Energy Efficient Sensor Data Collection Approach for Industrial Process Monitoring

Hassan Harb and Abdallah Makhoul

***Abstract*—The use of wireless sensor network (WSN) for industrial applications has attracted much attention from both academic and industrial sectors. Sensors are typically deployed to gather data from the industrial environment and to transmit it periodically to the end user. In this paper, we propose and compare three data collection mechanisms that allow each sensor node to adjust its sampling rate to the variation of its environment, while at the same time optimizing its energy consumption. The first one uses the analysis of data variances via statistical tests to adapt the sampling rate, while the second one is based on the set similarity functions, and the third one on the distance functions. Both simulation and real experimentations on telosB motes have been conducted where the obtained results proved that our methods can reduce the number of acquired samples up to 80% with respect to a traditional fixed-rate technique.**

***Keywords*-Industrial wireless sensor networks, data collection, adaptive sampling rate, analysis of variance, similarity functions, distance functions, telosB mote.**

Parameter	Description
M_i	set of measures collected during p
$Similar(m_i, m_j)$	function used to test if two measures are similar
ε	threshold for $Similar$ function
$\mathcal{T}$	the number of total measures during p
$wgt(m_i)$	weight of the measure m_i
M'_i	set of measures with their associated weights
$\|\|M'_i\|\|$	cardinality of M'_i
$Card_w(M'_i)$	the weighted cardinality of M'_i
R	the application risk level
$Behavior$	function used to adapt sensor sampling rate
L	the number of periods in each round
r	the round
S_{max}	the sampling rate maximum
$J'(M'_i, M'_j)$	Jaccard Similarity after assigning measure weights
t_J	the Jaccard threshold
$wgt_{min}(m'_i, m'_j)$	the minimum of the weights of (m'_i) and (m'_j))
E_d	Euclidean distance between two sets of measures
M'_{ir}	measures in the remained part of set M'_i

TABLE I
NOTATION USED IN THE ARTICLE.

LIST OF ACRONYMS

WSN	Wireless Sensor Network
IWSN	Industrial Wireless Sensor Network
Anova	ANalysis Of Variance
EDRR	Efficient Data Redundancy Reduction
BP	Back-Propagation
DPCM	Differential Pulse Code Modulation
FFNN	Feed Forward Neural Network
PFF	Prefix-Frequency Filtering
CH	Cluster-Head

I. INTRODUCTION

INdustrial wireless sensor networks (IWSN) are becoming more prevalent in most industrial companies [1]. Their applications cover the problems of air pollution, temperature, humidity monitoring, structural condition monitoring, production performance monitoring and improvement, etc. For instance, continuous monitoring of pressures eliminates the need for daily visits to the wellhead to manually record gauge readings. Also, the temperature monitoring on a rotating drier to ensure that the proper temperature is reached and maintained during the drying process is another exciting application of IWSN [2].

Unfortunately, energy consumption remains the performance limiting factor and the biggest constraint for IWSNs. Currently, most industrial applications request battery life of about five years and WSN systems are not viable in applications that require relatively large amounts of power [3]. Thus, it is important to monitor carefully the amount of data collected and sent, while preserving the quality of service expected by the application.

Since industrial sensor readings are sent to the sink on a periodic basis, the dynamics of the monitored condition or process can slow down or speed up [4]. Therefore, in order to keep the network operating for long time, adaptive sampling approach to periodic data collection constitutes a fundamental mechanism for energy optimization and data reduction. In this paper, we propose three different adaptive sampling techniques aiming to optimize the volume of data transmitted over the network thus saving energy consumption. The first technique searches the dependence of conditional variance between the generated data sets based on the one-way Anova model and the Bartlett test to adjust the sampling rate; the second one uses the similarity functions, such as Jaccard function, to search the similarity between data sets; while the third approach calculates dissimilarities between sets based on distance functions, such as Euclidean and Cosine in order to define the environment dynamics changing. In order to evaluate the performance of our techniques, both simulation and real experimentations were conducted and discussed.

The remainder of this paper is organized as follows. Section II presents the related work on data collection in sensor networks. In Section III, an intra-node preprocessing phase is presented. Sections V, VI, and VII present our techniques of adaptive sampling rate based on the Anova model, the Jaccard similarity function, and the Euclidean distance, respectively. Section VIII exposes the simulation and experimental results. Finally, Section IX concludes the paper and gives directions

H. Harb and A. Makhoul are with FEMTO-ST Institute/CNRS, the DISC department, Univ. Bourgogne Franche-Comt, Belfort, France e-mail: hassan.moustafa_harb@univ-fcomte.fr and abdallah.makhoul@univ-fcomte.fr.

for future work.

II. Related Work

As mentioned above, data collection is one of the fundamental operations in WSNs. Therefore, researchers have proposed different adapting sampling techniques with the aim of saving the energy of the sensors and enhancing the network lifetime [5], [6].

Some works, such in [7]–[9], adapt the sampling rate of the sensors based on the correlation between sensed data. The authors in [7] propose an energy-efficient adaptive sampling mechanism which employs spatio-temporal correlation among sensor nodes and their readings. The main idea is to carefully select a dynamically changing subset of sensor nodes to sample and transmit their data. In [8], the authors propose an adaptive sampling approach based on the dependence of conditional variance on measurement variations over time, which allows sensor node to adapt its sampling rate to the physical changing dynamics. An Efficient Data Redundancy Reduction (EDRR) scheme is proposed in [9]. EDRR integrates conjugative sleep scheduler scheme and basically utilizes Differential Pulse Code Modulation (DPCM) technique to reduce data redundancy over the network.

Other works such as [10]–[12] reduce data collected by the sensors using data compression techniques. In [10] the authors propose a Sequential Lossless Entropy Compression (S-LEC) which organizes the alphabet of integer residues obtained from differential predictor into increased size groups. S-LEC codeword consists of two parts: the entropy code specifying the group and the binary code representing the index in the group. Compared to other compression schemes, S-LEC is characterized by its efficiency and highly robustness for various sensor network data sets. The authors in [11]–[12] join data compression and encryption in order to keep secure data after compressed them. First, in [11], they used a fuzzy approximation technique called F-transform. They compared a F-transform based approach to the to a DWT (discrete wavelet transform) based model and showed that they can achieve high enough value of the compression rate with a lower distortion. Later, in [12], the authors complete their proposition by studying a cubic B-spline F-transform in order to have a higher accuracy with low computational cost, even when data are not correlated. They show also that their approach is also suitable for data security, by integrating it with an encryption algorithm.

In [13]–[15], adapting the sensors sampling rate was studied based on the computation of statistical means and moments as per the end user and application requirements. The authors in [13] propose an adaptive sampling algorithm based on the Kalman filter for air pollution monitoring sensor networks. The objective of the proposed algorithm is to eliminate the noise from the sensor measurements and adjust the sampling interval based on the difference between present and previous measurements. In [15], the authors propose an adaptive sampling algorithm, called AdaSense, dedicated to wireless body sensor network. Through a genetic programming algorithm, AdaSense is able to determine the optimal sensor sampling rates by reducing the acquisition rate required in activity event detection and multi-activity classification.

Other works, such in [5], [16], [17], try to eliminate data redundancy intra and inter nodes. The authors in [16] propose a two-level sensor fusion-based event detection technique for the WSN. In the first level, each sensor node is responsible for deciding whether an event has been occurred, using a feed forward neural network (FFNN) or Naíve Bayes classifier. In the second level, at cluster-head or gateway, a fusion algorithm is proposed to reach a consensus among individual detection decisions made by sensor nodes. Recently, the authors in [5] study new area within filtering data generated by sensors, the Prefix-Frequency Filtering (PFF) technique. Further to a local processing at the sensor node level, PFF uses Jaccard similarity function at the aggregators level to identify similarities between near sensor nodes and integrate their sensed data into one record.

Although the techniques proposed in the literature have successfully adapted the sampling rate, the most of them are performed in a centralized way [5], [7] and are based on organizing sensors into clusters [16], [17] that require huge computations and communications. Indeed, few efforts in distributed sampling algorithms [9], [13] are provided. However, most of them are applied at the physical layer and restricted by the type of the deployed sensors. This makes such algorithms not suitable and applicable for a huge number of WSN applications. In this paper specifically addressing industrial periodic sensor networks, we propose and compare three different methods for samling rate adaptation. They are applied in a distributed manner, less complex and suitable for limited resource sensor nodes. The proposed techniques are based on the variance, similarity, and distance study, respectively. Their aim is to reduce the data acquisition on sensors by adapting their sensing rates to the varying nature of the sensed data. Finally, simulations and real sensor network experimentations have been realized to show the effectiveness of the proposed methods and the results were discussed subsequently.

III. Intra-Node Pre-processing

In periodic applications like industrial applications, a period p is divided into time slots. In each slot s, each sensor S_i captures a new measure m_i, and forms a vector of measures during the period p as follows: $M_i = [m_1, m_2, \ldots, m_{\tau-1}, m_\tau]$ where τ is the number of total measures captured during the period p. Usually, sensor nodes take the same (or very similar) measures several times especially when s is too short or when the monitored condition varies slowly. Therefore, we define the $Similar$ function which allows each sensor node to eliminate redundant collected measures from the vector M_i.

Definition 1 ($Similar$ function): We define the $Similar$ function between two measurements m_i and m_j captured by the same sensor node S_i as:

$$Similar(m_i, m_j) = \begin{cases} 1 & \text{if} \quad |m_i - m_j| \leq \varepsilon, \\ 0 & \text{otherwise.} \end{cases}$$

where ε is a threshold fixed by the application and $|m_i - m_j|$ is computed based on the Equation (1):

$$|m_i - m_j| = \begin{cases} m_i - m_j & \text{if} \quad m_i \geq m_j, \\ m_j - m_i & \text{otherwise.} \end{cases} \quad (1)$$

Based on the above definition, two measures captured by a sensor are similar if and only if the $Similar$ function is equal to 1.

Then, we define the weight of a measure as follows:

Definition 2 (Weight of a measure m_i, $wgt(m_i)$): the weight of a measure m_i is defined as the number of the subsequent occurrence of the same or similar measurements (according to the $Similar$ function) in the same vector.

Subsequently, we describe how the sensor node searches the similarities between measures captured at the same period. In the first slot at the period, the sensor node S_i takes the first measure, initializes its weight to 1 and adds it to the final set which will be sent to the sink. Then, for each new captured measurement m_k, S_i searches for similarities in previous taken measurements in the same period. If a similar measurement is found, it deletes the new one and increments the corresponding weight by 1, else it adds the new measure to the set and initializes its weight to 1.

At the end of each period, S_i will transform the initial vector of measures, M_i, to a set of measures, M'_i, associated to their corresponding weights as follows: M'_i=$\{(m'_1, wgt(m'_1)), (m'_2, wgt(m'_2)), \ldots, (m'_k, wgt(m'_k))\}$, where $k \leq \tau$.

Based on the new set M'_i, we provide the two following definitions:

Definition 3 (Cardinality of M'_i): the cardinality of the set M'_i, represented by $||M'_i||$, is the number of measures in M'_i without their corresponding weights.

Definition 4 (Weighted Cardinality of the set M'_i, $Card_w(M'_i)$): the weighted cardinality of the set M'_i is equal to the sum of all weights of the measures in M'_i as follows:

$Card_w(M'_i) = \sum_{k=1}^{||M'_i||} wgt(m'_k)$,

where $m'_k \in M'_i$.

IV. Adaptation to Application Criticality

Since the applications have different criticality level, we define the risk level of an application by R. This risk can take values between 0 and 1 representing the lowest and the highest criticality levels, respectively. This criticality level is represented by a mathematical function $y = f_R(x)$ called $Behavior$ function.

Then, in order to model the $Behavior$ function, the Bezier curve is used which is flexible and can plot easily a wide range of geometric curves. Therefore, the $Behavior$ function curve can be drawn, using the Bezier curve, through three points $P_0(0,0)$ (original point), $P_1(b_x, b_y)$ (behavior point), and $P_2(h_x, h_y)$ (threshold point). Thus, when varying R between 0 and 1, P_1 will update its position based on the following function [4]:

$$Cr(R) = \begin{cases} b_x = -h_x.R + h_x, \\ b_y = h_y.R. \end{cases}$$

Subsequently, the $Behavior$ function is defined based on the Bezier curve as follows:

$$Behavior(X, h_x, R, h_y) =$$

$$\begin{cases} \frac{(h_y - 2b_y)}{4b_x^2} X^2 + \frac{b_y}{b_x} X & if\ (h_x - 2b_x = 0), \\ (h_y - 2b_y)(\propto (X))^2 + 2b_y \propto (X), & if\ (h_x - 2b_x \neq 0), \end{cases}$$

where

$$\propto (X) = \frac{-b_x + \sqrt{{b_x}^2 - 2b_x.X + h_x.X}}{h_x - 2b_x} \ \wedge \begin{cases} 0 \leq b_x \leq h_x, \\ 0 \leq X \leq h_x, \\ h_x > 0, \end{cases}$$

and X represents a given value on the x-axis. It changes in function of the technique selected for the adaptive sampling (see next sections).

V. Adapting Sampling Rate Using Anova Model and Bartlett Test

Adapting the sampling rate of the sensor node according to the dynamics of the monitored condition is an important task in WSN that can prevent collecting redundant measures and save energy. Therefore, studying the variance, or analysis of variance (Anova), between measures collected by a sensor node in several periods is useful to adapt the sampling rate of the sensor. Anova is a statistical model that is used to find out if the means, thus the variance, of data sets are significantly different or if they are relatively the same. The Anova computes a T-statistic value which is the ratio of the variance calculated based on the collected measurements. T can be calculated according to the appropriate statistical test. The sets are considered duplicated if the calculated T is less than the critical value of the T-distribution (or T_α) for some desired false-rejection probability (risk α).

In our previous work [4], we used the one-way Anova model to identify the variance between measures with three different tests: Fisher, Tukey, and Bartlett. Based on the obtained results, we concluded that Bartlett is the best test in terms of adapting sampling rate of the sensor and maximizing its lifetime. Therefore, in this paper, the results of Bartlett test is compared to those in other approaches.

A. Bartlett Test

The Bartlett test [18] is used to check if two or multiple data sets are from populations with equal variances. Equal variances across data sets is called homogeneity of variances. Thus, the Bartlett test is used to test the null hypothesis that variances of all data sets are equal against the alternative hypothesis that at least two are different. Therefore, if there is a round, r, of L periods with size n_l and variance σ_l^2 for each period then Bartlett test is applied as follows [4]:

$$T = \frac{(N - L)\ln(\sigma_p^2) - \sum_{l=1}^{L}(n_l - 1)\ln(\sigma_l^2)}{\lambda}, \quad (2)$$

where:

$$N = \sum_{l=1}^{L} n_l, \quad \lambda = 1 + \frac{1}{3(L-1)}\left(\sum_{l=1}^{L}\left(\frac{1}{n_l - 1}\right) - \frac{1}{N - L}\right),$$

and T is the Bartlett test condition. Furthermore, the pooled variance, e.g. σ_p^2, is defined in Equation (3):

$$\sigma_p^2 = \frac{1}{N-L}\sum_{l=1}^{L}\sigma_L^2. \quad (3)$$

Thus, the decision is based on the following:

- if $T > T_{L-1,\alpha}$ the variance between periods is significant with a false-rejection probability α.
- if $T \leq T_{L-1,\alpha}$ the variance between periods is not significant thus the measures captured in the L periods are considered correlated.

Note that $T_{L-1,\alpha}$ is a threshold which can be searched in the *chi-square* table based on L and α values.

Adapting to Anova model and Bartlett test, $Behavior$ function takes, based on Bezier curve, four variables as input: the variance measures T (replaces X), the threshold $T_{L-1,\alpha}$ (replaces h_x), the risk level R, and the original sampling rate at the time of network deployment S_{max} (replaces h_y). Then, it returns the instantaneous sampling rate, S_t, calculated after each round.

Algorithm 2 describes the adaptive sampling rate algorithm at the sensor node based on the variance study (Anova model and Bartlett test). For each round, every node decides to increase or decrease its sampling rate according to the variance condition and the application risk. As long as the energy is positive, each node calculates the parameters T and $T_{L-1,\alpha}$ then it uses the $Behavior$ function in order to find its new sampling rate.

Algorithm 1 Adaptive Sampling Rate Algorithm Based on Anova model and Bartlett Test.

Require: L (1 round = L periods), R, S_{max} (maximum sampling speed), α.
Ensure: S_t (instantaneous sampling speed).

```
S_t ← S_max
while Energy > 0 (the node is still alive.) do
    for i = 1 → L do
        takes measures at S_t speed
    end for
    for each round do
        compute T
        find T_{L-1,α}
        if T ≤ T_{L-1,α} then
            S_t ← Behavior(T, T_{(L-1,α)}, R, S_max)
        else
            S_t ← S_max
        end if
    end for
end while
```

VI. Adapting Sampling Rate Using Jaccard Function

Another technique for adapting sampling rate of a sensor node is by using similarity functions. These functions were used in various domains and applications in order to identify near duplicate records. Therefore, a variety of similarity functions have been proposed in the literature such as Overlap coefficient, Jaccard similarity, and Dice similarity [19]–[22]. In this work, we propose to use the Jaccard similarity function for several reasons: it is one of the most popular and used functions; it can be converted to many other functions; the condition of similarity is the hardest to be satisfied [20]. In this section, the sensor node uses the Jaccard function to search similarity between its data collected among successive periods then to adapt its sampling rate depending from the result of similarity.

The Jaccard similarity function returns a value in $[0, 1]$ where a higher value indicates that the sets are more similar. Thus, pairs of sets with high Jaccard similarity value are considered as near duplicates. The Jaccard similarity function, represented by $J(M_i, M_j)$, between two vectors of measures M_i and M_j, (before applying Algorithm 1), is defined as the size of the intersection divided by the union of the two sets as follows:

$$J(M_i, M_j) = \frac{||M_i \cap M_j||}{||M_i \cup M_j||} \geq t_J, \quad (4)$$

where t_J is the Jaccard threshold defined by the application itself.

To take into account the weights assigned to measures in Section III, we redefine the Jaccard similarity function between two sets of measures M_i' and M_j' as follows: (the proof is similar to that of Equation (1) in [22])

$$J'(M_i', M_j') \geq t_J \Leftrightarrow Card_w(M_i' \cap_s M_j') \geq \beta = \frac{2.t_J.Card_w(M_i')}{1+t_J}, \quad (5)$$

where $Card_w(M_i')$ is the sum of the frequencies of the measures in the set M_i', and "$\cap_s$" (similarity overlap) is defined as follows:

Definition 5: Consider two sets of measurements M_i' and M_j', then we define the overlap, $\cap_s$, between them as:

$$M_i' \cap_s M_j' = \{(m_i', m_j') \in M_i' \times M_j' \text{ with weight } wgt_{min}(m_i', m_j')/Similar(m_i', m_j') = 1\},$$

where $wgt_{min}(m_i', m_j') = min(wgt(m_i'), wgt(m_j'))$, the minimum value of the weights of m_i' and m_j'.

Fig.1 shows an example of Jaccard calculation between two sets M_i' and M_j'. The letters indicate the measures while the numbers represent their weights. There are four elements in their overlap, $M_i' \cap_s M_j' = \{A : 5, B : 3, C : 2, D : 2\}$. Therefore, $Card_w(M_i' \cap_s M_j') = 5 + 3 + 2 + 2 = 12$. In addition, $Card_w(M_i' \cup M_j') = Card_w(M_i') + Card_w(M_j') - Card_w(M_i' \cap_s M_j') = 15 + 15 - 12 = 18$, thus, $J'(M_i', M_j')$ = 12/18.

Similarly to the technique presented in Section V, we exploit the $Behavior$ function in order to adapt the sampling

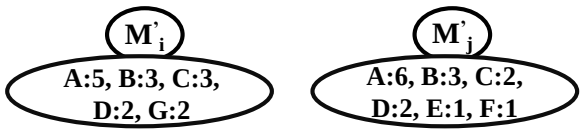


Fig. 1. Two sets with Jaccard similarity $12/18$.

rate (Algorithm 3). We consider a round r equals to two periods, e.g. r= L periods= 2 periods. Indeed, the input of the $Behavior$ function changes according to the similarity function study. Here, the $Behavior$ takes the Jaccard similarity computed between the data sets (i.e. $Card_w(M'_i \cap_s M'_j)$ in line 2 in Algorithm 3) and the Jaccard threshold (t_J) to adapt the sampling rate only if the sets are similar (line 3 in Algorithm 2).

Algorithm 2 Adaptive Sampling Rate Algorithm Based on Jaccard Similarity Function.

Require: round r =L =2 periods, the two sets of measures collected in r: M'_i and M'_j, R, S_{max} (maximum sampling speed), t_J.

Ensure: S_t (instantaneous sampling speed).

- Replace lines 7-13 in Algorithm 1 with

1) search similar measures between M'_i and M'_j, i.e. $M'_i \cap_s M'_j$
2) compute $I = Card_w(M'_i \cap_s M'_j)$
3) **if** $I \geq \beta$ **then**
   $S_t \leftarrow Behavior(I, t_J, R, S_{max})$
4) **else**
   $S_t \leftarrow S_{max}$
5) **end if**

VII. Adapting Sampling Rate Using Euclidean Distance

In this section, we study the utility of the distance functions in adapting the sensors sampling rate. Distance functions have been considered as important way of finding duplicated data sets by searching dissimilarity between these data. Hence, a great number of distance functions have been proposed in the literature [23]. In this paper, we are interested in the Euclidean distance[1] that is widely used in different domains, such as computer vision and face recognition applications [24], and that is already used in WSN during the deployment phase in terms of nodes localization [25] and inter-sensors distance estimations [26].

In mathematics, the Euclidean distance is the ordinary distance, e.g. straight line distance, between two points, sets or objects. Let us consider two data sets M'_i and M'_j, generated by the sensor node S_i in two successive periods. Therefore, M'_i and M'_j are considered redundant if the Euclidean distance (E_d) between them is less than a threshold (t_d) as follows:

$$E_d(M'_i, M'_j) = \sqrt{\sum (m'_i - m'_j)^2} \leq t_d, \tag{6}$$

[1]Cosine distance has been also tested but the obtained results were less important compared to those obtained with Euclidean distance.

where $m'_i \in M'_i$ and $m'_j \in M'_j$.

However, the weights of the measures used in our technique provide two challenges when using distance functions: **1)** calculating the distance between two data sets with different cardinality, and **2)** integrating the weights in the calculation of the distance. To face these challenges, ε-threshold is used which is introduced in the $Similar$ function, in computing the distance between the sets.

Then, in order to find the distance between two sets M'_i and M'_j, the first step consists in dividing each set on two parts: overlap and remained. The overlap part of the set M'_i (respectively M'_j) contains measures that are similar to those in M'_j (respectively M'_i) while the remained part contains the remaining measures of M'_i (respectively M'_j). Subsequently, the overlap part between two sets is already defined in Definition 5, i.e. $M'_i \cap_s M'_j$, while the remained part in each set is defined as follows:

Definition 6 (Remained part of M'_i, M'_{i_r}): Consider two sets of sensor measures M'_i and M'_j. We define the remained part M'_{i_r} (respectively M'_{j_r}) as all the measures in M'_i (respectively M'_j) minus the measures in the overlap part of M'_i (respectively M'_j) as shown in Equation (7):

$$\begin{cases} & M'_{i_r} = M'_i \ominus (M'_i \cap_s M'_j) \\ and & \\ & M'_{j_r} = M'_j \ominus (M'_i \cap_s M'_j) \end{cases} \tag{7}$$

where $\ominus$ is a new operator defined as:

Definition 7 (Minus Operator, $\ominus$): We define the minus operator, $M'_i \ominus M'_j$, between two sets M'_i and M'_j as all the measures in M'_i and not in M'_j as follows:

$M'_i \ominus M'_j = \{m'_i \in M'_i,$ with $wgt(m'_i) = wgt(m'_i) - wgt(m'_j)$ for all $m'_j \in M'_i \cap_s M'_j$ and $Similar(m'_i, m'_j) = 1\}$.

In order to compute the distance between M'_i and M'_j, we must transform M'_{i_r} (respectively M'_{j_r}) to a vector as follows:

$$vM'_{i_r} = [\underbrace{m'_{i_1}, \ldots, m'_{i_1}}_{wgt(m'_{i_1}) \text{ times}}, \underbrace{m'_{i_2}, \ldots, m'_{i_2}}_{wgt(m'_{i_2}) \text{ times}}, \ldots, \underbrace{m'_{i_{k_i}}, \ldots, m'_{i_{k_i}}}_{wgt(m'_{i_{k_i}}) \text{ times}}].$$

Then, we order the measures in vM'_{i_r} (respectively vM'_{j_r}) by increasing order of their values to ensure a logical comparison when calculating the distance between them. Based on the following proposition, the Euclidean distance between M'_i and M'_j is calculated.

Proposition 1: The Euclidean distance between M'_i and M'_j is calculated as follows:

$$E_d(M'_i, M'_j) = \sqrt{\sum_{k=1}^{|M'_{i_r}|} (m'_{i_k} - m'_{j_k})^2}, \tag{8}$$

where $m'_{i_k} \in M'_{i_r}$ and $m'_{j_k} \in M'_{j_r}$.

Proof: Consider two sets of data M'_i and M'_j. Then:

$$E_d(M'_i, M'_j) = \sqrt{(M'_i - M'_j)^2}$$

$$= \sqrt{\Big((M'_i \cap_s M'_j + M'_{i_r}) - (M'_i \cap_s M'_j + M'_{j_r})\Big)^2}$$

$$= \sqrt{\Big((M'_i \cap_s M'_j - M'_i \cap_s M'_j) + (M'_{i_r} - M'_{j_r})\Big)^2}$$

$$= \sqrt{(M'_{i_r} - M'_{j_r})^2}$$

$$= \sqrt{\sum_{k=1}^{|M'_{i_r}|}(m'_{i_k} - m'_{j_k})^2} \text{ where } m'_{i_k} \in M'_{i_r} \text{ and } m'_{j_k} \in M'_{j_r}$$

■

In the above, we consider that the Euclidean distance between the measures in the overlap part is equal to zero because they are redundant. Therefore, the Euclidean distance between two sets is only equal to the distance between measures in the remained parts of M'_i and M'_j, i.e. M'_{i_r} and M'_{j_r}, respectively (Algorithm 4).

Algorithm 4 Euclidean Distance Algorithm.

Require: two sets of measures: M'_i and M'_j.
Ensure: $E_d(M'_i, M'_j)$.
1: find M'_{i_r} and M'_{j_r}
2: $E_d = 0$
3: **for** $k \leftarrow 1$ to $||M'_{i_r}||$ **do**
4: $E_d = E_d + \sqrt{(m'_i[k] - m'_j[k])^2}$; where $m'_i[k] \in M'_{i_r}$ and $m'_j[k] \in M'_{j_r}$
5: **end for**
6: return E_d

Finally, Algorithm 5 describes how the sensor can adapt its sampling rate based on the Euclidean distance. We consider a round r consists of two periods, e.g. r= L periods= 2 periods. Instead of T and $T_{J-1,\alpha}$ used in Bartlett test (Algorithm 2), $Behavior$ function takes the Euclidean distance calculated between the sets in a round and the distance threshold in order to calculate the new sampling rate of the sensor in the case that the sets are redundant (Algorithm 5).

Algorithm 5 Adaptive Sampling Rate Algorithm Based on Euclidean Distance.

Require: round r=2 periods, two sets of measures collected in r: M'_i and M'_j, R, S_{max} (maximum sampling speed), t_d.
Ensure: S_t (instantaneous sampling speed).
- Replace lines 7-13 in Algorithm 2 with
1) $E_d \leftarrow$ Euclidean_Distance(M'_i, M'_j)
2) **if** $E_d \leq t_d$ **then**
$S_t \leftarrow Behavior(E_d, t_d, R, S_{max})$
3) **else**
$S_t \leftarrow S_{max}$
4) **end if**

VIII. Performance Evaluation

To show the effectiveness of our proposal both simulations and real experimentations were conducted. The obtained results are compared to recent data reduction and data compression existing techniques.

A. Simulations Results

In this section we present a set of tests conducted on multiple series of simulations using a custom Java simulator. Our simulations used the real world data set provided by the Intel Berkeley Research Lab [27]. In this dataset, every 31 seconds, 54 Mica2Dot sensors with weather boards collect humidity, temperature, light, and voltage values.

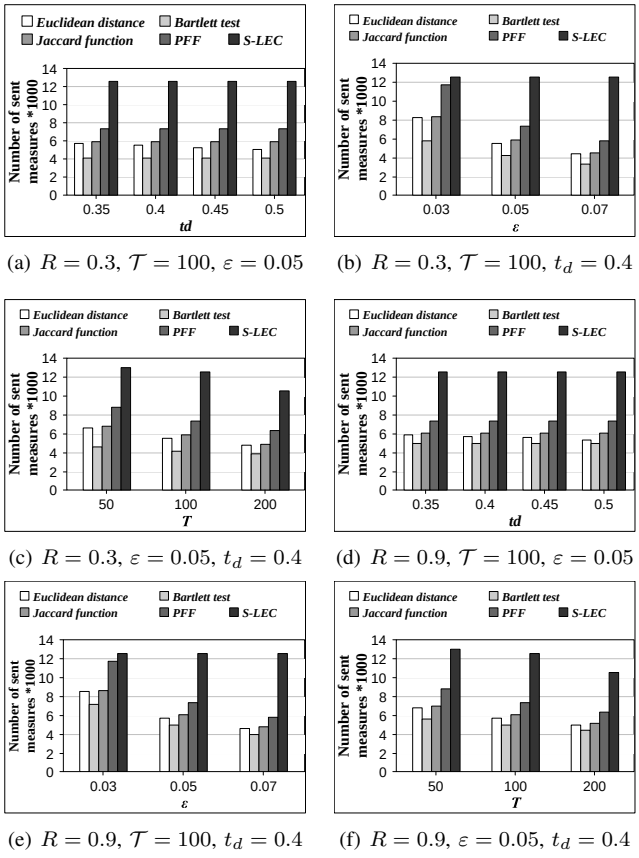
In the remainder and for the sake of simplicity we are only interested in the humidity[2] field. We assume that all nodes send their data to a common cluster-head (CH) placed at the center of the Lab. The objective of these simulations is to compare, first, the three proposed methods for adapting the sampling rate of the sensors under different parameters values. Second, the effectiveness of these methods is tested and compared to a data reduction technique proposed recently, (PFF) technique in [5] and a data compression technique (S-LEC) proposed in [10]. Table II shows the parameters used in the simulations.

Parameter	Description	Value
ε	$Similar$ function threshold	0.03, 0.05, 0.07
$\mathcal{T}$	number of measures taken during one period	50, 100, 200
S_{max}	maximum sensor sampling rate	20, 40, 80
r	round	2 periods
R	application criticality level	0.3, 0.9
t_J	Jaccard similarity threshold	0.75
t_d	distance threshold	0.35, 0.4, 0.45, 0.5
α	false-rejection probability in Anova model	0.05

TABLE II
Simulation environment.

1) Number of transmitted Measures: In this section, we show how our proposal is efficient in reducing the size of data collected and transmitted in the network. Fig. 2 shows the number of measures sent by each sensor after applying $Similar$ function over the collected measures, and using one of the three adapting methods. The results in function of t_d, ε and $\mathcal{T}$ are depicted in Fig. 2(a, b, and c, respectively), where R is fixed to 0.3. Then, in Fig. 2(d, e, and f), R is changed to 0.9 (high application risk) with the same values of parameters as in Fig. 2(a, b, and c). The obtained results show that the proposed methods can reduce at least 17% and 31% the measures sent to the CH, compared to PFF and S-LEC methods. Therefore, these techniques can successfully eliminate redundant collected measures and reduces the amount of data sent to the CH. We can also notice that Bartlett test is the best method in terms of minimizing the amount of the data sent. It can reduce up to 30%, 32%, 50%, and 69% of sent measures compared to Euclidean distance, Jaccard function, PFF, and S-LEC, respectively. The reason is that the Bartlett test searches for the means and variance inter and intra the data sets while the other methods calculate the differences between the sets.

[2]the other fields can be processed in the same manner.



(a) $R = 0.3$, $\mathcal{T} = 100$, $\varepsilon = 0.05$ (b) $R = 0.3$, $\mathcal{T} = 100$, $t_d = 0.4$

(c) $R = 0.3$, $\varepsilon = 0.05$, $t_d = 0.4$ (d) $R = 0.9$, $\mathcal{T} = 100$, $\varepsilon = 0.05$

(e) $R = 0.9$, $\mathcal{T} = 100$, $t_d = 0.4$ (f) $R = 0.9$, $\varepsilon = 0.05$, $t_d = 0.4$

Fig. 2. Number of measures sent by each sensor node.

Based on these results (Fig. 2) we can notice that:

- The sensor node sends less number of measures to the CH when t_d increases (Fig. 2(a or d)). This is because, the dissimilarity beween the data sets is more allowed when t_d increases.
- The number of the sent measures decreases, in the three proposed methods, when ε increases (Fig. 2(b or e)) or $\mathcal{T}$ decreases (Fig. 2(c or f)). According to $Similar$ function, the collected data will be more redundant, in each period and among the successive periods, when ε or $\mathcal{T}$ increases. Furthermore, the three proposed methods give much better results, compared to PFF and S-LEC, in terms of minimizing measures sent to the CH when ε or $\mathcal{T}$ decreases.
- The three proposed adaptive methods increase the amount of the sensed data when R increases. This supports our objective of sending more data in applications with high risk level.

2) Lifetime Estimation: In this section, our objective is to show the effectiveness of our approach in terms of maximizing the lifetime of the sensor node. We assume that each sensor has an energy level fixed to $40mJ$. To evaluate the energy consumption of our approach we used the same radio model as discussed in [27]. In this model, a radio dissipates $E_{elec} = 50nJ/bit$ to run the transmitter or receiver circuitry and $\beta_{amp} = 100pJ/bit/m^2$ for the transmitter amplifier. The equation used to calculate transmission costs for an m-bits message and for a distance d, e.g. distance between the sensor and its CH, is shown as follows:

$$E_{TX}(m, d) = E_{elec} * m + \beta_{amp} * m * d^2. \quad (9)$$

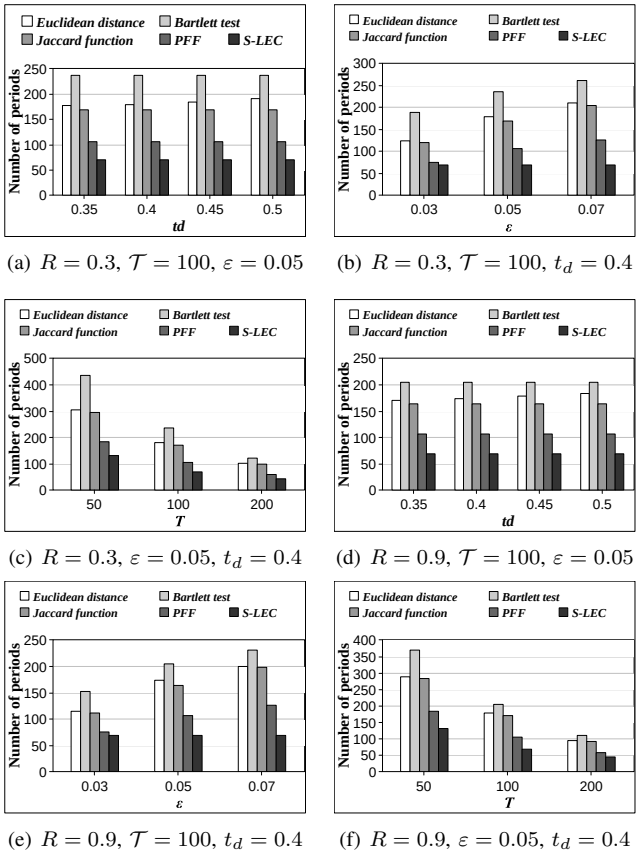
To collect a measure constituted of m-bits a sensor needs [28]:

$$E_{CX}(m, d) = E_{TX}(m, d)/7. \quad (10)$$

Fig. 3 shows the lifetime of a sensor node in terms of the number of periods when varying t_d, R, $\mathcal{T}$, and ε. Indeed, we can find many definitions of the network lifetime in the literature [29]. The most frequently used is that consider that the network dead when the first node fails [29]. Therefore, in this work, we define the network lifetime as the time until the first sensor node in the network runs out of energy. The obtained results show that, our adaptive methods can improve, when $R = 0.3$, the lifetime of the sensor up to 78% and 200% using Euclidean distance, up to 135% and 272% using Bartlett test, and up to 67% and 193% using Jaccard function, compared to the lifetime of the sensor when using the PFF technique and S-LEC, respectively. Otherwise, e.g. when $R = 0.9$, the sensor node can extend its lifetime, using Euclidean distance, Bartlett test, and Jaccard function, up to 72%, 100%, and 56% compared to PFF and up to 185%, 230%, and 182% compared to S-LEC. These results are obtained due to the fact that our methods have minimized significantly the energy consumption during the collection/transmission of data (see results of Fig. 2). Therefore, our methods can be effectively used to increase the sensor network lifetime for both high and low risk level applications, while still keeping the quality of the collected data high.

B. Real-world experimental results

In this section, we describe the experiments conducted on real sensors deployed in our laboratory in order to evaluate our adapting sampling sensor methods. The hardware platform used for data collection was Crossbow telosB motes. Five motes were deployed geographically close in order to monitor temperature and humidity data for four successive days. In the first two days, motes were placed inside the laboratory. They were then placed outdoor during the last two days in order to vary the monitored condition. Data collected by the motes were sent to a specified sink node called SG1000 [30] placed in the center room near about 10 meters from the sensors. The period size is set to 50 measures where each mote takes a new measure of temperature and humidity every 30 seconds, ($p = 25$ minutes). However, due to the limited bandwidth telosB mote, data collected for temperature and humidity fields were sent in two separated packets at the end of each period, after applying our methods. The SG1000 gateway assigned the ID 0 represents the sink node. The three proposed methods (Euclidean distance, Bartlett test, Jaccard function) are implemented on motes with IDs 1, 2, and 3, respectively. Whilst, the naive approach and S-LEC data compression are implemented on motes with ids 4 and 5. Finally, we fixed the parameters to the following values: $t_d = 0.4$, $R = 0.3$, $\alpha = 0.05$, $t_J = 0.75$, and $r = 2$.



(a) $R = 0.3$, $\mathcal{T} = 100$, $\varepsilon = 0.05$ (b) $R = 0.3$, $\mathcal{T} = 100$, $t_d = 0.4$

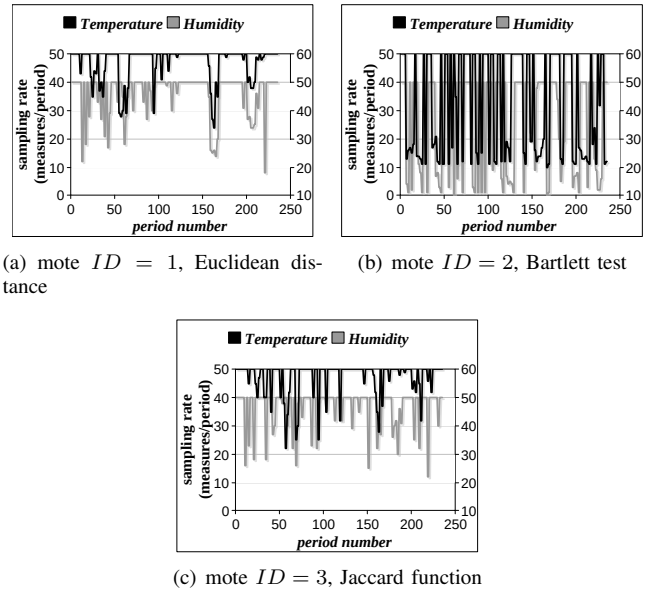
(c) $R = 0.3$, $\varepsilon = 0.05$, $t_d = 0.4$ (d) $R = 0.9$, $\mathcal{T} = 100$, $\varepsilon = 0.05$

(e) $R = 0.9$, $\mathcal{T} = 100$, $t_d = 0.4$ (f) $R = 0.9$, $\varepsilon = 0.05$, $t_d = 0.4$

Fig. 3. Lifetime of a sensor node.

1) Sampling Rate Adaptation: The main goal of this section is to show how our methods were able to adapt the sampling rate of the three deployed motes. Fig. 4 shows the instantaneous sampling rate results for the three motes. Based on the obtained results, we can notice that Euclidean distance, Bartlett test, and Jaccard function successfully adapt the sampling rate of both temperature and humidity sensors in each mote dynamically after each round. Results also confirm the reduction of the amount of collected data compared to the mote with ID=4 operating on $S_{max} = 50$ all time. We can also observe that: **1)** the sampling rates of temperature and humidity vary differently over time and the collected humidity measures are are more numerous than the temperature measures. This means that humidity condition has varied rapidly compared to the temperature condition. **2)** the mote with ID=2 has adapted its sampling rate more than the other motes. This is due to the flexibility of the variance condition calculated in Bartlett test compared to distance and similarity conditions calculated in motes ID=1 and ID=3.

2) Number of Measures Received at the Sink: In this section, we show the number of temperature and humidity measures sent by each mote by applying our methods, naïve, and S-LEC (Fig. 5). The obtained results show that the mote ID 2 with Bartlett test sent the minimum number of measures compared to other motes. Subsequently, the motes IDs 1, 2, and 3 have respectively reduced 27%, 44%, and 25% the temperature measures and 20%, 38%, and 16% the humidity



(a) mote $ID = 1$, Euclidean distance (b) mote $ID = 2$, Bartlett test

(c) mote $ID = 3$, Jaccard function

Fig. 4. Sampling rate adaptation.

measures sent to SG1000 gateway, compared to the S-LEC method implemented on mote ID 5. On the other hand, S-LEC method can reduce 56% of temperature measures and 24% of humidity measures sent to the sink compared to naïve approach. Furthermore, after comparing the humidity results obtained in Fig. 5 to those obtained in Fig. 2, we can observe that: **1)** Bartlett test is the best method in terms of minimizing data collection followed by Euclidean distance and Jaccard function methods, respectively. This confirms the good behavior of our methods in both simulations and experiments environments. **2)** Humidity data collected were more reduced using our methods in the simulations environment (Euclidean: 77%, Bartlett: 83%, Jaccard: 76%), compared to naïve method. This means that the humidity condition in Intel Lab varies slower than that in our laboratory.

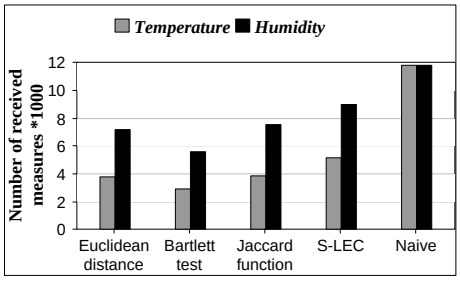


Fig. 5. Total number of measures received at the sink node.

3) Energy Consumption in each Sensor: In this section, we evaluate the performance of our methods in terms of energy consumption in the five motes. There is no model in nesC programming provided by tinyOS in order to measure the energy consumed in telosB [31]. In our experiments, the energy consumption is calculated based on the radio model proposed in [27] as the most used model to evaluate the energy consumption in WSNs. In such model, the energy

consumption in each mote is defined as the total energy dissipation during the collection and the transmission of data. Fig. 6 shows the energy consumption in each mote after four days of deployment, and compares our methods to the naïve approach. Since our adaptive approach reduces the amount of collected/transmitted data in the motes (Fig. 5), energy consumption will be also reduced. These results are shown clearly in Fig. 6 where our methods conserved the energy of the motes IDs 1, 2, and 3 by 29%, 47%, and 25%, respectively, compared to energy consumed in the fifth mote with S-LEC method.

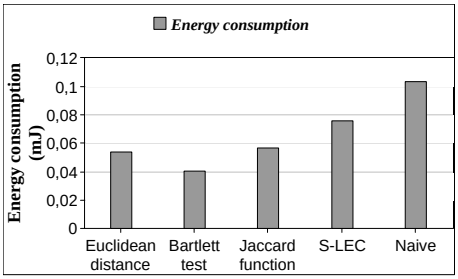


Fig. 6. Energy consumed in each sensor.

4) Data Accuracy: An important factor in WSNs is data accuracy which represents the measure "loss rate". In our experiments, we evaluated the data accuracy by searching periodically the lost measures after adapting the sampling rate of each mote based on our methods. A measure is considered as a lost one if it is captured by the mote ID 4, i.e. naïve method, during a period p and is not collected (similar values) by the other motes in the same period. Then, the global data loss is calculated at the end of the experimentations by considering the number of lost measures in each mote over the number of measures collected by the naïve mote, i.e. mote ID equal to 4. Fig. 7 shows the results of data accuracy for the motes implemented based on our methods and S-LEC. We observe that Jaccard function gives the best results for data accuracy, 3.2% in the worst case, compared to the Euclidean distance (up to 4.6%), Bartlett test (up to 7%), and S-LEC (up to 6.4%). The reason for this is that the Jaccard function is a strong constraint regarding the loss measures compared to distance and variance constraints which are more flexible. Such amounts of loss data are negligible compared to the amount sent to the sink thus, the amount of loss data does not affect the user decision making based on the received data. Therefore we can consider that our methods decrease the amount of collected data forwarded to the sink while conserving the integrity of the information.

C. Further Discussion

In this section, we give further consideration to our proposed methods. We give some directions as to which method should be chosen, under which conditions and in which circumstances of the application.

From the energy preserving point of view, the three proposed methods significantly reduce the energy consumption in sensor node and extends its lifetime (Figs. 3 and 6). In addition, we observe that the method based on Bartlett test conserves more energy compared to the methods based on Euclidean distance and Jaccard function. Therefore, in the applications where we need to conserve the energy of the network as long as possible, the Bartlett test method is more suitable.

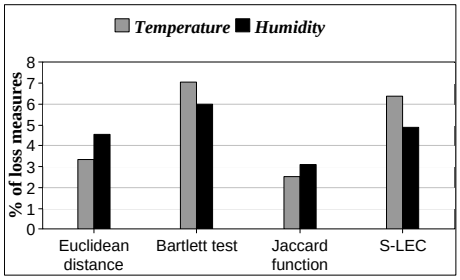


Fig. 7. Percentage of loss measures in each motes.

From the data accuracy point of view, the method based on the Jaccard function and S-LEC can save the integrity of the collected data more than those in other methods. Whilst, the Bartlett test gives the worst results in terms of data accuracy. Hence, if the application does not permit flexibility regarding data accuracy, the Jaccard function method and S-LEC are more suitable; else, Euclidean distance method can be used as a compromise between energy saving and data accuracy flexibility.

IX. Conclusion

This paper proposed three different adaptive sampling rate techniques for IWSNs, which can dynamically estimate the sampling frequency of the collected data. The first one uses the analysis of data variances via statistical tests to adapt the sampling rate, while the second one is based on the sets of similarity functions, and the third one on the distance functions. These techniques were originally conceived to reduce the energy consumption and the data transmissions of sensor networks for process-monitoring applications. We showed via both simulations and real experiments on telosB motes that our approach can be effectively used to increase the sensor network lifetime, while preserving the quality of service expected by the application.

As a future work, we have two major directions. In the first one, we seek to adapt our proposed approach to take into account the correlation between neighboring nodes. As the sensor nodes send their data at the same time (at the end of each period), collisions between packets are likely to happen repeatedly. Then it is essential for sensor nodes to be able to detect this repeated collision and introduce a phase shift between the two transmission sequences in order to avoid further collisions. In the second direction, we plan to allow our approach to adjust the sampling rate on the basis of the available energy beside the redundancies between measures collected in different periods.

Acknowledgements

This work is partially funded by the Labex ACTION program (contract ANR-11-LABX-01-01).

REFERENCES

[1] M. Erdelj, N. Mitton, and E. Natalizio, "Applications of industrial wireless sensor networks," *Industrial Wireless Sensor Networks: Applications, Protocols, and Standards*, pp. 1–22, 2013.

[2] Honeywell, "http://hpsweb.honeywell.com/cultures/en-us/products/wireless/solutions/default.htm," last access: October 2015.

[3] G. Zhao, "Wireless sensor networks for industrial process monitoring and control: A survey," *Network Protocols and Algorithms*, vol. 3, no. 1, pp. 46–63, 2011.

[4] A. Makhoul, H. Harb, and D. Laiymani, "Residual energy-based adaptive data collection approach for periodic sensor networks," *Ad Hoc Networks*, 2015, To appear.

[5] J. Bahi, A. Makhoul, and M. Medlej, "A two tiers data aggregation scheme for periodic sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 21, no. (1-2), pp. 77–100, 2014.

[6] M. Wu, L. Tan, and N. Xiong, "A structure fidelity approach for big data collection in wireless sensor networks," *Sensors*, vol. 15, no. 1, pp. 248–273, 2015.

[7] A. Masoum, N. Meratnia, and P. Havinga, "An energy-efficient adaptive sampling scheme for wireless sensor networks," *8th International Conference on Intelligent Sensors, Sensor Networks and Information Processing,IEEE*, pp. 231–236, 2013.

[8] D. Laiymani and A. Makhoul, "Adaptive data collection approach for periodic sensor networks," *9th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1448–1453, 2013.

[9] K. P. Sampoornam and K. Rameshwaran, "An efficient data redundancy reduction for sensed data aggregators in sensor networks," *Journal of Scientific and Industrial Research*, vol. 74, pp. 29–33, 2015.

[10] Y. Liang and Y. Li, "An efficient and robust data compression algorithm in wireless sensor networks," *IEEE Communications Letters*, vol. 18, no. 3, pp. 439–442, 2014.

[11] M. Gaeta, V. Loia, and S. Tomasiello, "Multisignal 1-d compression by f-transform for wireless sensor networks applications," *Applied Soft Computing*, vol. 30, pp. 329–340, 2015.

[12] ——, "Cubic b-spline fuzzy transforms for an efficient and secure compression in wireless sensor networks," *Information Sciences*, vol. 339, pp. 19–30, 2016.

[13] Y. Jon, "Adaptive sampling in wireless sensor networks for air monitoring system," *Thesis at the University of UPPSALA*, pp. 1–42, 2016.

[14] J. Yang, T. S. Rosing, and S. S. Tilak, "Leveraging application context for efficient sensing," *Proc. of the IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP '14)*, pp. 1–6, 2014.

[15] X. Qi, M. Keally, G. Zhou, Y. Li, and Z. Ren, "Adasense: adapting sampling rates for activity recognition in body sensor networks," *Proc. of the 2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pp. 163–172, 2013.

[16] M. Bahrepour, N. Meratnia, and P. J. M. Havinga, "Sensor fusion-based event detection in wireless sensor networks," *6th Annual International Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous '09*, pp. 1–8, 2009.

[17] Y. Yin, C. Zhang, and Y. Li, "A twostage data fusion model for wireless sensor networks," *International Journal of Sensor Networks*, vol. 15, no. 3, pp. 163–170, 2014.

[18] G. Snedecor and W. Cochran, "Statistical methods," *Eighth Edition: Iowa State University Press*, 1989.

[19] R. J. Bayardo, Y. Ma, and R. Srikant, "Scaling up all pairs similarity search," *16th international conference on World Wide Web, WWW'07*, pp. 131–140, 2007.

[20] S. Sarawag and A. Kirpal, "Efficient exact set-similarity joins," *32nd international conference on Very large data bases, VLDB'06*, pp. 918–929, 2006.

[21] S. Chaudhuri, V. Ganti, and R. Kaushik, "A primitive operator for similarity joins in data cleaning," *22nd International Conference on Data Engineering (ICDE'06)*, p. 5, 2006.

[22] H. Harb, A. Makhoul, A. Jaber, R. Tawil, and O. Bazzi, "Adaptive data collection approach based on sets similarity function for saving energy in periodic sensor networks," *International Journal of Information Technology and Management*, 2015, To appear.

[23] M. M. Deza and E. Deza, "Encyclopedia of distances," *Springer (2009)*, pp. 1–583, 2009.

[24] A. A. Oommen, C. S. Singh, and M. Manikandan, "Design of face recognition system using principal component analysis," *International Journal Of Research In Engineering And Technology*, vol. 3, no. 1, pp. 6–10, 2014.

[25] A. Y. Alfakih, M. F. Anjos, V. Piccialli, and H. Wolkowicz, "Euclidean distance matrices, semidefinite programming, and sensor network localization," *Portugaliae Mathematica*, vol. 68, no. 1, pp. 53–102, 2011.

[26] S. Vural and E. Ekici, "On multihop distances in wireless sensor networks with random node locations," *IEEE Transactions on Mobile Computing*, vol. 9, no. 4, pp. 540–552, 2010.

[27] S. Madden, "Intel berkeley research lab," *http://db.csail.mit.edu/labdata/labdata.html*, 2004.

[28] M. Bagaa, N. Lasla, A. Ouadjaout, and Y. Challal, "Sedan: Secure and efficient protocol for data aggregation in wireless sensor networks," *32nd IEEE Conference on Local Computer Networks, LCN 2007*, pp. 1053–1060, 2007.

[29] I. Dietrich and F. Dressler, "On the lifetime of wireless sensor networks," *ACM Transactions on Sensor Networks*, vol. 5, 2009.

[30] advanticsys, "http://www.advanticsys.com/wiki/index.php?title=sg1000," last modified: September 2012.

[31] J. Griessen, "http://tinyos-help.10906.n7.nabble.com/energy-consumption-on-telosb-td22083.html," 2012.