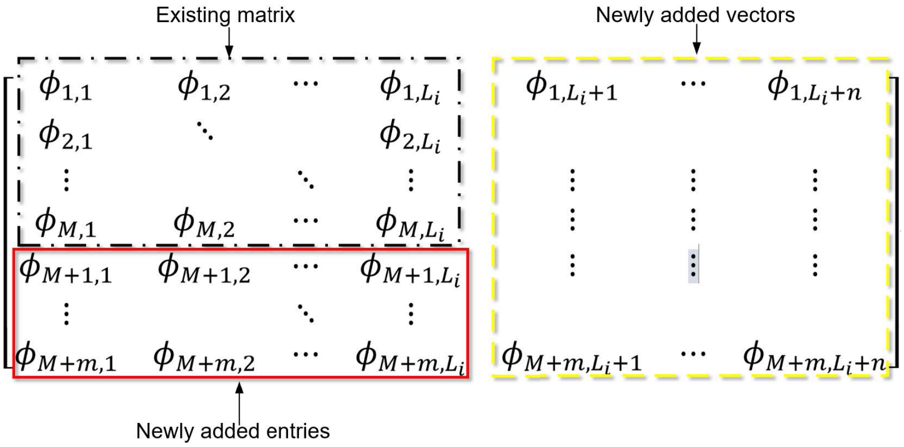
Existing matrix
Newly added vectors
$\phi_{1,1}$ $\phi_{1,2}$ $\cdots$ ϕ_{1,L_i}
$\phi_{2,1}$ $\ddots$ ϕ_{2,L_i}
$\vdots$ $\ddots$ $\vdots$
$\phi_{M,1}$ $\phi_{M,2}$ $\cdots$ ϕ_{M,L_i}
$\phi_{M+1,1}$ $\phi_{M+1,2}$ $\cdots$ ϕ_{M+1,L_i}
$\vdots$ $\ddots$ $\vdots$
$\phi_{M+m,1}$ $\phi_{M+m,2}$ $\cdots$ ϕ_{M+m,L_i}
ϕ_{1,L_i+1} $\cdots$ ϕ_{1,L_i+n}
ϕ_{M+m,L_i+1} $\cdots$ ϕ_{M+m,L_i+n}
Newly added entries

FIGURE 2. Illustration of constructing a new matrix which consists of existing matrix, newly added entries, and newly added vectors.

cases and analyze the corresponding weight vector allocation strategy respectively.

1) CASE 1

The newly added sensor node connects to the node on the non-longest path of existing topology or directly link to the sink node. It results in generating a new path to the sink node or expanding the existing non-longest path. In either case, we can find a valid weight vector in $\mathcal{D}^{M\times L_{max}}$ and assign it to the newly joined node. For example, if the newly added node joins in the i-th path with length L_i, the sink node only needs to assigns the $(L_i + 1)$-th column of $\mathcal{D}$ as the weight vector to the new node. If the new node connects to the sink node directly, we treat the initial path length as 0 and the sink node can directly assign the 1-th column of $\mathcal{D}$ to the new node. The energy consumption, in this case, can be measured as a simple data transmission process from the sink node to the new node by multi-hop.

2) CASE 2

The newly added sensor node links to the leaf node on the longest path j with length L_{max} ($L_j = L_{max}$), there is no valid weight vector that can be distributed to the new node since all the columns of $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}(M \geq L_{max})$ are consumed by existing nodes on path j. In this case, we have to update the *coding set* to a larger size, which can offer sufficient column vectors. Here, we further divide the *coding set* update strategy into two subcases:

Subcase 1: If the dimensions of the original *coding set* $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}$ meet $M > L_{max}$, one can always figure out $M - L_{max}$ new vectors which are independent with the existing *coding set* $\mathcal{D}$. Thus we call $M - L_{max}$ as a complimentary space of $\mathcal{D}$. Although producing these new vectors costs extra energy in the sink node, but the positive side is that the previously assigned vectors to the existing nodes will remain unchanged.

Thus, it is able to figure out a new vector $\boldsymbol{\phi}_{L_{max}+1} \in \mathbb{R}^M$ based on $\mathcal{D}$ and assign it to the new node. The updated coding set is given by matrix $[\mathcal{D}\ \boldsymbol{\phi}_{L_{max}+1}]$, which directly combines the newly generated vector $\boldsymbol{\phi}_{L_{max}+1}$ with the existing *coding set* $\mathcal{D}$. Furthermore, since the *coding set* $\mathcal{D}$ is designed based on a special structure of *Vandermonde matrix*, the columns of $\mathcal{D}$ are easily obtained and naturally independent of each other. According to the design principle of *coding set*, the newly added column can be derived as

$$\boldsymbol{\phi}_{L_{max}+1} = (1, s^{L_{max}+1}, \ldots, (s^{L_{max}+1})^{M-1})^T, \tag{8}$$

Here, we call $M - L_{max}$ as the volume of the complimentary space, which determines the scaling-up capacity of the data aggregation scheme. Since the parameter L_{max} remains fixed, if we directly assign a larger value to M, then we will obtain a larger complimentary space to add more new nodes to the longest path without changes the previous configurations. However, since a large value of M may lead to an increase in energy consumption, a trade-off performs efficiently when the value of M is less than a certain threshold. The detail will be described in Section. IV-A.

Subcase 2: If the dimensions of original *coding set* $\mathcal{D} \in \mathbb{R}^{M\times L_{max}}$ meet condition $M = L_{max}$, it is impossible to find a new vector independent with the columns of the existing *coding set* $\mathcal{D} \in \mathbb{R}^{M\times M}$. Thus, if a new node joins into the network, Sink has to expand the *coding set* $\mathcal{D}$ to a larger size.

The case, $L_{max} = M$, seems very extreme in *coding set* design, which leaves no complimentary space to allocate vector to the new node on the longest path. In fact, this situation frequently occurs as the number of nodes joining the network increases. In this case, the entire measurement matrix has to update itself as a new node connects, which is also the main issue affecting the practical application of the CS-based scheme and *TADA* in a scaled-up scenario.

Nevertheless, to tackle this problem, our designed *coding set* will be very efficient. Due to the regular structure of the entries in *Vandermonde* matrix, any sensor node i in the

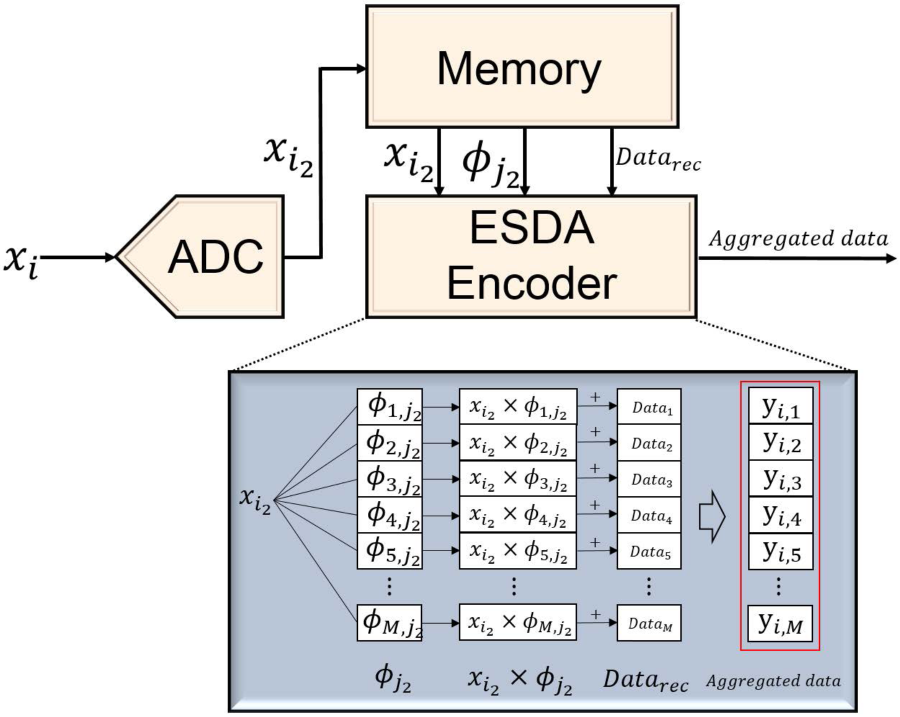

FIGURE 3. Illustration of *VSDA* implementation framework, in which x_{i_2} and ϕ_{i_2} are the quantification of raw data x_i and weight vector ϕ_i owned by node i respectively.

network only needs to update the parameter M and generates its newly added entries using base element s^i. The structure of updated *Coding Set* $\mathcal{D}' \in \mathbb{R}^{M' \times (L_{max}+1)}$ is given by

$$\mathcal{D}' = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ s & s^2 & \cdots & s^{L_{max}+1} \\ s^2 & (s^2)^2 & \cdots & (s^{L_{max}+1})^2 \\ & & \vdots & \\ s^{M'-1} & (s^2)^{M'-1} & \cdots & (s^{L_{max}+1})^{M'-1} \end{bmatrix}, \tag{9}$$

in which $M' \geq L_{max} + 1 > M$. The added entries for any i-th existing column are given by

$$(s^i)^m \ , m \in \{M, M+1, \ldots, M'-1\}. \tag{10}$$

The added column vector is derived by

$$\boldsymbol{d}_{L_{max}+1} = (1, s^{L_{max}+1}, \ldots, (s^{L_{max}+1})^{M'-1})^T, \tag{11}$$

which is the weight vector assigned to the newly added node. This update process is much simpler than re-generating a new and larger matrix completely like CS-based scheme and *TADA* scheme.

III. VSDA IMPLEMENTATION AND KEY PARAMETERS DESIGN

In this section, we propose a concise hardware framework to quantify the vector-encoding process within each sensor node. Combine with the structure of proposed *coding set* and data frame, we further propose the design methods of key parameters s and M in *coding set*, which are closely related to the property of *VSDA* scheme.

A. IMPLEMENTATION FRAMEWORK OF VSDA

To evaluate the performance of *VSDA* scheme in the practical application, we establish a concise *VSDA* implementation framework which consists of three main components: an analog-to-digital converter (ADC), a memory and a *VSDA* encoder. The graphical description of this encoding process is shown in Fig. 3.

1) ANALOG-TO-DIGITAL CONVERTER (ADC)

Without loss of generality, the raw signal x_i acquired by sensor i has been pre-amplified to the same scale voltage before passing to ADC [24]. The B_x bits output of ADC, denoted as x_{i_2}, provides the quantification of the raw datum x_i. $B_x/2$ bits are dedicated to the integer part of x, while the rest bits are for the decimal part of x.

2) MEMORY

Assuming the *coding set* $\mathcal{D} \in \mathbb{R}^{M \times L_{max}}$ ($M \geq L_{max}$) is used here, in which L_{max} represents the length of the longest path. The memory in a sensor node is used to store the assigned weight vector $\boldsymbol{\phi}_i$ and the received data (denoted as $Data_{rec}$) which is received from its child node. The weight vector of node i, $\boldsymbol{\phi}_i = [1 \ s^i \ \ldots \ (s^i)^{M-1}]^T$, is allocated from the i-th ($i \in \{1, 2, \ldots, L_{max}\}$) column of $\mathcal{D}$. Due to the fact that M and i are integers and determined by the sink node, we only need to consider the quantification of s here, which is pre-configured as B_ϕ bits. Here we denote $\boldsymbol{\phi}_{i_2}$

as the quantification of $\boldsymbol{\phi}_i$, and the j-th entry of $\boldsymbol{\phi}_{i_2}$ is denoted as $\boldsymbol{\phi}_{i,j_2}$.

3) VSDA ENCODER

The *VSDA* encoder implements the function that encodes data x_{i_2} through multiplying with each entry of weight vector $\boldsymbol{\phi}_{i_2}$ respectively, and then accumulates the encoded data with the received data $Data_{rec}$. The encoding process essentially amounts to perform M multiplications and M additions within each sensor node.

In addition, we also pre-design a simple payload structure [28], [29] within each data frame, which is divided into several equal-size segments, and each part is used to store an encoded entry, $x_{i_2} \times \phi_{ij_2}$ with the fixed resolution of N_e bits. The graphical description is shown in Fig. 4. The simple structure of data frame can be implemented easily without complex control overhead.

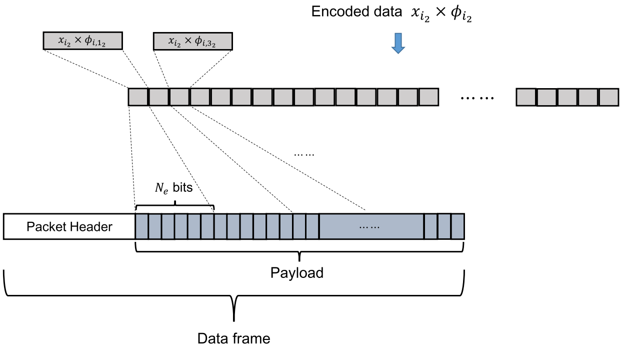
Encoded data $x_{i_2} \times \phi_{i_2}$
$x_{i_2} \times \phi_{i,1_2}$
$x_{i_2} \times \phi_{i,3_2}$
N_e bits
Packet Header
Payload
Data frame

FIGURE 4. An illustration of payload structure and data frame transmitting.

B. DESIGN BASE ELEMENT *s* OF CODING SET

In the memory of implementation framework, the parameter s is directly related to the quantification of each entry in weight vector $\boldsymbol{\phi}_i = [1\ s^i\ \ldots\ (s^i)^{M-1}]^T$. In this subsection, we put forward the feasible range of s to guarantee the performance of *VSDA*.

At first, without loss of generality, we consider s here as a positive number. Because the value of s is directly quantified and stored in the memory, the quantification of s is an important issue we need to care about. Meanwhile, to guarantee the independence between columns of *coding set* which is constructed by a *Vandermonde matrix*, the entries of matrix $1, s, s^2, \ldots, s^M$ should be all distinct. Thus, s is not equal to 0 and 1.

Notice that if s is not less than 2, the largest entry of weight vector, i.e., $s^{L_{max}(M-1)}$, consumes at least $L_{max} \times (M-1)$ bits in payload of data frame. Since we use the same quantization resolution N_e to represent each encoded entry in the payload structure, the case, $s \geq 2$, costs a huge space to store all the encoded entries in the payload of data frame. Meanwhile, it has a serious impact on the amount of transmissions, which will be discussed in the next subsection. So $s < 2$.

In addition, due to the fact that each entry of weight vector will multiply with raw data in *VSDA* encoder, if we set s as a decimal fraction, the higher power of s will be approximately equal to 0. As a result, the corresponding encoded entry is also approximately 0. Thus, we need more bits to accurately represent this encoded entry which causes the same issue as happened in the case $s \geq 2$. So $s > 1$.

In summary, the feasible range of s can be given by

$$s \in \{i | 1 < i < 2, , i \in \mathbb{Q}\}. \tag{12}$$

If we want to exactly represent s without any loss, its quantization should meet the condition by

$$s - \lfloor s \rfloor = \sum_{i=1}^{B_\phi - 1} a_i (\frac{1}{2})^i,\ a_i \in \{0, 1\}, \tag{13}$$

where B_ϕ is the quantization resolution of base element s. Here we keep one bit to store the integer part of s and remain $B_\phi - 1$ to represent the decimal part.

In addition, we reserve the integer part quantization resolution of the highest power of s, i.e., $(s^{L_{max}})^{M-1}$ to N_e-$B_{\lceil x \rceil}$ bits. $B_{\lceil x \rceil}$ is the required minimum quantitative resolution of $\lceil x \rceil$. Thus, to exactly represent this entry, the value of $(s^{L_{max}})^{M-1}$ is limited by

$$\lceil x (s^{L_{max}})^{M-1} \rceil \leq 2^{N_e - B_{\lceil x \rceil}}, \tag{14}$$

where L_{max}, M and N_e are pre-configured by the sink node. The feasible set of s can be obtained by Eq. (13) and Eq. (14).

Here, we further establish a simple example and obtain the corresponding set of feasible values of s. The topology is generated by the *BMST* algorithm designed in [21]. The longest path length of the network with 150 randomly distributed nodes is $L_{max} = 10$, thus we set $M = 11$. The resolution of each entry of weight vector is $B_\phi = 12$ bits and the resolution of each encoded data entry is $N_e = 24$ bits. The resolution of test data is $B_x = 12$. We implement the encoding process with each weight vector of coding set $\mathcal{D} \in \mathbb{R}^{11\times 10}$ which owns the structure as Eq. (7) shows. The feasible set of s in this scenario are shown in Table. 2.

We can validate the data recovery accuracy of the obtained feasible set through the metric *PRD* (the percent root-mean-square difference) [25], which is commonly used to measure the information loss. Its definition is given as follows.

$$PRD = 100 \sqrt{\frac{\sum_{n=1}^{N} \mid \widetilde{x[n]} - x[n] \mid^2}{\sum_{n=1}^{N} \mid x[n]^2 \mid}}, \tag{15}$$

where $x[n]$ and $\widetilde{x[n]}$ are the raw data acquired by node n and the recovered data by sink, respectively.

C. SECURITY ISSUE

In some key military scenarios, the sensor nodes in the network may encounter the node capture attack, which exacts the secret cryptographic keys (i.e. the parameter s in *VSDA*) and attack the communications of other nodes. Therefore, combining the feasible set of s obtained in the previous subsection, to tackle the security issue, we propose an enhanced-*VSDA* scheme which adds a key pooling mechanism [26], [27] to the data aggregation process.

At the initialization phase, the sink node generates a key pooling (e.g. $\{s_i\}_{i=n_1}^{n_q}$) with capacity q from the feasible set of keys for each path. Then, the sink node assigns the key pooling and an initial value of clock to the nodes on the same path. After that, each sensor node stores the received key pooling, M, and i in its memory. At the data aggregation time t, each node selects an s from the received key pooling according to a pseudo-random value produced by time t, and then generates the current weight vector. In this way, our proposed data aggregation scheme shows more robustness against the issue of the node capture attack.

D. DESIGN THE DIMENSION *M* OF CODING SET

As analyzed in Section. II-D, a larger value of M, namely the row dimension of *coding set*, is able to offer more complimentary space $M - L_{max}$ to allocate weight vectors to newly added nodes in the scaled-up scenario. However, it is unfeasible to set M to a large value directly because it will result in an obvious increase of the encoded data size ($N_e \times M$). If the encoded data size is much longer than the maximal payload size of a single data frame, the encoded data will be split into multiple data frames and sent out by several transmissions. A graphical description is shown in Fig. 5.

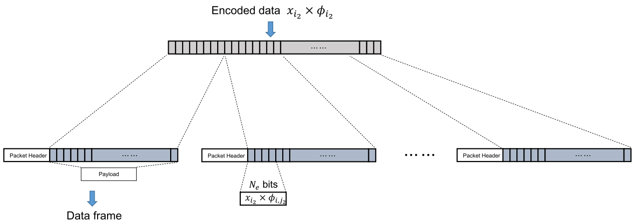

FIGURE 5. **Illustration of segmenting encoded data into multiple data frames.**

Thus, a single encoded data has to cost N_t transmissions to be sent out completely, in which N_t is given by

$$N_t = \lceil \frac{M \times N_e}{l_p} \rceil, \tag{16}$$

where l_p is denoted as the size of payload in a data frame. Therefore, we can not directly set the initial M a large value due to potentially increased amount of transmissions.

Since the length of a single data frame has little effect on the energy consumption, we consider to increase the data size $M \times N_e$ as much as possible while keeping the number of transmissions N_t unchanged. In this way, we can get a maximum of M when N_e and N_t are fixed, and thus we can obtain the larger complimentary space $M - L_{max}$.

Here, we call the maximum of M as the *critical point* M_{p0}, which offers the most complimentary space under the same amount of transmissions. The *critical point* can be given by

$$\frac{(M_{p0}+1) \times N_e}{l_p} \geq N_t \geq \frac{M_{p0} \times N_e}{l_p}, \tag{17}$$

which is equivalent to

$$\frac{l_p \times N_t}{N_e} - 1 \leq M_{p0} \leq \frac{l_p \times N_t}{N_e}, M_{p0} \in \mathbb{Z}. \tag{18}$$

Here we give a simple example to verify the *critical point* in *VSDA* scheme. Assuming a scenario with 200 randomly distributed nodes, and the topology is generated based on *BMST* algorithm [21]. The longest path length is 11 and thus we set the initial size of the weight vector as $M = 12$. In the designed data frame structure, the resolution of each encoded entry is fixed at $N_e = 40$ bits and the payload size l_p is 102×8 bits [30]. Here we verify the impact of the increased weight vector size M on the amount of data transmissions. This whole process is implemented in *PYTHON* language, the results are shown in Fig. 6.

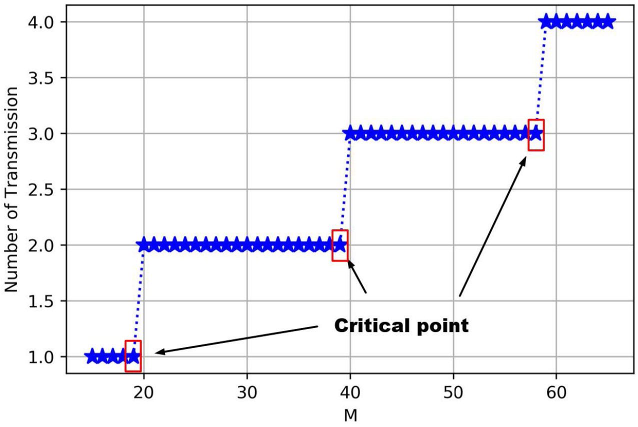

FIGURE 6. **The amount of transmissions versus the increasing value of *M*.**

For example, in the case of the initial row dimension of *coding set* is $M = 12$ and thus the amount of transmissions of each node is 1. Under the same amount of transmissions, the *critical point* can be obtained as $M_{p0} = 19$, which means that if the row dimension of *coding set* is not more than 19, the amount of transmissions of each node is still 1. In this case, we can update the pre-designed *coding set* row dimension to $M = 19$, which owns more complimentary space to generate new weight vectors and to accommodate newly added nodes without affecting the existing nodes. Meanwhile, under the same amount of transmissions, we can ignore the impact on transmission energy consumption by increasing data length ($N_e \times M$), which is negligible as presented in [31].

We further give some tests of *critical points* in the scenario, in which the number of nodes is 200 and the dimension of weight vector is $M = 11$. The resolution N_e is set to 25 bits, 30 bits, 35 bits, and 40 bits individually. The results are shown in Fig. 7.

IV. EVALUATE PERFORMANCE OF VSDA SCHEME

In this section, we compare the performance of our proposed scheme *VSDA* with classical *Gaussian* scheme [23], topology-aware scheme *TADA* [21], *Chain-like* scheme [14] and *Random* scheme [18] in terms of the amount of transmissions, energy consumption caused by network scaling and the

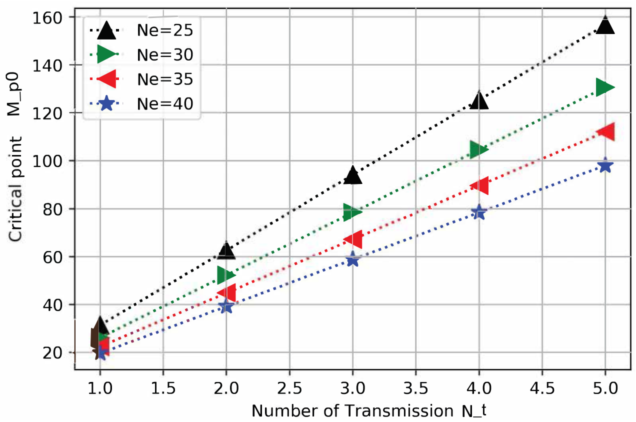

FIGURE 7. **The critical points versus the amount of transmissions.**

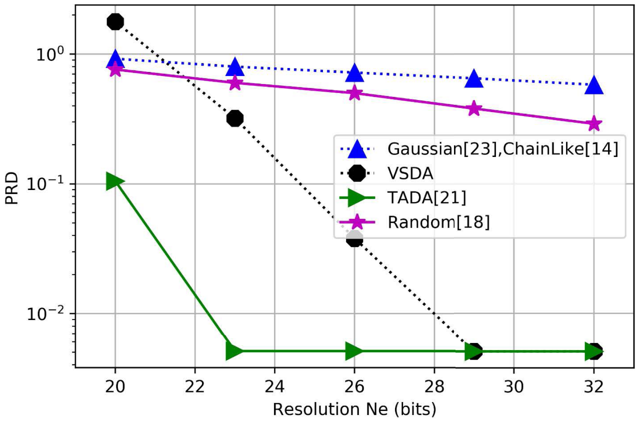

FIGURE 8. **The metric PRD versus encoded entry resolution N_e.**

required storage space. We simulate the scenario in PYTHON language, and our test data are random numbers ranging from 0 to 40.

A. TRADE-OFF BETWEEN DATA ACCURACY AND AMOUNT OF TRANSMISSIONS

As discussed in Section. III-D, the number of transmissions required to send a complete encoded data is determined by l_p, M and N_e. Here, with l_p and M fixed, we further discuss the impact of N_e on the amount of transmissions and data accuracy between different data aggregation schemes.

Here, we first define the amount of transmissions in the entire whole network as the sum of transmissions required to gather a group of network-wise data, which is given by

$$N_{ad} = N_t \times N_{up}, \tag{19}$$

where N_{up} is the total number of uplink transmissions for collecting a group of network-wise data, and N_t was defined in Eq. (16).

Since the energy consumption in data transmission is more significant than any other functions within a sensor node, reducing the required number of transmissions for each encoded data vector, namely N_t, is an effective way to prolong the lifetime of a sensor.

As given in Eq. (16), it is straightforward to find that the fewer transmissions N_t is obtained if the size of each encoded entry N_e is small enough. However, notice that reducing N_e, the quantization resolution of each encoded entry, may cause serious information loss during encoding process. The information loss can be measured as the metric *PRD* (see Eq. (15)). The lower the value of *PRD*, the higher the accuracy of the restored data. Thus, we test the relationship between quantization resolution N_e and the metric *PRD* of different schemes. The resolution of raw data is set as $B_x = 12$ bits. The topology in the network is generated by the BMST algorithm [21], and the longest path length of the network with 150 randomly distributed nodes is $L_{max} = 10$, thus we set $M = 11$ in *TADA* and *VSDA*. In additional, the base element s in *VSDA* is 1.125. The results are shown in Fig. 8.

Due to the fact that the larger resolution N_e means the less information is lost during data transmission process. Thus, we prefer to set the resolution N_e as larger as possible. As shown in Fig. 8, the performance of *TADA* is superior because its column vectors are orthogonal to each other, which guarantees a stronger property in the face of quantization errors. By comparison, the *Gaussian* scheme, *Chain-like* scheme and *Random* scheme are more susceptible to the quantization errors. Our proposed scheme, *VSDA*, has the almost same performance as *TADA* with the increase of quantization resolution N_e.

However, according to the Eq. 16, the larger resolution N_e means more transmissions N_t. So, there is a trade-off between data accuracy and transmission quantity. Then, we further put forward some tests to exhibit the relationship between resolution N_e and the amount of transmissions. We consider a WSN network with N randomly and uniformly distributed nodes. Initially, we generate the network topology based on [21] and obtain the *coding set* $\mathcal{D} \in \mathbb{R}^{M \times L_{max}}$ ($M \geq L_{max}$) with the structure designed in Eq. (7). In addition, we apply the data unit which works with the *IEEE 802.15.4* MAC layer and the corresponding data frame in PHY layer supports maximum 102 bytes of payload [30]. We use the ADC with $B_x = 12$ bits to digitize the acquired raw datum x_i. The resolution of each encoded entry, N_e, is allowed to be as larger as needed and with a minimal value equivalent to the resolution of raw data, B_x. The raw sensed data of node i is x_i and node i will encodes it with the weight vector randomly selected from $\mathcal{D}$ with the base element $s = 1.125$. The encoding process is presented as Fig. 3 shows. The test results of the amount of transmissions N_{ad} versus resolution N_e in the scenario with $N = 150$ nodes are given in Fig. 9.

Obviously, the *Random* scheme results in a large amount of transmissions during the data aggregation process. Unlike other schemes that directly transmit the whole encoded data vector, the *Random* scheme transmits only one entry of the encoded data vector at a time. So, it costs more uplinks (N_{up}) than other schemes. We did not show the results of *TADA* scheme because its performance is almost the same as *VSDA*

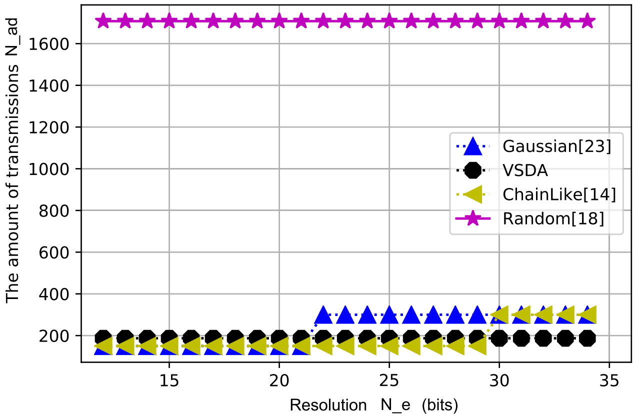

FIGURE 9. The amount of transmission N_{ad} versus encoded entry resolution N_e in the scenario with 150 nodes.

scheme. To clearly show the performance comparison results, we further test the performance of *VSDA* scheme, *Gaussian* scheme and *Chain-like* scheme in four different scenarios with 150 nodes, 200 nodes, 250 nodes and 300 nodes, respectively. The corresponding results are given in Fig. 10.

As defined in Eq. (19), the amount of transmissions N_{ad} is directly related to the number of uplinks (N_{up}) and the amount of transmissions required to transmit an encoded data vector (N_t). As shown in Fig. 10, due to the fact that the transmission processes in *VSDA* are independent among all paths, it naturally consumes more uplinks (N_{up}) in data aggregation process. Therefore, the transmission amount N_{ad} in *VSDA* is slightly higher than the other two schemes as the resolution N_e is low. However, as the resolution N_e increases, the transmission quantities of the other two schemes, *Gaussian* and *Chain-like*, increase significantly after the resolution N_e exceeds the critical points (such as the resolution $N_e = 21$ bits in Fig. 10(b)). By comparison, the *VSDA* scheme keeps unchanged in our tests, and the reasons are given as follows.

For *Gaussian* scheme, the length of the encoded data vector is given by $M \times N_e$, in which the parameter M is decided by the number of nodes. The relationship between M and N is limited by the CS-property, which is given as follows.

$$M \geq Cklog\frac{N}{k}, \tag{20}$$

where C is a constant [32] and k is set to $N/10$ here. Thus, as the number of nodes N increases, the size of the produced encoded data ($M \times N_e$) increases synchronously. Meanwhile, the parameter M in *Chain-like* scheme is also restricted by the CS-property, but its value is slightly less than that in *Gaussian* scheme due to the adopted hierarchical transmission mechanism. By contrast, since the parameter M in *VSDA* is equal to the longest path length in the network, which is not much sensitive to the increased total number of nodes, so, the length of the encoded data in *VSDA* scheme is naturally much shorter than that in above two schemes. However, as mentioned in Sec. III-D, once the length of the encoded data vector $M \times N_e$ exceeds the payload size of a data frame (see Fig. 5), it has to be segmented into several parts and send one part in one transmission process. Therefore, *Gaussian* scheme and *Chain-like* scheme are more costly when a higher resolution N_e is required. The *VSDA* scheme does not require more transmission processes due to its data vector length is much shorter than its critical points (see Fig. 7), which provides more efficient performance in data aggregation process.

B. PERFORMANCE ANALYSIS IN A SCALED-UP SCENARIO

In this subsection, we compare the performance of the energy consumption caused by scenario scaling-up and the required storage capacity between different schemes.

1) ENERGY CONSUMPTION CAUSED BY SCENARIO SCALING-UP

As analyzed in Section. II, the most serious impact by adding a new node in the scenario is that it may change the weight vectors of the original nodes, which consumes more energy in re-assigning weight vectors from a newly generated matrix. Since the energy cost in data transmission is greater than any other functions within a wireless sensor, the energy dissipation of transmission during scenario scaling-up is the mainly considered metric in this paper. Here, assume that the sink node has limited communication power, so that it can only pass the update weight vector to its neighbor nodes. Therefore, for non-neighbor nodes, the update weight vectors will be relayed through the neighbor nodes of the sink node and transmitted by multi-hop routing.

Assume that a new node joins in the longest path in the network, the impact of the existing scenario can be measured as updating the weight vectors to the existing sensor nodes. The energy dissipation E_o caused by the vector re-allocation process can be formulated as

$$E_o = (N_t \times E_{d_{ave}}) \times N_{down}, \tag{21}$$

where N_t is defined in Eq. (16), and $E_{d_{ave}}$ is defined as the energy dissipation [31] in transmitting one data frame between adjacent node, which can be considered as a constant in simulation test. N_{down} is the number of downlink transmissions needed for weight vector updating.

We perform several simulations to compare the performance of the above five schemes. Assuming the tested scenario consists of N randomly distributed nodes, and the topology is generated by *BMST* algorithm [21]. The resolution of raw data and each encoded entry are set to $B_x = 12$ bits and $N_e = 14$ bits here, respectively. The energy consumption in passing one data frame between adjacent nodes, $E_{d_{ave}}$, is set to $8mJ$. The energy consumption caused by adding one new node in the existing scenario is shown in Fig. 11. Here, l_p is set to be 102×8 bits [30].

For the *Gaussian* scheme, the weight vectors are picked from the measurement matrix which is produced by *Gaussian distribution* integrally. The length of weight vector, namely the row dimension of measurement matrix, is decided by the

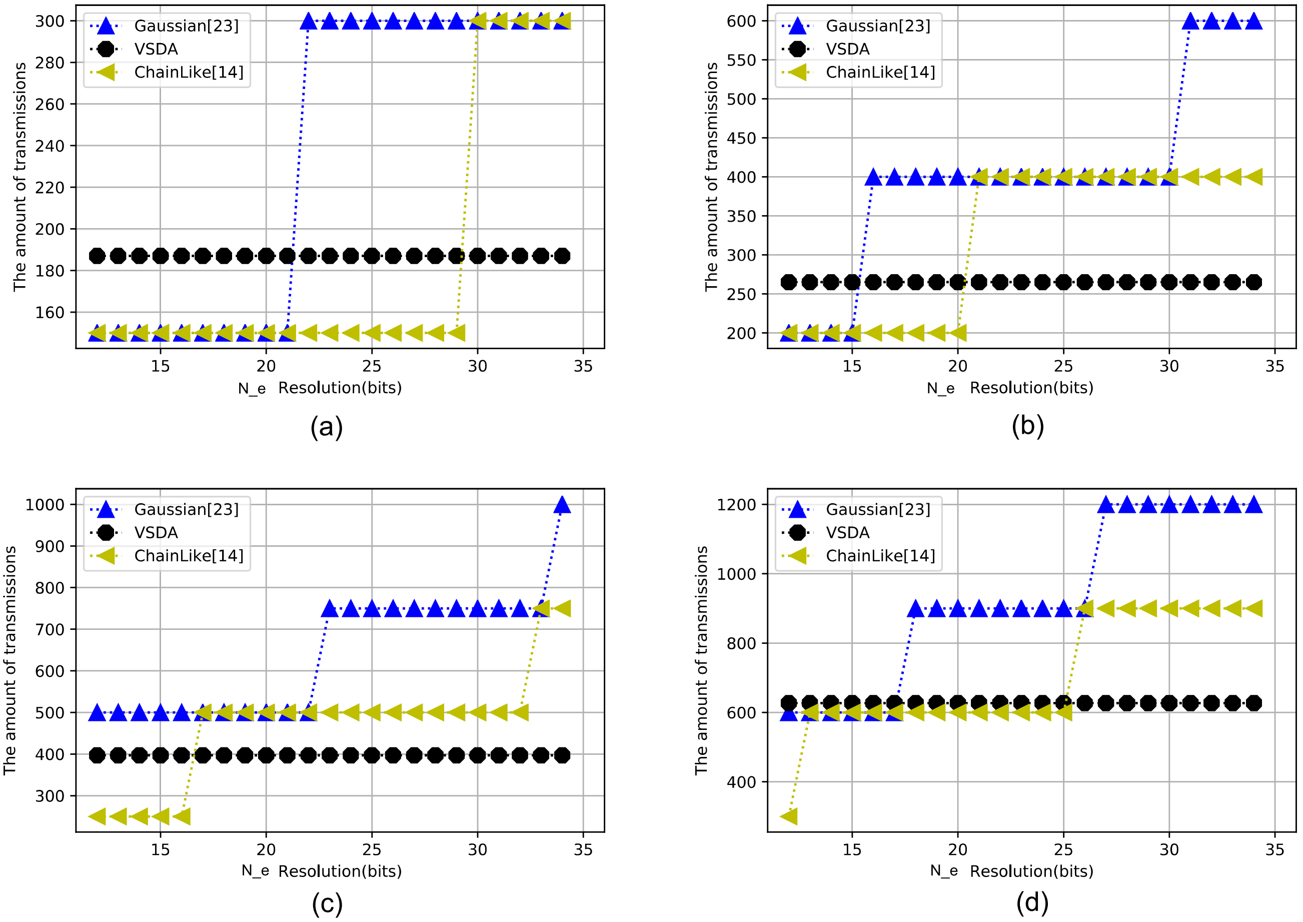

FIGURE 10. The amount of transmissions versus the resolution of encoded entry N_e, in which the amount of nodes in (a)-(d) are 150, 200, 250, and 300, respectively.

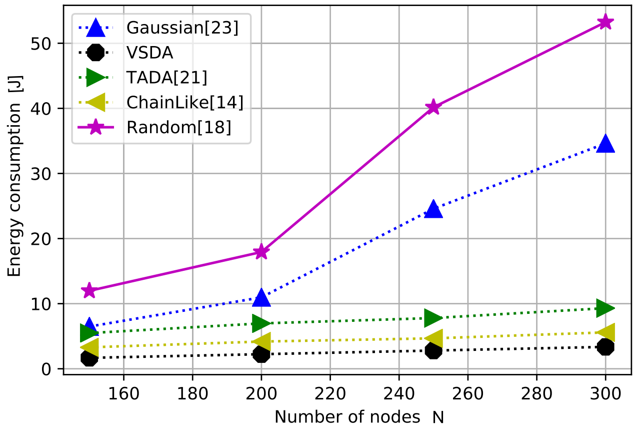

FIGURE 11. The energy consumption of different schemes as a new node joins the network.

total number of nodes in the network (see Eq. (20)). Therefore, to meet the CS property, the more reliable way for this scheme is re-generating a new matrix with a larger size as a new node joins the scenario and allocating the column vector of the new matrix to each node. Meanwhile, the number of downlinks N_{down} is significantly larger than the total number of nodes N.

For *TADA* scheme, it will face the same problem as happened in *Gaussian* scheme as the new node joins the longest path in the topology. Moreover, such the worst case is inevitable as the number of newly added nodes increases. Meanwhile, the number of downlinks N_{down} in *TADA* scheme is significantly larger than the total number of nodes N.

In *Random* scheme, it has been proven that the designed measurement matrix can satisfy the CS-condition if the number of non-zeros in each row (i.e., t) and the row dimension of the matrix (i.e., M) have to meet the conditions $t = \lceil \frac{3N}{k} \rceil$ and $M = \lceil 2k \times log(N/k) \rceil$ simultaneously, where N is the total number of nodes in the network and k is the sparsity of the raw data. In the scaled-up scenario, the total number of nodes increases from N to $N+1$ as a new node joins in the network. Consider an inevitable case that the previous parameter t or M is not feasible for the current scenario as N increases, in this case, the *Random* scheme has to re-generate a new matrix and re-assign its column vector to each node by multi-hop routing.

By comparison, regardless of each scaling-up case in *VSDA* scheme, the sink node only needs to transmit the updated

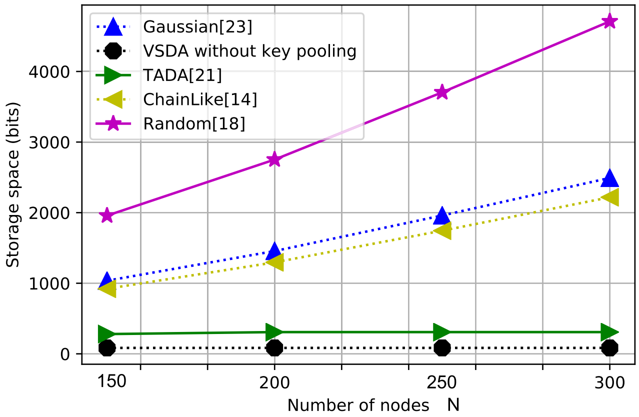

FIGURE 12. The required storage space of different schemes to store the weight vector in a sink node.

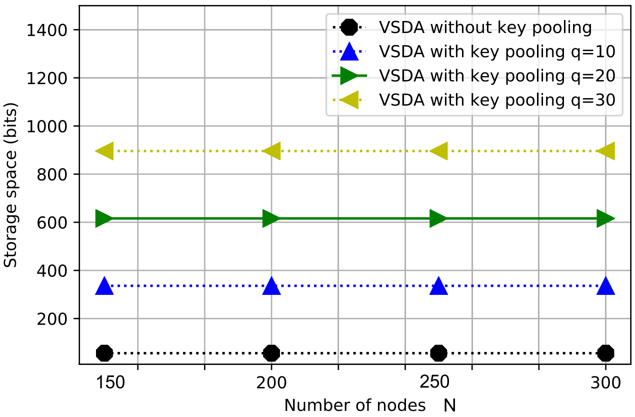

FIGURE 13. The required storage space of the enhanced-VSDA scheme with q-capacity.

parameter M to these prior nodes. Since the M used by all nodes is the same, each node only need to pass it to its child-nodes, and thus the number of downlinks N_{down} in *VSDA* is equal to the total number of nodes N.

For *Chain-like* scheme, it achieves a good performance due to the adopted hierarchical topology structure. That is, adding a new node will only affect the nodes which are distributed under the same sub-tree. Then, the sink node only need to assign the newly generated weight vectors to theses influenced nodes.

In addition to the above analysis, the results also show that the energy dissipation in *VSDA* scheme increases slowly as the total number of nodes in the network increases. The reason is that the weight vector in *VSDA* scheme is selected from the *coding set* whose dimension is closely related to the longest path of topology instead of the total number of nodes in the network. Therefore, the transmission load in *VSDA* scheme will not increase dramatically by just adding a new node. The similar situation will occur in *Chain-like* scheme and *TADA*. By comparison, the transmission loads (*i.e.*, $M \times N_e$) in other schemes are directly determined by the total number of nodes, which increases significantly as the number of nodes increases.

Remark 1: Assume that the sink node has large transmission range, and is able to broadcast the weight vector to all the nodes directly, the proposed scheme will be more efficient because all nodes only need to receive a broadcast about the update parameter M, while other schemes need to broadcast multiple times.

2) REQUIRED STORAGE SPACE

Finally, due to the limited memory within a typical wireless sensor node, the required memory space to store the weight vector is also an important indicator to measure the performance of the aggregation scheme. The required storage space of storing the weight vector in a sensor node is given by

$$V = B_{\phi} \times M, \tag{22}$$

where M is the weight size and B_{ϕ} is the corresponding resolution of each entry of the weight vector. Here, we put forward some simulation tests to compare the required practical storage space for storing the weight vector between different aggregation schemes. Assuming the tested scenario consists of N randomly distributed nodes, and the topology is generated based on *BMST* algorithm [21]. The resolution of each entry of weight vector is set to $B_{\phi} = 12$ bits here. The results are given as follows.

Obviously, without adding the key pool mechanism, our proposed *VSDA* scheme does not require a large amount of storage space due to the regular structure of *coding set*. The weight vector of *VSDA*, which is picked from *coding set*, is easily obtained by only offering the base element s, M and $i \in [1, 2, \ldots, L_{max}]$. Thus, each sensor node does not need to store the full-size weight vector. Also, for the *TADA* scheme, the required storage space will not increase significantly because the length of its weight vector only depends on the length of the longest path in the network, which is not very sensitive to the total number of nodes. By contrast, the weight vectors stored in other schemes need a large storage burden in the sensor nodes.

We further test the required storage space in the enhanced-VSDA scheme. The corresponding results are given as follows.

The parameter q represents the capacity of the key pooling in enhanced-VSDA scheme. For example, $q = 10$ means that the sink node assigns 10 feasible values of base element s to sensor nodes at the initialization phase. It can be seen that the space required for the enhanced-VSDA scheme has multiplied after combining the key pooling mechanism, but at the same time, can be more robust against node capture attacks.

V. CONCLUSION

This paper proposes a data aggregation protocol named *VSDA* to alleviate the issues incurred by scaling-up a WSN scenario. Inspired by CS-based schemes, *VSDA* also encodes raw data of sensor nodes with weight vector, and decodes the raw data at the sink node with the measurement matrix which is formed by weight vectors. The existing schemes, including *Gaussian* scheme, *Random* scheme, *Chain-like* scheme and *TADA* scheme, face the challenge of high energy consumption when a new node is added to the network. In this case, our scheme *VSDA* proposes a new structure of weight vector, which exhibits strong scalability to address the problem caused by network expansion. In addition, we verify the performance of *VSDA* in terms of energy consumption, data accuracy, and storage space under an implementation framework, which is more realistic to show the practical performance. The simulation results show that our proposed *VSDA* outperforms others in all performance terms. However, our tests ignore the other affecting factors such as channel noise, link failure and other effects from hardware circuit. It poses a significant challenge to implement the data aggregation scheme on actual hardware, which will be tackled in our future work.

REFERENCES

[1] A. Boubrima, W. Bechkit, and H. Rivano, "Optimal WSN deployment models for air pollution monitoring," *IEEE Trans. Wireless Commun.*, vol. 16, no. 5, pp. 2723–2735, May 2017.

[2] M. Z. A. Bhuiyan, J. Cao, and G. Wang, J. Wu, "Deploying wireless sensor networks with fault-tolerance for structural health monitoring," *IEEE Trans. Comput.*, vol. 64, no. 2, pp. 382–395, Feb. 2015.

[3] D. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2009, pp. 1–14.

[4] R. C. Shah and J. M. Rabaey, "Energy aware routing for low energy ad hoc sensor networks," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2002, pp. 350–355.

[5] G. Hua and C. W. Chen, "Correlated data gathering in wireless sensor networks based on distributed source coding," *Int. J. Sensor Netw.*, vol. 4, no. 1, pp. 13–22, Jan. 2008.

[6] J. H. Chou, D. Petrovic, and K. Ramachandran, "A distributed and adaptive signal processing approach to reducing energy consumption in sensor networks," in *Proc. IEEE Comput. Commun.*, vol. 2, Mar. 2003, pp. 1054–1062.

[7] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[8] E. J. Candès, J. Romberg, and T. Tao, "Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Jun. 2006.

[9] E. J. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[10] X. Xu, R. Ansari, A. Khokhar, and A. V. Vasilakos, "Hierarchical data aggregation using compressive sensing (HDACS) in WSNs," *ACM Trans. Sensor Netw.*, vol. 11, no. 3, pp. 1–25, 2015.

[11] M. T. Nguyen, K. A. Teague, and N. Rahnavard, "CCS: Energy-efficient data collection in clustered wireless sensor networks utilizing block-wise compressive sensing," *Comput. Netw.*, vol. 106, no. 1, pp. 171–185, Sep. 2016.

[12] X. Li, X. Tao, and G. Mao, "Unbalanced expander based compressive data gathering in clustered wireless sensor networks," *IEEE Access*, vol. 5, pp. 7553–7566, 2017.

[13] C. Liu, S. Guo, Y. Shi, and Y. Yang, "Deterministic binary matrix based compressive data aggregation in big data WSNs," *Telecommun. Syst.*, vol. 66, no. 3, pp. 345–356, 2017.

[14] K.-C. Lan and M.-Z. Wei, "A compressibility-based clustering algorithm for hierarchical compressive data gathering," *IEEE Sensors J.*, vol. 17, no. 8, pp. 2550–2562, Apr. 2017.

[15] L. Yin, C. Liu, S. Guo, and Y. Yang, "Sparse random compressive sensing based data aggregation in wireless sensor networks," *Concurrency Comput. Pract. Exper.*, vol. 4, no. 8, pp. 44–55, 2018.

[16] S. P. Tirani and A. Avokh, "On the performance of sink placement in wsns considering energy-balanced compressive sensing-based data aggregation," *J. Netw. Comput. Appl.*, vol. 107, pp. 38–55, Apr. 2018.

[17] D. G. Zhang, T. Zhang, J. Zhang, Y. Dong, and X.-D. Zhang, "A kind of effective data aggregating method based on compressive sensing for wireless sensor network," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 159–174, 2018.

[18] 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 Trans. Parallel Distrib. Syst.*, vol. 26, no. 1, pp. 35–44, Jan. 2015.

[19] Y. Li and Y. Liang, "Compressed sensing in multi-hop large-scale wireless sensor networks based on routing topology tomography," *IEEE Access*, vol. 6, pp. 27637–27650, 2018.

[20] X. Liu *et al.*, "CDC: Compressive data collection for wireless sensor networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 8, pp. 2188–2197, Aug. 2015.

[21] X. Wang, Q. Zhou, and C.-T. Cheng, "A UAV-assisted topology-aware data aggregation protocol in WSN," *Phys. Commun.*, vol. 34, pp. 48–57, Jun. 2019.

[22] W. Du, J. Deng, Y. S. Han, P. K. Varshney, J. Katz, and A. Khalili, "A pairwise key predistribution scheme for wireless sensor networks," *ACM Trans. Inf. Syst. Secur.*, vol. 8, no. 2, pp. 228–258, 2005.

[23] J. Wang, S. Tang, B. Yin, and X.-Y. Li, "Data gathering in wireless sensor networks through intelligent compressive sensing," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 603–611.

[24] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic, "Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors," *IEEE J. Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, Mar. 2012.

[25] F. Chen, F. Lim, O. Abari, A. Chandrakasan, and V. Stojanovic, "Energy-aware design of compressed sensing systems for wireless sensors under performance and reliability constraints," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 60, no. 3, pp. 650–661, Mar. 2013.

[26] J. Zhao, "On resilience and connectivity of secure wireless sensor networks under node capture attacks," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 3, pp. 557–571, Mar. 2017.

[27] M. V. Bharathi *et al.*, "Node capture attack in wireless sensor network: A survey," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res.*, 2012, pp. 1–3.

[28] N. A. Cloete, R. Malekian, and L. Nair, "Design of smart sensors for real-time water quality monitoring," *IEEE Access*, vol. 4, pp. 3975–3990, 2016.

[29] C. M. G. Algora, V. A. Reguera, N. Deligiannis, and K. Steenhaut, "Review and classification of multichannel MAC protocols for low-power and lossy networks," *IEEE Access*, vol. 5, pp. 19536–19561, 2017.

[30] A. Muhammad, Y. Hongnian, and C. Shuang, "IEEE 802.15.4 frame aggregation enhancement to provide high performance in life-critical patient monitoring systems," *Sensors*, vol. 17, no. 2, pp. 241–266, 2017.

[31] C. Karakus, A. C. Gurbuz, and B. Tavli, "Analysis of energy efficiency of compressive sensing in wireless sensor networks," *IEEE Sensors J.*, vol. 13, no. 5, pp. 1999–2008, May 2013.

[32] E. J. Candès, "Compressive sampling," in *Proc. Int. Congr. Math.*, vol. 3. Madrid, Spain, 2006, pp. 1433–1452.

XINDI WANG received the B.Sc. degree from Anhui Normal University, Wuhu, China, in 2012, and the master's degree from the Hefei University of Technology, Anhui, China, in 2015, where he is currently pursuing the Ph.D. degree. He is also a Visiting Scholar with the Dongguan University of Technology. His research interests include compressive sensing, wireless sensor networks, and machine learning.

QINGFENG ZHOU received the B.Sc. degree from the University of Science and Technology of China (USTC), Hefei, China, the M.Sc. degree from Clemson University, SC, USA, and the Ph.D. degree in information engineering from The Hong Kong Polytechnic University, Hong Kong, in 2010. He is currently a Professor with the Dongguan University of Technology. His research interests include wireless communications, particularly focusing on interference alignment, distributive MIMO, sensor networks, and smart wearable devices.

JUN TONG received the B.E. and M.E. degrees from the University of Electronic Science and Technology of China (UESTC), in 2001 and 2004, respectively, and the Ph.D. degree in electronic engineering from the City University of Hong Kong, in 2009. He is currently a Senior Lecturer with the School of Electrical, Computer and Telecommunications Engineering, University of Wollongong, Australia. His research interest includes signal processing and its applications to communication systems.