In-network data processing approach for heterogeneous wireless sensor networks

Ibrahim Atoui, Abdallah Makhoul, Raphaël Couturier and David Laiymani
Univ. Bourgogne Franche-Comté,
FEMTO-ST Institute, CNRS, France
first.last@femto-st.fr

***Abstract*—A wireless sensor network (WSN) is a set of specialized devices that commonly monitor environmental and physical conditions. A critical aspect of applications with WSNs is their limited resources especially in multivariate sensor features when transmitting large amount of data from the nodes to the base station. The aim is then to optimize power consumption during data transmission by using data reduction methods. In this article, we study multivariate data reduction at node's level. We propose a new efficient model based on reducing collected data by aggregation and polynomial regression. We evaluate and compare our method with existing data aggregation techniques, and with the following well-known compression techniques (xz, bzip2, brotli and gzip). The simulation results show that our approach outperforms the existing methods and offers a good approximation of data quality with small approximation errors.**

***Index Terms*—Heterogeneous WSN - data aggregation - data reduction - similarity functions**

I. Introduction

A wireless sensor network (WSN) is composed of a large number of sensor nodes deployed over an area (large or small). These nodes are cheap and small (but not only) devices which sens environmental data and then collaborate to send them to a base station usually called a sink. The main drawback of WSNs is that their resources are strongly constrained (energy and processing power, communication range and memory size).

Heterogeneous WSN are composed of nodes which are able to monitor heterogeneous environmental features such as temperature, humidity, light etc. We then speak of multivariate data (in opposition of univariate data i.e. the sensing of a unique feature). It is also known that the amount of energy consumed during data transmission is much higher than that used for data computation [23]. So, as power consumption in WSN is highly correlated with radio communication, it appears that optimizing the transmission of multivariate data is an important challenge (even more in a large scale deployment) [12], [25].

In this paper, we propose a 2-steps approach which focuses on optimizing the energy consumption in HWSN while optimizing the data transmitted in the network. This aggregation approach is based on the Euclidean Distance computation and on polynomial regression techniques and try to tackle the energy and memory constraints of WSN. Several existing works have been conducted on data aggregation in the WSN context and the majority of them focus on a one field data [14], [15], [19]. We present in this paper an extension with more experimentation of our previous work [6]. Our approach is composed of two levels of data reduction. The first one is data aggregation and the second one is dedicated to data correlation while using polynomial regression. Here, the aim is to only send to the sink one of the correlated fields instead of all. In this way a polynomial regression function is computed and only its parameters are sent to the sink.

The remaining of this paper is organized as follows: Section 2 presents a state of the art. Sections 3 and 4 describe the two phases of data reduction. Experimental results are presented in Section 5. Section 6 concludes the paper and provides future work.

II. State of the art

Energy conservation in WSN is an extensively studied topic. Several approaches have been proposed and here we only focus on data reduction technique which can be categorized in three main groups.

- *Data compression.* The goal here is to reduce the number of bits needed to represent information and in this way decreasing the overall inter-node communication volume. It implies a compression process at the node level and a decoding counterpart at the sink level [13], [16], [29]. The compression can be losslessly (data can be reproduced exactly by the decoding process) or lossily (the compression is better bu the decoding process can only approximate the original data). Lossily compression reduces the application area of this technique, since data such as medical data, emails and other text generally do not tolerate any information loss. We can cite here well-known compression tools such as [3], bzip2 [2], xz [5] and brotli [1]. The main drawback of compression techniques is that the computational cost induced by coding and decoding algorithms is particularly hight
- *Data aggregation*. These techniques reduce the amount of data to be transmitted by removing redundant data due to geographical or temporal proximity, for example [10], [20]–[22]. In a tree-based approach the aggregation processes are computed along a tree, data flowing from leaf nodes to the root i.e. the sink [9]. The main

difficulty is here to build a tree which balances the overall energy consumption in the network [28]. Cluster-based algorithms (hierarchical approach) divide the network in several clusters which elect a cluster-head among their members [28].
Another approach based on prefix frequency filtering (PFF) technique is proposed in [7], [8]. The idea is to find redundant data sets generated by neighboring sensor nodes (by neighboring we mean geographically and/or temporally).

- *Data correlation*. In [11], Banerjee et al. proposed to model the sensed data as a polynomial function in the 2D space via a regression technique. In this way, only the coefficients of the function (and not the raw data) are sent into the network. Their method (TREG) is a tree based polynomial regression algorithm based on the degree of correlation that exists between the sensor data. The authors of [27] also applied a regression technique to propose a new data aggregation algorithm which exploit the spatial correlation of the data. Here, the sensor network is a 3D one.

In this paper, we use both data aggregation and data correlation techniques. This method differs from other methods since it works on two data reduction phases. The two phases are at the node level. First an aggregation technique is applied on the multivariate data sensed. Then a polynomial regression function is computed and its parameters are sent to the sink in a way that the reconstructed values at the sink bear approximation errors.

III. Heterogeneous WSN

Homogeneous data structures are composed of one data field (e.g. light), while heterogeneous data contain several fields (e.g. humidity, temperature, voltage, etc.). In this approach we consider heterogeneous sensor networks where each node collects measures that correspond to different different sensing fields.

In this work, let $N = \{N_1, N_2, \ldots, N_n\}$ represents the set nodes, and n is the total number of nodes in the network. We consider that N_i is equipped of sensors $S_i = \{S_{i_1}, S_{i_2}, \ldots, S_{i_K}\}$, related to different fields. We consider periodic approach and each period is divided into τ slots. At each slot j each sensor S_{i_k} collects a measure. Subsequently, at each slot j, each node N_i takes a data record $M_{i_j} = \left[m_{i_{j_1}}, m_{i_{j_2}}, \ldots, m_{i_{j_K}}\right]$, where $m_{i_{j_k}}$ is collected by the sensor S_{i_k} for slot j. Therefore, at each period p, N_i will form a data matrix V_i as follows:

$$V_i = \begin{bmatrix} M_{i_1} \\ M_{i_2} \\ \ldots. \\ M_{i_\tau} \end{bmatrix} = \begin{bmatrix} m_{i_{1_1}} & m_{i_{1_2}} & \ldots & m_{i_{1_K}} \\ m_{i_{2_1}} & m_{i_{2_2}} & \ldots & m_{i_{2_K}} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i_{\tau_1}} & m_{i_{\tau_2}} & \ldots & m_{i_{\tau_K}} \end{bmatrix}$$

The aim of our first step is then to search the similarity between the line in this matrix while using the Euclidean distance.

A. Euclidean distance computation

The Euclidean distance is a common metric used in a large number of applications. In most of them it is defined as a threshold and is computed between two objects (images, points, lines, sensors...) [24]. In our method the nodes use this distance to compute the similarity between two vectors and the frequency parameter $Freq(M_{i_j}$ of similar data records int eh matrix. The Euclidean distance (E_d) between two data vectors M_{i_a} and M_{i_b} is evaluated as follows:

$$E_d(M_{i_a}, M_{i_b}) = \sqrt{\sum_{l=1}^{L}(m_{i_{a_l}} - m_{i_{b_l}})^2}$$

where $m_{i_{a_l}} \in M_{i_{a_l}}$ and $m_{i_{b_l}} \in M_{i_{b_l}}$.

Thus, M_{i_a} and M_{i_b} are said to be redundant if $E_d(M_{i_a}, M_{i_b}) \leq t_{E_d}$, where t_{E_d} to be determined by the end-user.

The frequency of a data vector M_{i_j}, noted as Freq (M_{i_j}), is the number of subsequent instances of the similar vectors in the same matrix V_i after the Euclidean distance estimation.

In order to be able to perform exact comparison between data sets, normalization of the distance data must be computed. The aim of this step is to constraint the different distances into the $[0, 1]$ range. The length of the vector M_{i_a}, noted as $length(M_{i_a})$, is computed as the distance from the origin vector (or zero's vector) to the vector M_{i_a} as follows:

$$length(M_{i_a}) = \sqrt{\sum_{k=1}^{K} m_{i_{a_k}}^2}, \quad \text{where } m_{i_{a_k}} \in M_{i_a}.$$

The normalisation phase of the Euclidean distance can be done as follows:

$$E_{d_{Norm}}(M_{i_a}, M_{i_b}) = \frac{E_d(M_{i_a}, M_{i_b})}{max\{length(M_{i_a}), length(M_{i_b}\}}$$

B. Similarity computation Algorithm

The data reduction and similarity detection between vectors are presented in Algorithm 1. At a first slot, a sensor node N_i captures the first data vector and initializes its weight to 1 and saves it as the first row in the final matrix of measures (lines 2-4) before sending it to the sink. Then, for each new collected vector, the node searches for similarities with this row with other vectors already save din the final matrix based on the Euclidean distance.

If a similarity is detected, the node deletes the new vector and increments the corresponding frequency of the similar row by 1 (lines 8-12), else it adds it as a new row in the final matrix and assigns its frequency to 1 (lines 15-16). At the end of period, every node will have a reduced matrix with no redundant vectors in terms of rows. Then, it executes the second phase, 'data correlation' and further reduces the matrices of data in terms of columns.

Algorithm 1 Rows similarity at the Nodes level.

Require: new data row $M_{i_j} = \{m_{i_{j_1}}, m_{i_{j_2}}, \ldots, m_{i_{j_K}}\}$ collected at slot s_j, period p.
Ensure: data matrix composed of rows and their frequencies: V_i.

$V_i \leftarrow \emptyset$
if $j = 1$ (s_j is the first slot in p) **then**
  $Freq(M_{i_j}) \leftarrow 1$
  $V_i \leftarrow V_i \cup \{(M_{i_j}, Freq(M_{i_j}))\}$
else
  $found \leftarrow false$
  while $\big((M_{i_k}, Freq(M_{i_k})) \in V_i\big)$ && $(!found)$ **do**
    if $E_d(M_{i_j}, M_{i_k}) \leq t_{E_d}$ **then**
      $Freq(M_{i_k}) \leftarrow Freq(M_{i_k}) + 1$
      disregard M_{i_j}
      $found \leftarrow true$
    end if
  end while
  if $(!found)$ **then**
    $Freq(M_{i_j}) \leftarrow 1$
    $V_i \leftarrow V_i \cup \{(M_{i_j}, Freq(M_{i_j}))\}$
  end if
end if
return V_i

IV. Data correlation

A correlation matrix groups together the correlations of several variables with each other, the coefficients indicating the influence that the variables have on each other which is vey useful when using multivariate techniques. It is, then, possible to better analyse the correlation among features, by reducing the number of dimensions of the underlying structures. To perform this step, many statistical tools exist (PCA, canonical correlation analysis, etc.).

A. Polynomial regression

In this phase, to further reduce the quantity of data that transit in the network, each sensor fits its correlated feature's data to a polynomial function. The objective of this phase is to find the relationship/correlation between the measures of two different fields in the dataset $X_{S_{i_p}}$ and $X_{S_{i_q}}$, where the data-set of a node N_i is $X_i = \{X_{S_{i_1}}, X_{S_{i_2}}, \ldots, X_{S_{i_k}}\}$ and $1 \leq p < q \leq k$.

In order to compute this correlation, we use the polynomial regression method to obtain the following f function as follows:

$$f(X_{S_{i_p}}) = \beta_0 + \beta_1 X_{S_{i_p}} + \beta_2 X^2_{S_{i_p}} + \beta_3 X^3_{S_{i_p}} + ... + \beta_n X^n_{S_{i_p}}$$

where $\beta_i = 1, 2, 3, ..., n$ are the coefficients of the function, and β_0 is noted as the $intercept$ term. However, the assumption of the existence of a linear relationship between data sets is not sufficient. Then, the aim is to assemble our linear model:

$$linearModel = \quad lm(X_{S_{i_p}} \sim X_{S_{i_q}}, \quad datas - et)$$

where $X_{S_{i_p}}$ and $X_{S_{i_q}}$ are the correlated parameters observed in the correlation matrix, and based on a correlation threshold α fixed by the application criticality. When the criticality of the application increases, the value of the threshold tends to 1. The degree of the polynomial depends on the degree of needed precision. Here we have to find a balance between the precision of the model and the complexity of the calculations. Now the quality of a regression prediction can be measured by the coefficient of determination, denoted (R^2). This coefficient can be viewed as a statistical measure of fit i.e. how well a a statistical model fits a data set. An (R^2) of 1 means that the model fits very well the data. Our tests show that beyond degree 3 the accuracy gains become negligible. These results are confirmed with the use of the ANOVA table. In this way, we choose to use a polynomial regression model of the third degree.

Now, if we use the R statistical free software [4], the lm function is the following:

$$fit = \quad lm(X_{S_{i_q}} \sim X_{S_{i_p}} + I(X^2_{S_{i_p}}) + I(X^3_{S_{i_p}}), data - set).$$

Figure 1 sums up our data reduction method. The objective is to reduce the collected matrix in both columns and rows. The data aggregation phase aims at reducing the rows of the matrix, while the correlation phase goal is to reduce the number of columns.

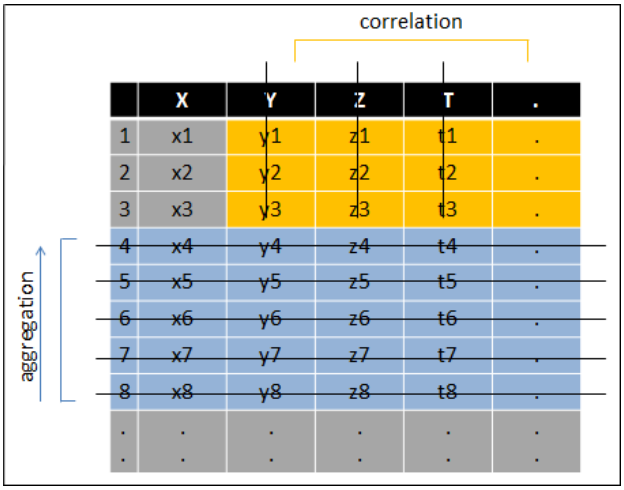


Fig. 1. The reduced matrix of data of our approach.

V. Experimental results

The proposed technique describe in this paper is implemented at the sensor level. Redundant data sensed during the aggregation phase are deleted and the number of parameters sent to the aggregator are reduced with the correlation phase. In this section, we propose to experimentally validate our approach by running a R-based simulator (of our own) on

the well-known Intel Berkeley Research Lab [26] data set. This data set represents the data (humidity, temperature, light and voltage) sensed every slot $s = 31$ seconds by 54 sensors deployed in the lab

We choose ten nodes to run our simulation, with an aggregator located at the center of the lab. We aim at demonstrating the efficiency of our technique more specifically in terms of power consumption. Each node is initialized with a curve fitting algorithm, reads data (measures) saved in file and applies the aggregation phase and tests the correlation between different fields and sends the data (vectors/frequecncies) to the coordinator/aggregator while executing the correlation phase and computing the coefficient values for the polynomial function.

A record means a set of 4 different measures collected at one slot s. The metrics that we evaluate in our experiments are: the the percentage of aggregated data using the Euclidean distance, the percentage of data sent to the aggregator during the second phase, the energy consumption and the data accuracy. We also compared our approach to the PFF technique [7] that considers clustering based networks.

A. Data aggregation at the node level

Here, by using the using the Euclidean distance, similar data (record) are aggregates by each sensor. The frequency assignment for each vector is also performed. The goal is twofold: first , decreasing the size of the sensed data while preserving their integrity. Two parameters are central here: the threshold t_{E_d} and the number of collected vectors by period $\mathcal{T}$.

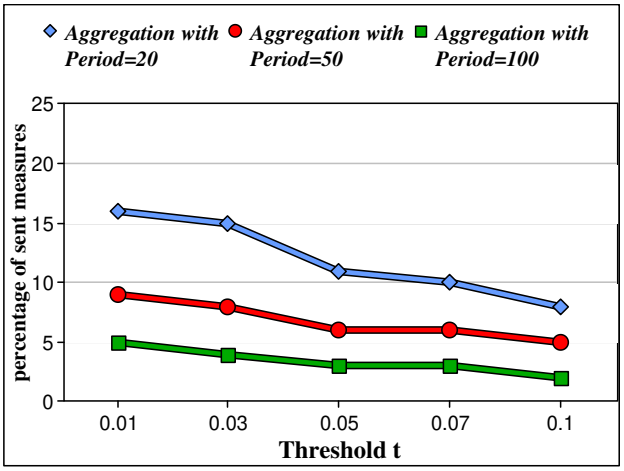


Fig. 2. Percentage of remaining data after the aggregation phase.

In this serie of evaluation, we varied the threshold t_{E_d} between 0.01 and 0.1 (according to the measured field) and $\mathcal{T}$ between 20 and 100. In Figure 2 we show the percentage of the remaining data after using the Euclidean distance. It is noticed that our approach, in the worst scenario, reduces up to 86%. Note also, that, the amount of redundant data decreases when $\mathcal{T}$ or t_{E_d} increases.

B. Correlation results

The objective of this part of simulations is to find the correlated parameters. In our simulations (weather data), α was fixed to 0.9. In Figure 3 we present the correlation matrix of sensor number 1 (after the aggregation phase).

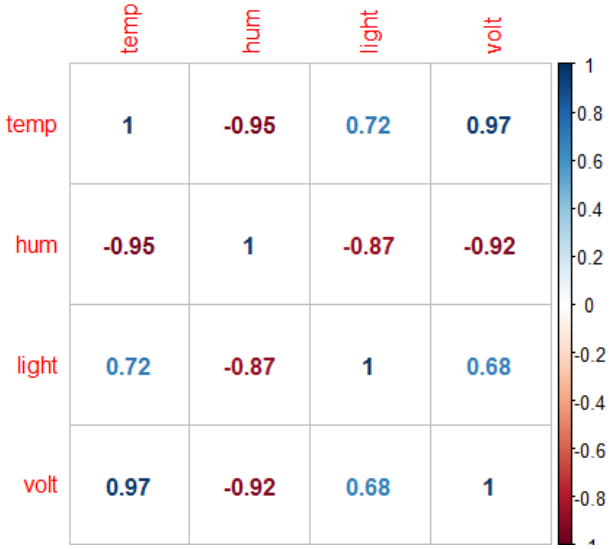


Fig. 3. Correlation matrix of data in sensor 1 after aggregation.

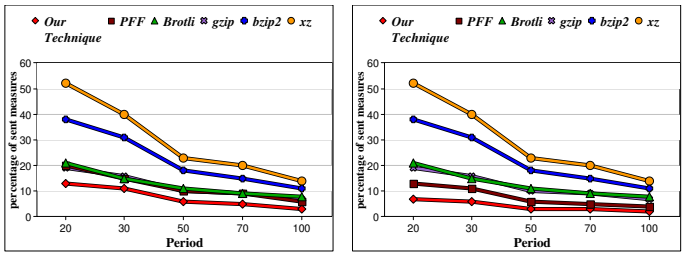
The correlated parameters are $temp$, hum and $volt$ for all the sensor nodes. While studying the used data for our simulations, it appeared that the $temp$ field is the most correlated with other fields. Therefore, the result of the fitting function is realized based on the field of temperature. For instance, when α = 0.9, for node 1 [1], and in order to depict the correlation between $temp$ and hum, the following formula can be used:

$$lm(hum \sim temp + I(temp^2) + I(temp^3))$$

and so forth for $temp$ and $volt$. After this step, the coefficients of the functions and the $temp$ measures are sent to the sink instead of the whole data ($temp$, hum and $voltage$). The sink will then extract the missed values. We compared our approach to the Prefix Frequency Filtering technique PFF [8], [10], and with other compression techniques (xz, bzip2, brotli, and gzip). Figure 4 shows the results of the percentage of vectors sent from sensor nodes to their aggregator, while varying the threshold of the Euclidean distance t_{E_d} and the number of digits n to the right of the decimal point of the values of our data (in order to convert the real numbers into natural for compression methods).

In an another set of experiments, t_{E_d} was fixed at 0.01 and 0.1 while $n = 0, 1, 2$ (we only illustrate $n = 0$ in figure 4). The obtained results show that our approach allows each node to reduce from 2 to 13% of the sets to send to the aggregator. We show also that our method outperforms PFF and the four compression techniques.

[1] the same for other nodes



(a) t_{E_d}= 0.01 (b) t_{E_d}= 0.1

Fig. 4. Percentage of sets sent to the sink at each period with $n = 0$

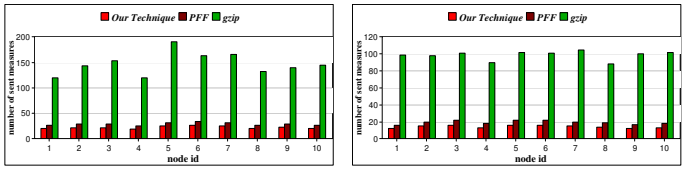
	min	max	median	mean	R^2	MSE
$real$	29.9	39.45	37.37	35.89	-	-
$hum \sim volt$	30.63	39	37.88	35.89	0.9	1.2
$hum \sim temp$	30.28	39.21	37.36	35.89	0.95	0.5

TABLE I
HUMIDITY SUMMARY BEFORE/AFTER THE FITTING

	min	max	median	mean	R^2	MSE
$real$	2.663	2.762	2.7	2.71	-	-
$volt \sim hum$	2.681	2.751	2.705	2.71	0.89	0.09
$volt \sim temp$	2.678	2.751	2.703	2.71	0.95	0.04

TABLE II
VOLTAGE SUMMARY BEFORE/AFTER THE FITTING

Fig. 5 shows an illustrative example for the number of remained data sent to the aggregator after applying our technique, PFF and gzip (i.e. the best compression technique among others). We randomly take ten nodes from the network then, we varied the period size while taking the following values 20, 50 and 100 measures. Similarly to the results shown in Fig. 4(b), we can observe that our technique allows each node to significantly reduce its data sent to the aggregator, compared to other techniques. Indeed, in our technique, the results show that the amount of data sent is varying from node to another; the nodes with $ids = 1$ and 4 sent the minimum number of measures ($\simeq 44$ when $\mathcal{T} = 20$, 20 when $\mathcal{T} = 50$ and 13 when $\mathcal{T} = 100$) to the aggregator while the maximum number of measures is sent by nodes with $ids = 6$ and 7 ($\simeq 65$ when $\mathcal{T} = 20$, 27 when $\mathcal{T} = 50$ and 16 when $\mathcal{T} = 100$)).



(a) $\mathcal{T}$=50. (b) $\mathcal{T}$=100.

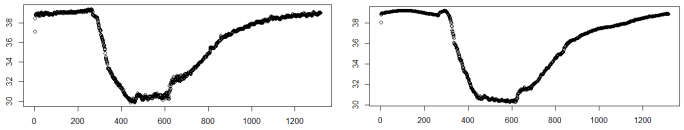
Fig. 5. Number of measures sent to the aggregator with t_{E_d}= 0.1.

C. Data accuracy

In order to study the accuracy of our technique, the difference of R-squared of the correlated parameters, and The mean square error (MSE) are evaluated to see how close the line of regression is close to a set of points. The smaller the MSE is, the closer one is to find the line of best fit.

In tables I and II, one can notice the values of the R-squared and the MSE for the predicted measures of the humidity feature through the fitting with the voltage and the temperature, and the predicted measures of the voltage feature through the fitting with the humidity and the temperature. The results confirm our decision to send the temperature feature instead of the others, and to extract other features through it using the fitting function.

Figures 6 (a, b) illustrate the plotting of the humidity measurements at the nodes and at the sink level. When comparing both figures, it appears clearly that there is no real difference between the plotting of the estimated features between regression function and the original.



(a) at node level. (b) at sink level.

Fig. 6. Humidity measurements plot.

D. Energy consumption study

The energy consumption of the communication radio mainly relies on the volume of data sent over the network and is much higher than data calculation [23]. By reducing the volume of the collected data to be transmitted, our technique aims at extending the network lifetime. This reduction is performed by a data aggregation during the first phase, following by expressing the correlated features by one of them in the second phase. The information integrity is also preserved. In this study we use the well-known energy consumption radio model presented in [17], [18].

$$E_{TX}(k,d) = E_{elec} * k + \beta amp * k * d^2.$$

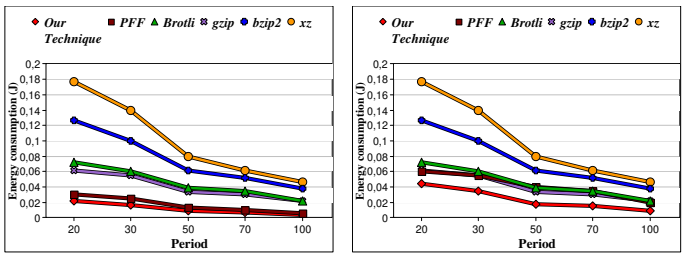
$$E_{comp} = N_{add}\epsilon_{add} + N_{sht}\epsilon_{sht} + N_{cmp}\epsilon_{cmp}$$

$$E = E_{TX} + E_{comp}.$$

Every sensor, at the end of each period, builds a m vectors set with their respective frequencies. This set will be sent by the sensor and its size is equal to the frequencies number plus the number of vectors sent. Each vector is considered to be equal to $64 * p$ bits, p referring to the number of parameters. Figure 7 shows the energy consumption comparison between our technique and other methods while varying the threshold of the Euclidean distance t_{ED} and the number of digits n to the right of the decimal point depending on the period $\mathcal{T}$. We varied $t_{E_d} = 0.1, 0.01$ and $n = 0, 1, 2$ and the obtained results show that our technique outperforms PFF and the compression methods for all values of thresholds and it reduces from 90% to 98%. In figure 7(a) and 7(b) only $n = 0$ is presented.

From these results the following points can be deduced:

- our technique reduces more energy consumption when t_{E_d} increases,
- our technique conserves more energy when $\mathcal{T}$ increases.



(a) $t_{E_d} = 0.1$ (b) $t_{E_d} = 0.01$

Fig. 7. Energy consumption at each sensor with t_{E_d}= 0.01.

VI. Conclusions

A two-phase data reduction approach is described to save energy in heterogeneous WSN. Using the the Euclidean distance function, sensor nodes aggregate the vectors of data before sending them to the coordinator/aggregator. Then, the high correlated parameters are fitted in a manner that the estimated coefficients representing the values of the slope computed by regression are obtained. The deleted data are then reconstructed at the final base station using the fitting function and the computed coefficient values. We showed the efficiency of our method by reducing the size of the data transmitted in the network and thus increasing the network lifetime while guaranteeing the data integrity. In a future work, a matrix similarity approach at the aggregator level will be applied.

Acknowledgement

This work has been supported by the EIPHI Graduate School (contract "ANR-17-EURE-0002").

References

[1] brotli. http://http://www.ietf.org/rfc/rfc7932.txt/.
[2] bzip2. http://http://www.bzip.org/.
[3] gzip. http://http://www.gzip.org/.
[4] R project. https://www.r-project.org/.
[5] xz. http://http://tukaani.org/xz/.
[6] I. Atoui, A. Makhoul, R. Couturier, and J. Demerjian. A multivariate data reduction approach for wireless sensor networks. In *3rd IEEE Middle East and North Africa COMMunications Conference, MENACOMM 2021*, pages 43–48, 2021.
[7] J. Bahi, A. Makhoul, and M. Medlej. Frequency filtering approach for data aggregation in periodic sensor networks. *NOMS 2012*, pages 570–573, 2012.
[8] J. M. Bahi, A. Makhoul, and M. Medlej. Data aggregation for periodic sensor networks using sets similarity functions. *Wireless Communications and Mobile Computing Conference (IWCMC), 2011 7th International*, pages 559–564, 2011.
[9] J. M. Bahi, A. Makhoul, and M. Medlej. An optimized in-network aggregation scheme for data collection in periodic sensor networks. *Ad-hoc, Mobile, and Wireless Networks*, pages 153–166, 2012.
[10] J. M. Bahi, A. Makhoul, and M. Medlej. A two tiers data aggregation scheme for periodic sensor networks. *Adhoc & Sensor Wireless Networks*, 21(1):77–100, 2014.
[11] T. Banerjee, K. Chowdhury, and D. P. Agrawal. Tree based data aggregation in sensor networks using polynomial regression. *Information Fusion, 2005 8th International Conference on*, 2:8–16, 2005.
[12] M. D. de Assunção, A. D. S. Veith, and R. Buyya. Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. *J. Net. and Comput. Applications*, 103:1–17, 2018.
[13] A. Dzhagaryan and A. Milenkovic. On effectiveness of lossless compression in transferring mhealth data files. In *E-health Networking, Application & Services (HealthCom), 2015 17th International Conference on*, pages 665–668. IEEE, 2015.
[14] H. Harb, A. Makhoul, D. Laiymani, and A. Jaber. A distance-based data aggregation technique for periodic sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):32:1–32:40, 2017.
[15] H. Harb, A. Makhoul, S. Tawbi, and R. Couturier. Comparison of different data aggregation techniques in distributed sensor networks. *IEEE Access*, 5:4250–4263, 2017.
[16] J. He, G. Sun, Z. Li, and Y. Zhang. Compressive data gathering with low-rank constraints for wireless sensor networks. *Signal Processing*, 131:73–76, 2017.
[17] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, Maui, HI, USA*, pages 3005–3014. ACM, 2000.
[18] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Trans. on Wireless Comm.*, 1(4):660–670, 2002.
[19] S. M. A. Iqbal and Asaduzzaman. Adaptive forwarding strategies to reduce redundant interests and data in named data networks. *J. Network and Computer Applications*, 106:33–47, 2018.
[20] D. Laiymani and A. Makhoul. Adaptive data collection approach for periodic sensor networks. In *2013 9th International Wireless Communications and Mobile Computing Conference, IWCMC 2013, Sardinia, Italy, July 1-5, 2013*, pages 1448–1453, 2013.
[21] A. Makhoul, H. Harb, and D. Laiymani. Residual energy-based adaptive data collection approach for periodic sensor networks. *Ad Hoc Networks*, 35:149–160, 2015.
[22] K. Maraiya, K. Kant, and N. Gupta. Wireless sensor network: a review on data aggregation. *International Journal of Scientific & Engineering Research*, 2(4):1–6, 2011.
[23] P. Mohanty and M. R. Kabat. Energy efficient structure-free data aggregation and delivery in wsn. *Egypt. Informatics J.*, 17(3):273–284, 2016.
[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*, 3(1):6–10, 2014.
[25] M. A. Rassam, A. Zainal, and M. A. Maarof. Principal component analysis–based data reduction model for wireless sensor networks. *Int. J. of Ad Hoc and Ubiquitous Computing*, 18(1-2):85–101, 2015.
[26] M. S. Intel berkeley research lab. available at http://db.csail.mit.edu/labdata/labdata.html/, 2004.
[27] A. Tripathi, S. Gupta, B. Chourasiya, and A. Jain. Data aggregation for spatially correlated data using polynomial regression in 3d wireless sensor network. *IJARCCE*, 3(10):8353–8358, 2014.
[28] L. A. Villas, A. Boukerche, H. S. Ramos, H. A. de Oliveira, R. B. de Araujo, and A. A. Loureiro. Drina: A lightweight and reliable routing approach for in-network aggregation in wireless sensor networks. *Computers, IEEE Transactions on*, 62(4):676–689, 2013.
[29] E. Zimos, D. Toumpakaris, A. Munteanu, and N. Deligiannis. Multiterminal source coding with copula regression for wireless sensor networks gathering diverse data. *IEEE Sensors Journal*, 17(1):139–150, 2017.