Efficient Compression Algorithm with Limited Resource for Continuous Surveillance

Ling Yin[1], Chuanren Liu[2], Xinjiang Lu[3], Jiafeng Chen[1] and Caixing Liu[1]
[1] College of Mathematics and Informatics, South China Agricultural University, China
[e-mail: yin_ling@scau.edu.cn]
[2] Department of Decision Sciences and Management Information Systems, Drexel University, United States
[e-mail: chuanren.liu@drexel.edu]
[3] School of Computer Science, Northwestern Polytechnical University, Xi'an China
[e-mail: xjlu@mail.nwpu.edu.cn]

Received June 19, 2016; revised September 2, 2016; accepted September 28, 2016; published November 30, 2016

Abstract

Energy efficiency of resource-constrained wireless sensor networks is critical in applications such as real-time monitoring/surveillance. To improve the energy efficiency and reduce the energy consumption, the time series data can be compressed before transmission. However, most of the compression algorithms for time series data were developed only for single variate scenarios, while in practice there are often multiple sensor nodes in one application and the collected data is actually multivariate time series. In this paper, we propose to compress the time series data by the Lasso (least absolute shrinkage and selection operator) approximation. We show that, our approach can be naturally extended for compressing the multivariate time series data. Our extension is novel since it constructs an optimal projection of the original multivariates where the best energy efficiency can be realized. The two algorithms are named by ULasso (Univariate Lasso) and MLasso (Multivariate Lasso), for which we also provide practical guidance for parameter selection. Finally, empirically evaluation is implemented with several publicly available real-world data sets from different application domains. We quantify the algorithm performance by measuring the approximation error, compression ratio, and computation complexity. The results show that ULasso and MLasso are superior to or at least equivalent to compression performance of LTC and PLAMlis. Particularly, MLasso can significantly reduce the smooth multivariate time series data, without breaking the major trends and important changes of the sensor network system.

This research was supported by the China Scholarship Council under Grant No. 201408440035.

http://dx.doi.org/10.3837/tiis.2016.11.015 **ISSN : 1976-7277**

Keywords: lossy compression algorithm, fused lasso, wireless sensor network, energy efficiency

1. Introduction

Nowadays, real-time monitoring and surveillance supported by large-scale wireless sensor networks (WSNs) are frequently employed in commercial applications, such as environmental monitoring [23], remote patient monitoring in healthcare [9], animal behavior classification [7], and structural health monitoring for infrastructures [10]. For the best of application performance, high speed and continuous sensor sampling are needed, which will generate large volume of raw sensor data. Accordingly, transmitting and storing the sensor data are very energy expensive. For example, if all the sampled data need to be transmitted to the base station, battery used by the remote wireless sensor node will be drained quickly. Indeed, Kimura and Latifi [8] concluded that approximately 80% of power consumed in each sensor node is used for data transmission; Barr and Asanovic [4] showed that the energy consumed for transmitting a single bit of information is approximately the same as that required by the processing unit for executing a thousand of computing operations. In sum, the resource-constraints imply significant challenges to the operating and managing of large-scale WSNs in commercial applications.

There are several strategies developed to reduce energy consumption and prolong the lifetime of sensor nodes, such as battery replenishment, reducing the sampling rate, and data compression [6]. In many real physical environments, replenishment of batteries can be even more expensive and impractical. Lower sampling rate can shrink the data volume for transmitting and storing, but the reduced data often lose informative details of the monitored system. A much higher sampling rate will be necessary for the best of data quality and application performance. On the other hand, the data compression is more promising in improving the energy efficiency and reduce the energy consumption of WSNs. The reason is that, data compression algorithms can allow some level of inaccuracy in the recovered signal as long as major trends and important changes are preserved [22]. Using data compression method means that we can trade some energy for computing the compression to reduce much more energy for data transmission and storage.

There have been some well-known solutions to compress time series sensor data. However, most of them are presented and test on single variate sensor data such as temperature, relative humidity, etc. In many practical and complicated scenarios the WSNs are deployed to collect multi-dimensional time series. For example, three-dimensional moving acceleration data is collected to monitor the motion states and three-dimension magnetometer is used to measure the direction of magnetic field. Moreover, the EEG (Electrocardiography) heart or brain data can be of over one hundred of dimensions. Although the WSNs with capturing multi-dimensional data are more common, few literature works have discussed the compression of the multivariate time

series sensor data. Therefore, this paper proposes novel data compression algorithms suitable for both univariate and multivariate systems. Major contributions of this work are highlighted as follows:

- We develop efficient algorithms which perform lossy compression on both univariate and multivariate time series data. Our algorithms (ULasso and MLasso) are based on Lasso approximation. In particular, our multivariate extension (MLasso) can construct the optimal projection of the original multivariates where the best energy efficiency can be realized.
- In addition to the theoretical formulation of ULasso and MLasso, we also develop efficiency solvers which can compute the optimal compression with much less running time. We also provide practical guidance on selecting important parameters in our algorithms.
- We demonstrate the superiority of our algorithms in comparison with the state-of-the-art competing algorithms on several publicly available real-world data sets from different application domains. The experimental results show that our algorithms perform significantly better, particularly for smooth multivariate time series data sets, which are typical in applications of behavior monitoring.

The remainder of this paper is structured as follows: Section 2 presents related work. And then we present univariate and multivariate lasso compression algorithms, as well as the implementation details in Section 3. Section 4 describes several real-world data set and their characteristic, then compares various methods for sensor data compression. Finally, Section 5 concludes the paper.

2. Related work

Energy efficiency is becoming more important for wireless sensors which aims at long-term and real-time continuous monitoring. Limited by the node processor computation and storage, many existing compression algorithms are not directly applicable for sensor nodes. Only specifically designed compression algorithms are suitable for tiny sensor nodes applications [29]. The dedicated data compression techniques for sensor nodes can be classified into two categories: (1) spatial compression techniques, and (2) temporal compression techniques.

(1) Spatial compression techniques. This kind of techniques exploit distributed compression to reduce the transmission data on cooperative and dense network nodes [22,11]. This compression technique needs nodes collaborate with each other to carry out tasks. Specifically, the topology of the wireless sensor network of neighboring nodes needs to be considered. Many researchers discussed the distributed compression approaches, such as distributed source modeling (DSM) [17], distributed transform coding (DTC) [3], distributed source coding (DSC) [19], and compressed sensing (CS) [5], etc. A popular application of spatial compression techniques is image compression. This technique leverages the following observation, that is, one pixel often reveals similar information against its neighbors which exhibits high spatial correlation.

(2) Temporal compression techniques. Exploiting temporal correlation usually acts on a node and involves more traditional data compression techniques. Due to the limited resource, sensor nodes are usually not capable to communicate with each other,. Not like spatial distributed compression approaches, there few works focusing on temporal compression algorithms.

Temporal compression techniques applied in a single node can also be classified into two categories: lossless and lossy compression algorithms, which are typically based on different principles. Lossless compression ensures the correctness of information by removing redundancy from data during compression and decompression process, which preserve the data accuracy [22]. On the contrary, lossy compression techniques emphasize the higher compression ratio which will discard some of the original information [22]. Obviously, with the higher compression ratio, lossy compression gets the less data quality.

2.1. Lossless compression algorithm

Several typical lossless compression techniques have been developed and discussed to apply in the sensors. For example, one well-known method adopted in sensor network is Sensor Lempel-Ziv-Welch (S-LZW) algorithm [3], which is dictionary-based lossless compression approach. Another efficient lossless compression approach is Lossless Entropy Compression(LEC) algorithm, which is based on traditional information encoding, requires very low computational power and achieves higher compression ratio [15]. For wireless sensors, the lossless compression techniques are not suitable due the resource limited environment.

2.2. Lossy compression algorithm

Lossy compression loosens the tolerable observation error margins to achieve the flexibility on trading off between reconstruction accuracy for higher compression ratio and less energy consumption, which can lengthen the lifetime of wireless sensors in turn. A very low-complexity Lightweight Temporal Compression(LTC) technique is presented by Schoellhammer, which introduce a small amount of error into each reading, bounded by a control knob [21,14]. LTC adopted by piecewise linear approximation approaches with the low-complexity are widely used in sensor network. Unfortunately it performs poorly if the sensor readings fluctuate frequently, even when the fluctuations follow some fixed patterns over time and can only be used for temporal data compression [1]. Based on piecewise linear approximation scheme, Piecewise Linear Approximation with Minimum Number of Line Segments (PLAMLiS) algorithm exploits the optimal minimum number of segments to approximate the given time series such that the difference between any approximation value and its actual value is less than the given error bound ε [18].

The above two well-known temporal lossy compression algorithms use piecewise linear to represent time series, and the piecewise linear technique seems suitable for varying slowly and gradually signals like temperature. However, the LTC and PLAMlis

are not effective to handle higher dimensional or multivariate time series. For multivariate or multi-dimensional temporal data, the feasible solution for existing lossy compression techniques is that separate multivariate or multi-dimensional data into several one-dimension arrays to compress, which is obviously not effective and efficient enough. In this paper, we provide compression method for both univariate and multivariate time series in sensor networks.

3. The compression algorithm

3.1. The univariate algorithm

We start with the simple scenario where we have observed the one-dimensional reading $y = (y_1, y_2, \dots, y_N)$ and the corresponding time stamps $t = (t_1, t_2, \dots, t_N)$ where $t_1 \le t_2 \le \cdots \le t_N$. In other words, we observed the value y_i at time t_i from the sensor. For example, the sensor reading in y can be used to measure the blood pressure of a cow in the time window (t_1, t_N). It is often that the reading is constantly changing and subsequently all reading values in *y* and time stamps in *t* are stored in data base for modeling and analysis. Nonetheless, during some time period, the changes in the reading can be very insignificant. In the extreme case, suppose we have observed that the $y_p = y_{p+1} = \cdots = y_q$ for some $p < q$. In this case, we could store in the data base the reading $y = (\dots, y_p, y_{q+1}, \dots)$ with corresponding time stamps $t = (\dots, t_p, t_{q+1}, \dots)$ to reduce data storage. Note that, by identifying such 'smooth' time window (t_p, t_q) we can save the storage space for $|p - q|$reading values and also time stamps. Moreover, by shrinking the storage space, we can also reduce the energy used to transmitting the data. As shown later in Section 4, the economic benefit of reducing the data transmission and storage can be very significant.

In practice, the reading values may not be exactly constant but with slight perturbations in some time period, e.g., the time window (t_p, t_q). To compress the time series data in these cases, we present an efficient algorithm based on Lasso regularization [13]. The Lasso regularization has been utilized in applications including image denoising [13], comparative genomic hybridization [24], prostate cancer analysis [25], and time-varying networks [2], where features can be ordered in some meaningful way.

Specifically, to remove the slight perturbations in *y*, we compute a smoother approximation, *x*, by minimizing the regularized differences:

$$J(x) = \frac{1}{2}\sum_{n=1}^{N}(y_n - x_n)^2 + \lambda \sum_{n=1}^{N-1}|x_{n+1} - x_n| \quad (1)$$

Specifically, while we use the first term $\frac{1}{2}\sum_{n=1}^{N}(y_n - x_n)^2$ to minimize the differences between the actual reading *y* and approximation *x*, we use the second term (fused lasso) to encourage the piecewise constant in the approximation *x*. The degree of the regularization, λ, can control the smoothness of the approximation. The larger the value of λ is, the smoother the solution *x* will be.

3.2. The multivariate algorithm

Now we continue to the general setting where our observation can be multivariate time series $Y \in R^{D\times N}$. As aforementioned for the univariate case, N is still the number of observations and the corresponding time stamps are $\mathrm{t} = (\mathrm{t}_1, \mathrm{t}_2, \dots, \mathrm{t}_N)$ with $t_1 \le t_2 \le \cdots \le t_N$. However, now we have totally D readings at each time stamp. Specifically, by letting $Y_{d*} = Y_{d1}, Y_{d2}, \dots, Y_{dN}$ be the d-th row in the matrix Y, we observe D readings $Y_{1n}, Y_{2n}, \dots, Y_{Dn}$ at the n-th time stamp t_n.

To extend the univariate algorithm to suit the multivariate case, one straightforward formulation is as follows:

$$J(X) = \sum_{d=1}^{D} J(X_{d^*}) \tag{2}$$

where X_{d^*} is the d-th row in the solution matrix $X \in R^{D\times N}$ and $J(X_{d^*})$ is the univariate objective function:

$$J(X_{d^*}) = \frac{1}{2}\sum_{n=1}^{N}(Y_{dn} - X_{dn})^2 + \lambda \sum_{n=1}^{N-1}\left|X_{d,n+1} - X_{dn}\right| \tag{3}$$

Therefore, it follows that

$$\begin{aligned} J(X) &= \sum_{d=1}^{D} J(X_d) \\ &= \sum_{d=1}^{D}\left(\frac{1}{2}(Y_{dn} - X_{dn})^2 + \lambda \sum_{n=1}^{N-1}\left|X_{d,n+1} - X_{dn}\right|\right) \\ &= \frac{1}{2}\sum_{n=1}^{N}\|Y_{*n} - X_{*n}\|^2 + \lambda \sum_{n=1}^{N-1}\left|X_{*,n+1} - X_{*n}\right| \\ &= \frac{1}{2}\|X - Y\|_F^2 + \lambda \sum_{n=1}^{N-1}\left|X_{*,n+1} - X_{*n}\right| \end{aligned}$$

However, such a straightforward extension may fail to compress Y optimally. The reason is that, the optimal solution X may be more smooth with some projections of the original multivariates. Therefore, to facilitate the optimal lasso compress with multivariate time series, we propose the objective function as follows:

$$J(X, V) = \frac{1}{2}\|YV - X\|_F^2 + \lambda \sum_{n=1}^{N-1}\left|X_{*,n+1} - X_{*n}\right| \tag{4}$$

where V is an orthogonal matrix such that $V'V = I$ where I is the identify matrix. The optimal V minimizing the function $J(X, V)$ can be used to construct the optimal projection of the original multivariates where the best energy efficiency can be realized. In comparison with Equation 2, the new problem formulation in Equation 4 can better unify the compression of the multivariate sensor readings by finding the smooth approximation in the transformed space.

3.3. Learning Algorithm

We use alternative algorithm to solve X and V iteratively. Particularly, we initialize the algorithm with $V = I$, and then update X with V fixed and then update V with X fixed. Such iteration is repeated until the objection function $J(X,V)$ converges or the maximal number of iterations is reached.

More specifically, when updating X, the problem is equivalent with the multiple univariate problems, where each dimension of X can be computed independently. When updating V, the problem can be simplified to

$$\begin{aligned} &\min_V \frac{1}{2}\|YV - X\|_F^2 \\ &\text{subject to: } V'V = I \end{aligned} \tag{5}$$

To solve this problem, we use the updating procedure proposed in [26]. Specifically, with the current solution V, we compute its gradient $G = Y'(YV - X)$. Then, we define a skew-symmetric matrix $S = GV' - VG'$, and then compute the new solution:

$$\begin{aligned} V &\leftarrow \left(I + \frac{\tau}{2}S\right)^{-1}(I - \frac{\tau}{2}S)V \\ V &\leftarrow (I + \frac{\tau}{2}S)^{-1}(I - \frac{\tau}{2}S)V, \end{aligned} \tag{6}$$

where $\tau \geq 0$ is the learning rate. It can be shown that this updating procedure can decrease the objective function $\frac{1}{2}\|YV - X\|_F^2$ while satisfy the constraints $V'V = I$.

4. Experimental results

We evaluate the performance of our proposed algorithm with univariate and multivariate real-world data sets collected from sensor nodes deployed in WSNs.

4.1. Experimental Data

The first univariate data set is published by SensorScope HES-SO FishNet Deployment [20], which consists of temperature readings (FN_T) and relative humidity readings (FN_H). The range of the value in FN_T is relatively small compared to FN_H, and both of the time series of FN_T and FN_H are smooth. Another univariate data set is Microphone data (SC_M), which is collected by deploying sensor nodes at Strata Clara convention center[1]. SC_M contains lots of peaks and noises, and can be considered as discontinuous (non-smooth) signals.

Besides, we use several multivariate data sets. The first multivariate data set is collected by the Mobile Health (MHealth) program which comes from the UCI machine learning repository [12]. The MHealth data set deals with human behavior analysis based on multi-modal body sensing, which comprises body motion and vital sign recordings

[1]http://datasensinglab.com/data/

collected by ten volunteers while performing several physical activities. Sensors placed on the volunteer's chest, right wrist and left ankle to measure the motion experienced by diverse body parts, namely, acceleration, rate of turn and magnetic field orientation. From the MHealth program, we selected two-lead ECG (MH_ECG), three-axis acceleration of right wrist (MH_AR), three-axis acceleration of left ankle (MH_AL), and three-axis magnet data (MH_MG). The second multivariate data set is published by CRAWDAD[2]. This data set is mainly three-axis acceleration readings (CJ_A), which is collected through different drivers' mobile phones to record the motions of their vehicles. The third multivariate data set (CM_A) is collected to monitor cows' behavior, which aims at detecting whether these cows are in the estrus [27]. CM_A is generated from three-dimensional accelerator. **Table 1** illustrates the detailed information about the data sets utilized in our experiments. In WSNs, the statistical characteristics of measured data can affect the performance of compression algorithms. Therefore, to validate our proposed compression algorithm, we select these data sets generated in different scenarios. We can see from **Table 1** in which the data sets we employed in our experiments are quite different in terms of statistical characteristics.

Table 1. Basic information of the data sets.

Data	#Samples	Sampling interval	#Dimensions	Time
FN_T	14721	2 minute	1	2007/08/06 - 2007/09/02
FN_H	14721	2 minute	1	2007/08/06 - 2007/09/02
SC_M	2887	4-9 second	1	2013/2/27 - 2013/3/1
CJ_A	16060	0.0625 second	3	2012-11-03
CM_A	2073600	0.1 second	3	2011/12/8 - 2011/12/31
MH_ECG	483840	0.02 second	2	2014/12/07
MH_AR	483840	0.02 second	3	2014/12/07
MH_AL	483840	0.02 second	3	2014/12/07
MH_MG	483840	0.02 second	3	2014/12/07

4.2. Evaluation Metrics

4.2.1. Compression Ratio (CR)

The data compression ratio is often defined as the ratio of the compressed size to the original size, or the ratio of the saving size relative to the uncompressed size. Here we define the compression ratio as follows,

$$CR = 100 \times \left(1 - \frac{comp\_size}{orig\_size}\right) \tag{7}$$

where the $comp\_size$ and $orig\_size$ represent the size of compressed data and the size of original sensor data respectively.

[2]http://www.crawdad.org/jiit/accelerometer/20121103/

4.2.2. Approximation Mean Error (AE)

We compute the approximation error of each compression method with the following equation:

$$AE = \frac{1}{N}\|YV - X\|_F^2 \tag{8}$$

where Y is the original multivariate data, and X is the compressed version. For the univariate compression algorithms (such as LTC, PLAMlis, and univariate LASSO), the transformation matrix $V = I$ is the identify matrix. For our multivariate compression algorithm, the optimal V is computed using Equation 4.

4.2.3. Energy Consumption (EC)

The total energy consumption of sensor node consists of two parts: energy consumption for compression and energy consumption for transmission. For compression, we take into account the number of operations processed by CPU, without considering the additional cost generated by other peripherals of micro-controller. For the energy consumption generated by transmission, we only consider the cost of transmitting and receiving data, which is often the main cost of data transmission. Subsequently, we have the following metric for energy consumption,

$$E_{total} = M \times E_{opt} + N \times E_{bit} \tag{9}$$

Where E_{bit} is the energy consumed to transmit a data bit, E_{opt} is the cost of one CPU operation, and M is the number of operations needed for accomplishing the compression task.

In our experiments, we estimate the energy consumption on the sensors equipped with Chipcon MSP430 MCU[3] and Chipcon CC2500[4] radio transceiver. The MSP430 is powered by a current of $433.86\mu A$ at 3.0V, and it has a clock rate of 5.33MHz. Thus, the energy consumption of MSP430 is 0.244nJ/clock. For CC2500, the current associated with the transmission activity is 21.2mA with a supply voltage of 3.0v at an effective data rate of 250kbps. Therefore, while transmitting data, the energy cost by CC2500 is 254.4nJ/bit.

4.2.4. Compression Time (C Time)

For a wireless sensor node, to achieve better compression performance, i.e. higher compression ratio, the computation cost to operate such compression must be higher. Therefore, we use compression time to quantify the tradeoff between computational task and communication task. With the compression time, we can find the better compression strategies for saving energy consumption. Here, we define the compression time as the mean computation time of compressing one data bit.

[3]http://www.ti.com/lit/ds/symlink/msp430fr5739.pdf
[4]http://www.ti.com.cn/cn/lit/ds/symlink/cc2500.pdf

4.3. Baseline Algorithms

To show the effectiveness of our algorithm, we compare our algorithm against the following state-of-the-art methods:

- Lightweight Temporal Compression (LTC), is a low-complexity piecewise linear approximations lossy compression algorithm. It fits the consecutive measurements as a straight line within the desired error margin [16]. The greater compression ratio is obtained, while the larger error bound is given.
- Piecewise Linear Approximation with Minimum number of Line Segments (PLAMLiS), takes an n-length sequence of measurements and finds the minimum number of line segments required to represent the sequence within an error bound [16].

LTC has a complexity of $O(n)$, and the optimal solution of PLAMLiS requires the time complexity of $O(n^2 \log n)$, where n is the number of data items.

4.4. Experimental Settings

Before demonstrating the evaluation results, we need to present some experimental settings first.

4.4.1. The degree of regularization λ

In Section 3, we present the proposed algorithms, i.e. ULasso and MLasso. Both of Ulasso and MLasso are determined by the degree of regularization λ, thus, we need to determine the value (or range) of λ for our evaluation.

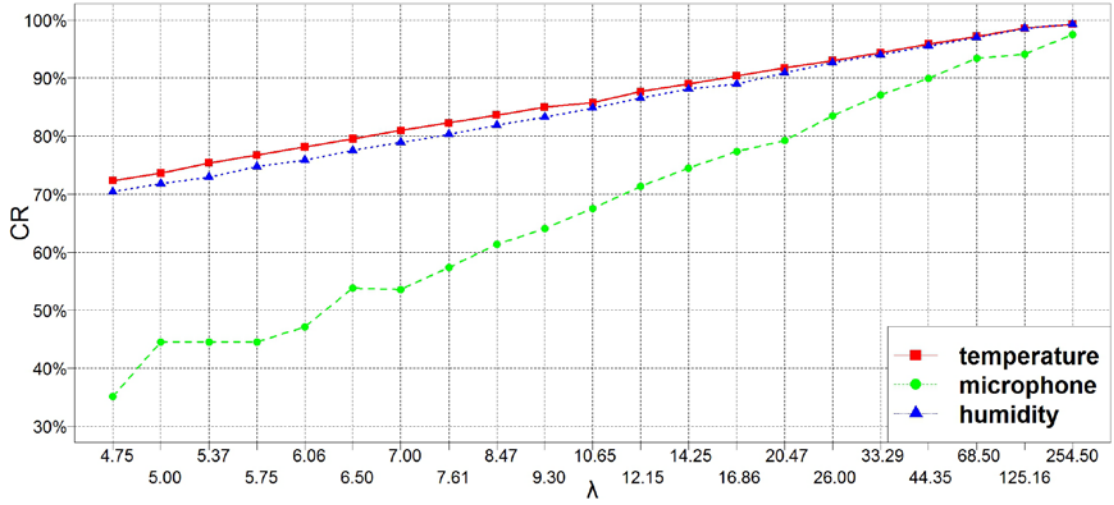


Fig. 1. Compression ratio coincide with the value of λ.

Unlike the other lossy compression algorithm using the error bound, lasso penalizes their successive differences with the parameter λ, that ULasso and MLaso compression

algorithms give the flexibility of compression ratio by specifying the different value of λ. **Fig. 1** shows the compression benefit at different value of λ for ULasso. The results in **Fig. 1** reveal that smooth signals like temperature and humidity can get the higher compression ratio applying ULasso. The smaller λ corresponds to the lower compression ratio, in other words, ULasso (with small λ) emphasizes the minimization of difference between the actual reading and the approximation. Moreover, the compression ratio climbs up slowly and flatly when increasing λ. **Fig. 2** depicts the approximation of the temperature at different value of λ. It can be seen that, bigger λ leads to greater difference between the approximated value and the original value.

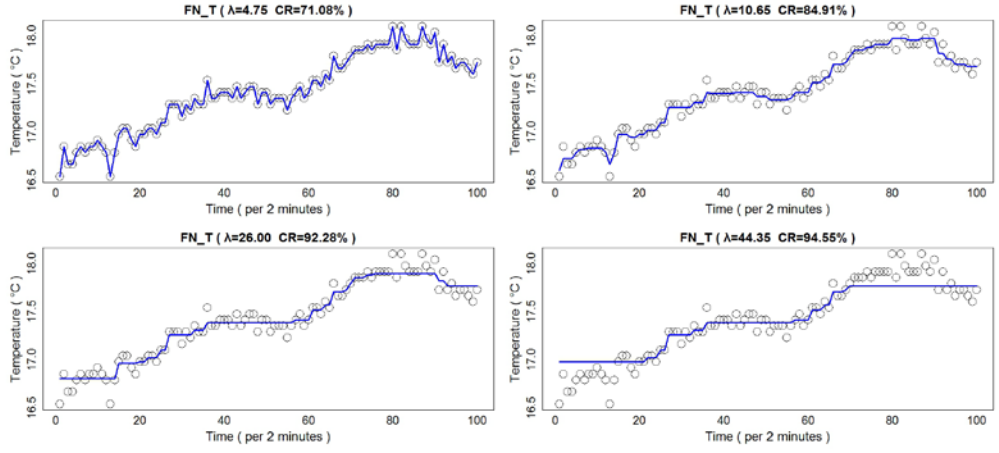


Fig. 2. The performance of temperature compressed data at different value of λ.

Fig. 3 shows the picture of Approximation Mean Error (AE) against λ. The value of AE increases with the grown up of λ, but the gradients on different data sets vary.

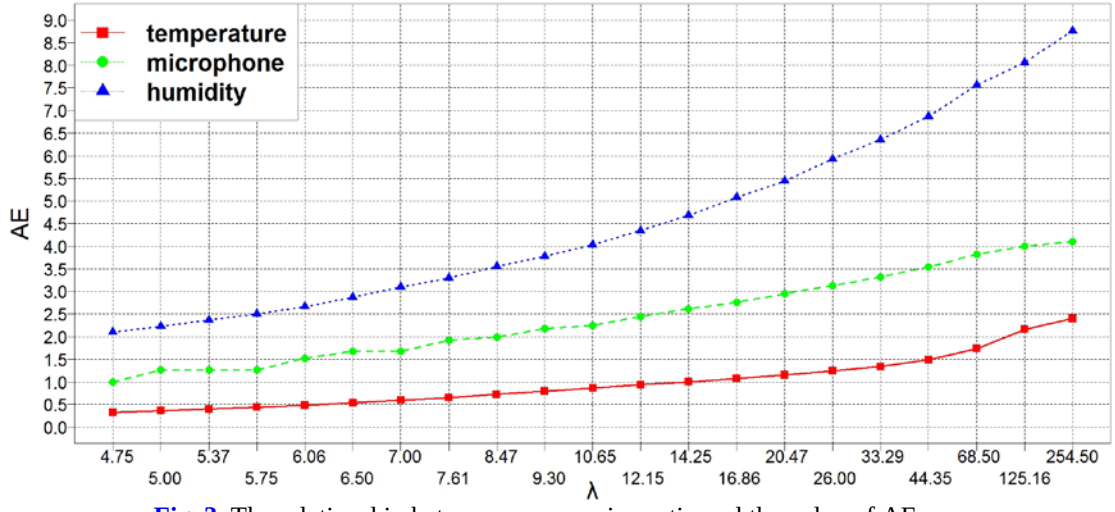


Fig. 3. The relationship between compression ratio and the value of AE.

For the smooth data, the AE of humidity grows much more quickly than temperature. The reason is that, the variance of humidity time series is most high in the univariate data sets (see **Table 2**). The non-smooth data, like the microphone, shows the medium increasing lope as λ climbs up, since its variance is not as much high as humidity though it is non-smooth data.

Table 2. Statistical characteristics of the experimental data sets.

Data set	Axis	Std.	CV	Mean	Q1	Q3
FN_T		2.64	0.17	15.31	13.48	17.16
FN_H		9.88	0.12	84.20	77.39	91.68
SC_M		4.21	0.78	5.38	2.00	7.00
CM_A	x	2.71	-0.08	-32.31	-34.00	-32.00
	y	8.90	9.01	0.99	-5.00	8.00
	z	5.02	1.42	4.00	1.00	6.00
CJ_A	x	1.19	0.23	5.25	4.86	5.74
	y	1.01	0.79	1.28	0.73	1.84
	z	0.78	0.10	7.98	7.62	8.35
MH_ECG	1	0.50	-63.36	-0.01	-0.19	0.15
	2	0.64	-18.27	-0.04	-0.21	0.16
MH_AR	x	4.32	-1.03	-4.18	-6.27	-1.82
	y	6.29	-1.29	-4.87	-9.27	-0.06
	z	3.62	1.66	2.18	0.17	4.39
MH_AL	x	4.67	3.80	1.23	0.10	2.64
	y	4.06	-0.42	-9.67	-10.07	-9.03
	z	5.03	-3.05	-1.65	-3.29	0.65
MH_MG	x	48.90	109.77	0.45	-11.27	3.64
	y	52.40	-54.71	-0.96	-8.95	5.18
	z	36.95	-35.15	-1.05	-2.07	5.70

So we select the λ by not only considering the compression ratio, but also the value of AE. The smaller λ, i.e. the lower compression ratio and lower AE, let the output fit well to the original data. The larger AE results in bigger differences between the original reading and approximation value, eventually the approximation may totally distort the original data. Indeed, the λ is a key parameter on trading off the compression output performance and the compression ratio.

4.4.2. The compression batch size λ

The constraint of using the lasso for compression is the computational cost, especially when the data scale is large. Here we explore the dependence of the compression ratio and computation complexity against the scale of compressed data, and discuss how to choose batch size N in one-time compression for reducing the computational complexity of algorithm, so it can be executed efficiently on a sensor node. Before giving the description of results, we define the original data length N for one-time compression

procedure as the batch size. From the lasso mathematical models, batch size N is a key parameter to determine the computational complexity. When new sampling point is coming and added to the lasso model, the iterative procedure should be updated to minimize the residual sum of squares to get the optimal solution, therefore the cost of computation procedure of Lasso models grows as the number of sampling data grows up. Increasing the length of batch size will increase the scale of model operation as well as the memory capacity required. All these will lead to more computation time, hence, much higher energy consumption. So the underlying idea of reducing the computation complexity of lasso model is to select a reasonable batch size of sampling data, and the compression procedure execute at the batch size scale. **Fig. 4** shows the temperature data compression ratio corresponding to the given the batch size N. The result reveals that, when increasing N, the compression ratio varies among all these methods. When the λ is small, the ULasso compression ratio fluctuate greatly as the batch size increases. Particularly, at first, the compression ratio changes up and down many times at the downward trend as N increases, and finally, compression ratio will keep at the constant value. While λ is large, the ULasso compression ratio almost keep the stable value which unaffected by N values. That is to say, for the small λ, we should carefully determine the batch size to obtain higher compression ratio and reduce computation time. The appropriate batch size N is also a key parameter for the compression performance of the ULasso and MLasso.

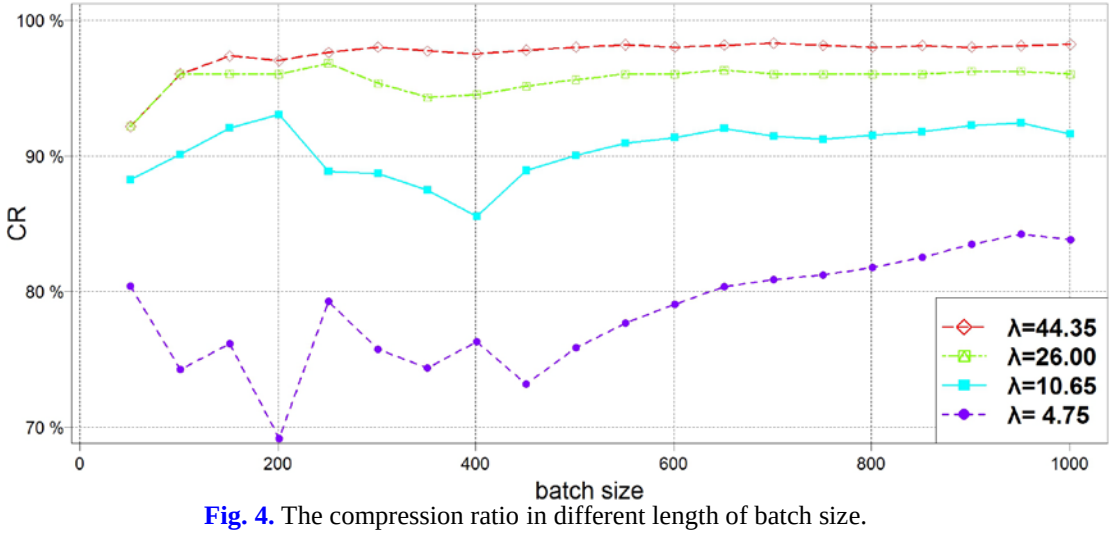


Fig. 4. The compression ratio in different length of batch size.

4.5. Evaluation on Proposed Algorithms

For the lossy compression algorithms, e.g. LTC and PLAMlis, the error bound is the key factor impacting compression performance. The error bound is defined as the maximum acceptable difference between each individual raw reading by the sensor and

the recovered one after receiving the compressed representation [1]. The error bound can be conveniently specified as the proportion of sensor manufactured error (SME), therefore, we conduct experiments on lossy compression algorithms with specific SME (which is subjected to the hardware of sensors) instead of error bound.
In the following, we report our evaluation results on our method compared to the baseline algorithms.

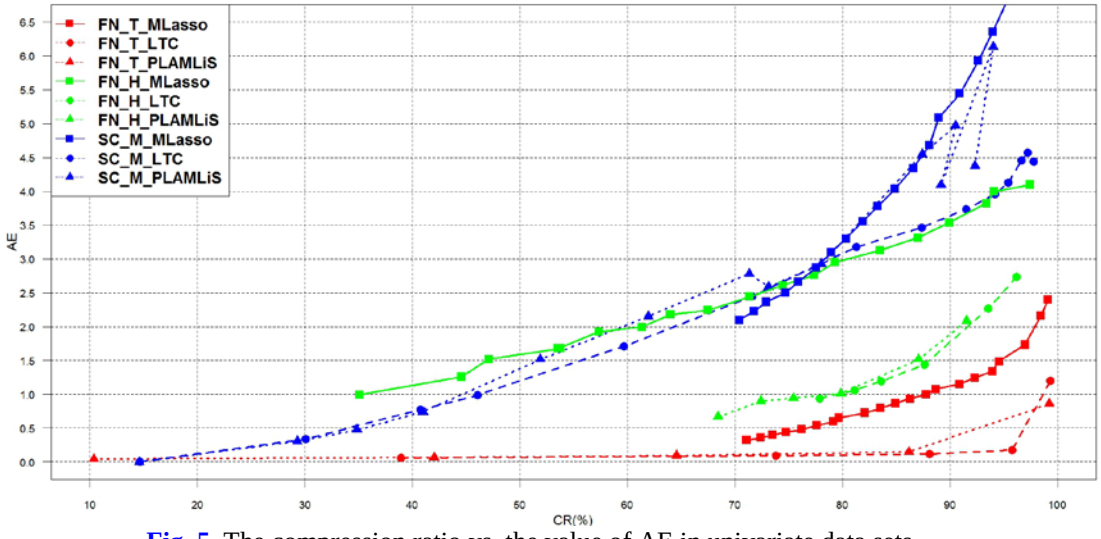


Fig. 5. The compression ratio vs. the value of AE in univariate data sets.

4.5.1. Compression Performance

(1) Univariate data sets.

As observed in **Fig. 5**, compared with LTC and PLAMlis, ULasso has the almost same compression ratio, but also has the high AE which performed poorly especially on smooth sensor data like temperature and humidity. LTC and PLAMlis represent time series with the piecewise linear, which constrains the endpoints of each segment to coincide with intermediate points. ULasso uses piecewise constant to represent each segment. For the continuous univariate time series signal with small changing, the ULasso algorithm performed poorly compared to LTC and PLAMlis in terms of compression ratio or energy consumption. However for the multivariate data, each variation contains respe ctively amplitude fluctuations and transitions. Have the LTC and PLAMlis the same advantage in multidimensional data?

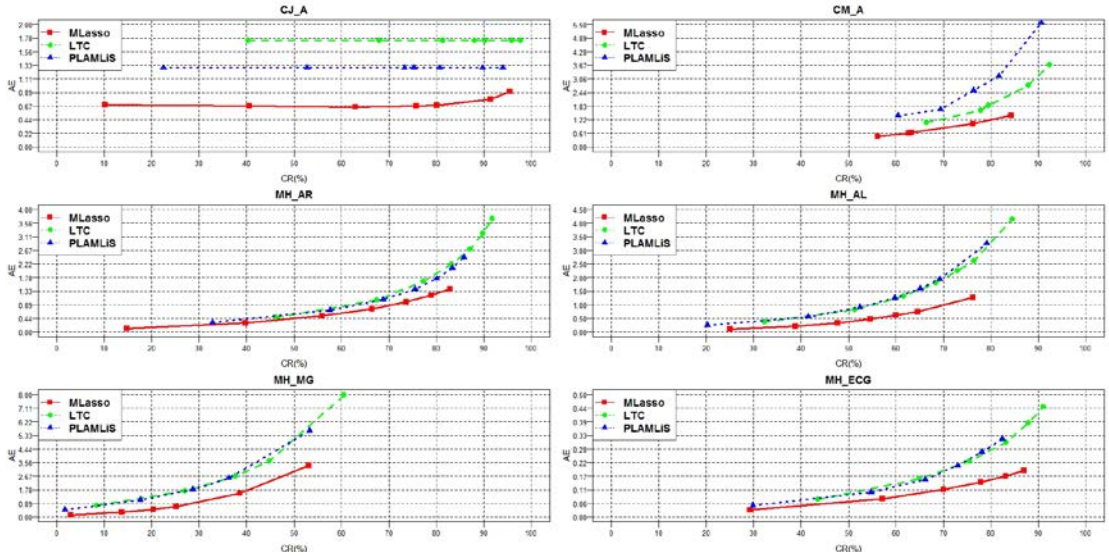


Fig. 6. The compression ratio and the value of AE in multivariate data sets.

(2) Multivariate data sets.

To handle the multivariate data sets, LTC and PLAMlis algorithms need to divide the multivariate or multi-dimension original data into several single variation or one-dimensional arrays first, then compress them separately, and synthesize the results. While, MLasso algorithm is proposed to operate the multivariate sensor data as the whole to find the closest adjacent points in the multidimensional sequence. **Fig. 6** plots the approximation mean error and compression ratio of LTC, PLAMlis and MLasso algorithms based on six real-world sensor data sets. For LTC and PLAMlis in **Fig. 6**, the approximation mean error values compared with compression ratio increasing sharply from left to right. However the approximation mean error kept the smooth and slower increasing with the compression ratio using the MLasso. Moreover as shown, MLasso provides the lower approximation mean error at the same compression ratio compared with LTC and PLAMlis. For example, MLass with CR = 60.0% resulted in approximately mean error is 0.62, while LTC with CR = 61.6% and PLAMlis with CR = 59.8% resulted in approximation mean error respectively is 1.31 and 1.25 on the MH AL data set. The approximation mean error results present that MLasso compression technique performs better than LTC and PLAMlis in multivariate data set. From the results, Mlasso may not get the highest CR, but at the same CRMLasso has the lowest AE. In other words, the higher CR, the difference of AE between these algorithms is much larger. The reason is that these multivariate data sets, such as ECG signal which contains two variations of electrical activity of the heart, and acceleration which consists of three-axis movement data, are typically smooth data with higher temporal correlations. For these smooth data, LTC and PLAMlis can get higher compression ratio from each single variation, but each variation has different amplitude fluctuations that the synthesized results will bring more considerable error that produce larger approximation mean error. MLasso compress

multivariate data as the whole proves to be the best solution for different coordinate systems, which get the best compression performance, but it need more computational time. Experimental results give us conclusion that MLasso algorithm is particularly effective on the multivariate smooth data set, and obtain better compression performance than LTC and PLAMlis. And all of LTC, PLAMlis and MLasso perform poorly when applied to non-smooth data set. About the energy consumption analysis we will discuss in the next subsection.

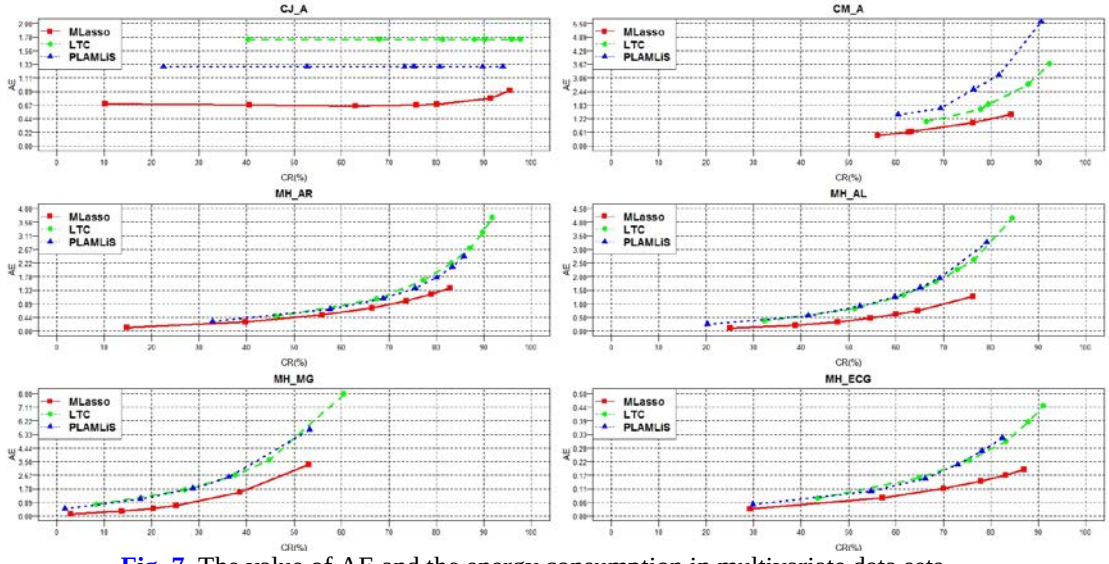


Fig. 7. The value of AE and the energy consumption in multivariate data sets.

4.5.2. Energy consumption

One purpose of performing compression on sensor node is energy saving, while running the compression algorithm could consume additional computation energy. In our experiments we assess the computation time and total energy consumption of each compression algorithms on various data sets. We only consider the energy consumption on transmission and computation which are the main cost of energy consumption on sensors. The computation consumption is estimated by computation time calculated by the actual system running time. And the transmission consumption uses theoretical calculation results. Lasso uses linear regression techniques to calculate a constant line segment fitting the original data with the minimum mean squared error. We record the computation time that is running on the selected batch size to reduce the computation complexity.

Fig. 7 gives the energy consumption and AE comparison results handled by three compression algorithms on multivariate data sets. For the LTC and PLAMlis, the compression technique is the same as the univariate scenario just one by one compress every variation. So the time computation for every bit is almost the same as univariate.

However, to obtain the optimal value, MLasso has to do much time multi-dimension iteration operation, which will result in higher computation complexity. From these six multivariate data sets, MLasso spends several ten times computation time compared to LTC and PLAMlis, but **Fig. 7** shows at the same AE, MLasso uses the least energy consumption. There is for the energy consumption on transmitting data by wireless sensor node is much higher than computation (transmitting one bit data will consume at least 1042 times more power than compressing one bit data). Though MLasso cost more computation time, the higher compression ratio can tradeoff the high expenditure computational energy consumption to obtain the most saving of total energy consumption than other compression algorithms. For example, to transmit an original data bit will cost 254.5nJ, after performing MLasso compression algorithm, the CJ A decline to 26.26 nJ/bit, thus the total energy saving is 89.7%. All the five smooth multivariate data sets save total energy range from 96.39% to 51.27%. Even the non-smooth MH MG data can save 11.6% total energy. In summary, MLasso algorithm achieves good balance among accuracy, compression ratio, and energy consumption competitive against LTC and PLAMlis.

4.6. Case Study: Accuracy of Target Detection (AD)

Table 3. The accuracy of detection cow's estrus

Method	Accuracy (%)	CR (%)	Save energy (%)
No compression	93.75	0	0
LTC	87.50	76.06	76.00
PLAMlis	90.63	64.03	63.83
MLasso	89.06	91.50	90.87

The purpose of real-world application with employing WSNs can be categorized into two groups: tracking and monitoring [22]. For example, the ZebraNet system is tracking, which is to record the position data in order to track long term animal migrations [28]. For the monitoring instance, collecting a cow's moving accelerate data is to monitor the behavior of the cow, and to further determine its health or estrus states.

There are also some other applications performing target detection. The sensor data collected from such scenario are based on high frequency sampling and multi-dimension spatio-temporal correlation, which is consequently hard to evaluate the quality of reconstructed compressed data. Approximation mean error can evaluate the distance between the original data and the recovered, However, in some specified applications, they mostly concern the magnitude and trend of continuous sensor data. Since the general shape and the trend of the physical phenomena evolving curve is the key factor. In this scenario, approximation mean error is not only means to evaluate compression performance. One of our test data set collects cows moving accelerator to classify various behavior, then detect cow estrus state. For this specified application scenario (target detection), we use the accuracy of target detection (after applying the compressed data) to

determine the performance of compression algorithm. Here, we compare the prediction accuracy using the compressed data, and the cow behavior detecting algorithm is proposed by this work [27]. **Table 3** shows the accuracy of predicting the estrus time of cow by recovered compressed data and original data.

The results shows that PLAMlis obtains the highest accuracy 90.63% in all compression algorithms closing the accuracy of original data 93.75%, but with lower compression ratio and larger total energy consumption. The MLasso gets the 89.06% accuracy, almost the same with PLAMlis. The compression ratio of MLasso is 91.50%, and the total energy consumption of one bit is 23.23 nJ, which can save 90.87% total energy. Actually, data are collected by accelerator sensors which, due to noise, produce different readings even when they are sampling an unchanging behavior. So, even if the sensor readings fluctuate frequently, the cow estrus detecting algorithm just needs catch the major trends and important changes in the sensor data.

5. Conclusion

In this paper, we proposed two compression algorithms, ULasso for univariate data and MLasso for multivariate data, which are purposely-designed for the wireless sensor nodes. Our method differs from existing compression techniques applied in WSN. We explored multivariate sensor data compression in different coordinate space by taking advantage of the structure and characteristic of sensor data to decrease the total energy consumption. Both ULasso and MLasso compression schemes exploit the high temporal correlation that typically exists in time series generated by sensors. To evaluate our method, we first investigated the performance of ULasso algorithm on three different real-world univariate sensor data, which consist of smooth and non-smooth signals. The experimental results show that ULasso obtains higher compression ratio and comparable energy consumption against baselines, but higher approximation error. Then, we assessed the MLasso algorithm by compressing six various multivariate data sets which are selected to represent smooth and non-smooth signals from various application domains. Experimental results demonstrate that MLasso achieves the higher compression ratio, lower approximation error and much more saving energy compared to the baselines. MLasso particularly suits to the high sampling rate, dense and smooth multivariate data. Although the computation complexity of MLasso is high, it can trade off compression efficiency and complexity so as to achieve considerable compression ratio and minimize the total power consumption.

References

[1] Mohammad Abu Alsheikh, Puay Kai Poh, Shaowei Lin, Hwee-Pink Tan, and Dusit Niyato. “Efficient data compression with error bound guarantee in wireless sensor networks,” in *Proc. of the 17th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems*, pages 307–311. ACM, 2014. Article (CrossRef Link)

[2] Amr Ahmed and Eric P Xing, "Recovering time-varying networks of dependencies in social and biological studies," in *Proc. of the National Academy of Sciences*, 106(29):11878–11883, 2009. Article (CrossRef Link)
[3] Alon Amar, Amir Leshem, and Michael Gastpar, "Recursive implementation of the distributed karhunen-loeve transform," *Signal Processing, IEEE Transactions on*, 58(10):5320–5330, 2010. Article (CrossRef Link)
[4] Kenneth C Barr and Krste Asanovic, "Energy-aware lossless data compression," *ACM Transactions on Computer Systems (TOCS)*, 24(3):250–291, 2006. Article (CrossRef Link)
[5] Donoho D L., "Compressed sensing[J]," *IEEE Transactions on information theory*, 52(4): 1289-1306, 2006. Article (CrossRef Link)
[6] Elena Fasolo, Michele Rossi, Jorg Widmer, and Michele Zorzi, "Innetwork aggregation techniques for wireless sensor networks: a survey," *Wireless Communications, IEEE*, 14(2):70–87, 2007. Article (CrossRef Link)
[7] LA Gonzalez, GJ Bishop-Hurley, RN Handcock, and C Crossman, "Behavioral classification of data from collars containing motion sensors in grazing cattle," *Computers and Electronics in Agriculture*, 110:91–102, 2015. Article (CrossRef Link)
[8] Naoto Kimura and Shahram Latifi, "A survey on data compression in wireless sensor networks," in *Proc. of Information Technology: Coding and Computing, ITCC 2005, International Conference on*, volume 2, pages 8–13. IEEE, 2005. Article (CrossRef Link)
[9] M Kozlovszky, L Kovacs, and K Karoczkai, "Cardiovascular and diabetes focused remote patient monitoring," in *Proc. of VI Latin American Congress on Biomedical Engineering CLAIB 2014*, Paran´a, Argentina 29, 30 & 31 October 2014, pages 568–571. Springer, 2015. Article (CrossRef Link)
[10] Hong-Nan Li, Ting-Hua Yi, Liang Ren, Dong-Sheng Li, and Lin-Shen Huo, "Reviews on innovations and applications in structural health monitoring for infrastructures," *Structural Monitoring and Maintenance*, 1(1): 1–45, 2014. Article (CrossRef Link)
[11] Junlin Li and Ghassan AlRegib, "Distributed estimation in energyconstrained wireless sensor networks," *Signal Processing, IEEE Transactions on*, 57(10):3746–3758, 2009. Article (CrossRef Link)
[12] M. Lichman, UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml. Article (CrossRef Link)
[13] Jun Liu, Lei Yuan, and Jieping Ye, "An efficient algorithm for a class of fused lasso problems," in *Proc. of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–332, ACM, 2010. Article (CrossRef Link)
[14] Jialiang Lu, Fabrice Valois, Mischa Dohler, and Min-You Wu. "Optimized data aggregation in wsns using adaptive arma," in *Proc. of Sensor Technologies and Applications (SENSORCOMM), 2010 Fourth International Conference on*, pages 115–120. IEEE, 2010. Article (CrossRef Link)
[15] Francesco Marcelloni and Massimo Vecchio, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *The Computer Journal*, 52(8):969–987, 2009. Article (CrossRef Link)
[16] Dennis Parker, Milica Stojanovic, and Chu Yu, "Exploiting temporal and spatial correlation in wireless sensor networks," in *Proc. of Signals, Systems and Computers, 2013 Asilomar Conference on*, pages 442–446. IEEE, 2013. Article (CrossRef Link)
[17] Fernando Perez-Cruz and Sanjeev R Kulkarni, "Robust and low complexity distributed kernel least squares learning in sensor networks," *Signal Processing Letters, IEEE*, 17(4):355–358, 2010. Article (CrossRef Link)

[18] Ngoc Duy Pham, Trong Duc Le, and Hyunseung Choo, "Enhance exploring temporal correlation for data collection in wsns," in *Proc. of Research, Innovation and Vision for the Future, 2008, RIVF 2008. IEEE International Conference on*, pages 204–208. IEEE, 2008. Article (CrossRef Link)

[19] Sharadh Ramaswamy, Kumar Viswanatha, Ankur Saxena, and Kenneth Rose, "Towards large scale distributed coding," in *Proc. of Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1326–1329, IEEE, 2010. Article (CrossRef Link)

[20] Thomas Schmid, Henri Dubois-Ferriere, and Martin Vetterli, "Sensorscope: Experiences with a wireless building monitoring sensor network.," in *Proc. of Workshop on Real-World Wireless Sensor Networks (REALWSN" 05)*, number LCAV-CONF-2005-015, 2005. Article (CrossRef Link)

[21] Tom Schoellhammer, Ben Greenstein, Eric Osterweil, Michael Wimbrow, and Deborah Estrin, "Lightweight temporal compression of microclimate datasets," *Center for Embedded Network Sensing*, 2004. Article (CrossRef Link)

[22] Tossaporn Srisooksai, Kamol Keamarungsi, Poonlap Lamsrichan, and Kiyomichi Araki. "Practical data compression in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, 35(1):37–59, 2012. Article (CrossRef Link)

[23] L Taheriazad, C Portillo-Quintero, and GA Sanchez-Azofeifa, "Application of wireless sensor networks (wsns) to oil sands environmental monitoring," *osrin report no. Technical report*, TR-48. 51 pp. http://hdl.handle.net/10402/era. 38858, 2014. Article (CrossRef Link)

[24] Robert Tibshirani and Pei Wang, "Spatial smoothing and hot spot detection for cgh data using the fused lasso," *Biostatistics*, 9(1):18–29, 2008. Article (CrossRef Link)

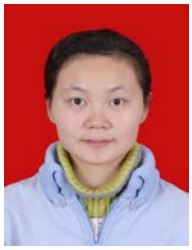
[25] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005. Article (CrossRef Link)

[26] Zaiwen Wen and Wotao Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, 142(1-2):397–434, 2013. Article (CrossRef Link)

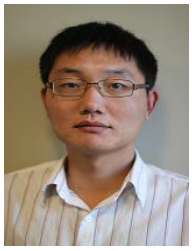
[27] Ling Yin, Tiansheng Hong, and Caixing Liu, "Estrus detection in dairy cows from acceleration data using self-learning classification models," *Journal of Computers*, 8(10):2590–2597, 2013. Article (CrossRef Link)

[28] Pei Zhang, Christopher M Sadler, Stephen A Lyon, and Margaret Martonosi, "Hardware design experiences in zebranet," in *Proc. of the 2nd international conference on Embedded networked sensor systems*, pages 227–238, ACM, 2004. Article (CrossRef Link)

[29] Davide Zordan, Borja Martinez, Ignasi Vilajosana, and Michele Rossi, "On the performance of lossy compression schemes for energy constrained sensor networking," *ACM Transactions on Sensor Networks (TOSN)*, 11 (1):15, 2014. Article (CrossRef Link)



Ling Yin received the Ph.D. degree in agricultural electrification and automation from South China Agriculture University, in 2011.She is currently an associate professor of College of Mathematics and Informatics with South China Agricultural University, Guangzhou, China. Her current research interests include wireless sensor network applications and agricultural information processing, with a focus on animal surveillance and agricultural data mining. She has published several papers in journals and conference proceedings.



Chuanren Liu received the Ph.D. degree from Rutgers, the State University of New Jersey, Newark, NJ, USA, now he is working in Decision Science and Management Information Systems Department, Drexel University, Philadelphia, PA. His current research interests include data mining and knowledge discovery, and their applications in business analytics. He has published several papers in refereed journals and conference proceedings, such as Knowledge and Information Systems, IEEE ICDM, SIAM SDM, and ACM SIGKDD.

Xinjiang Lu is currently a Ph.D. student in Computer Science at Northwestern Polytechnical University, Xi'an, China. He received the M.S. degree in Software Engineering from Norwestern Polytechnical University in 2011, and the B.E. degree in Computing Mathematics from Xinjiang University, Urumqi, China, 2007. His research interests include data mining and mobile intelligence.

Chen Jiafeng received B.S. and M.S. degrees in computer science and technology from South China Agricultural University, China in 2013 and 2016, respectively. His research interests include network application, performance control and medium access control protocols for wireless sensor networks.

Liu Caixing received the B.E. degree in computer science from Nanjing University, Nanjing, China. He is currently a Professor and the chair of College of Mathematics and Informatics and the chair of Software with South China Agricultural University, Guangzhou, China. He is also a senior member of China computer federation, vice chairman of Guangdong computer federation. His main research interests include embedded systems and wireless sensor networks, network and information security, big data analysis.