A Multivariate Data Reduction Approach for Wireless Sensor Networks

Ibrahim Atoui[a], Abdallah Makhoul[a], Raphaël Couturier[a], and Jacques Demerjian[b]
[a]*Femto-St Institute, UMR 6174 CNRS, Université de Bourgogne Franche-Comté, France*
[b]*LaRRIS, Faculty of Sciences, Lebanese University, Fanar, Lebanon*

***Abstract*—Efficient data reduction methods are needed to minimize the power consumption in multivariate Wireless Sensor Network (WSN). In this paper, we proposed a distributed multivariate data reduction model for sensor nodes. It is based on reducing data matrices during two phases: in-network data aggregation and polynomial regression. To evaluate the performance of the proposed technique, experiments on real sensor data have been conducted. The obtained results show that our proposed technique outperforms the existing ones in terms of the size of data transmitted over the network and in terms of the energy consumption.**

***Keywords*-Wireless Sensor Network (WSN); data reduction; correlation matrix; polynomial regression;**

I. Introduction

Based on the application purposes of wireless sensor networks (WSN), sensor data can be categorized into univariate and multivariate. Univariate data represent a sample of one phenomenon feature (e.g. temperature) whereas multivariate data represent different features of the phenomenon. Nowadays, sensor nodes are equipped with different types of sensors that provide the ability to monitor different phenomenon features (e.g. temperature, humidity, light, etc.). Indeed, it is clear that the transmission of multivariate data will increase the power consumption because of the high radio communication cost incurred by multivariate sensing data [1], [2].

In this paper, an in-network data processing approach is proposed which uses Euclidean distance intending to tackle the constraints in energy and memory. Data aggregation has been known to be very helpful in saving storage space. However, in this work our focus is on saving energy by reducing the size of the transmitted data. Data aggregation, as defined in some previous works [3], [4], [5], is the process of minimizing the data packets coming from different sources. This reduces the number of needed transmissions and avoids overwhelming amounts of traffic in the network. There has been extensive work on data aggregation schemes in the sensor networks context. The majority of these approaches focus on univariate data reduction [6], [7], [8]. Indeed, these approaches are applied on a set of observations composed of one feature in each node or applied on a set of measures from different nodes. In our approach, multivariate data reduction, based on polynomial regression, is proposed. Actually, each sensor node is equipped with several sensors collecting multivariate data sets such as temperature, humidity, voltage, etc. simultaneously. The proposed model is composed of two phases. The first one aims at reducing the data collected by the sensor node using similarity functions. The second one focuses on the multivariate data reduction in which polynomial regression is applied in each node to transfer one of the correlated features instead of all the features to the sink for approximation.

The rest of this paper is organized as follows: Section II presents a background for multivariate sensor networks. The first phase of our technique and the similarity functions are presented in Section III, while the data correlation and polynomial regression are presented in Section IV. Experimental results are exposed in Section V. Finally, we conclude our paper and we provide some directions for future work in Section VI.

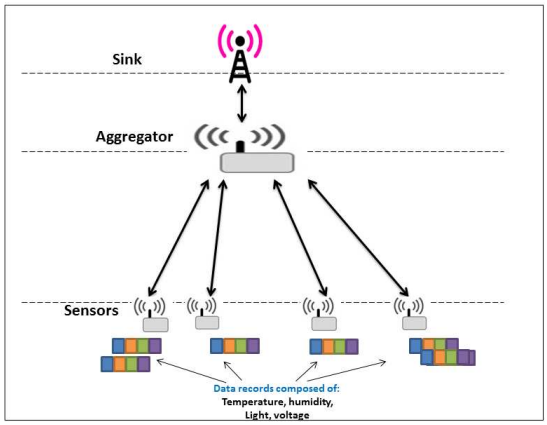

Fig. 1. Multivariate data collection.

II. Background

Homogeneous data structures are those that only contain one data feature e.g. temperature. While heterogeneous data contain a variety of different features, e.g. temperature, humidity, light, voltage, etc. Heterogeneous wireless sensor networks have recently emerged as a new wireless sensor network category which expands the nodes resources and capabilities. Performing data aggregation in heterogeneous sensor networks is more challenging than in homogeneous sensor networks. In the network model adopted for this study and shown in Figure 1, each node collects data records composed of measures of different features.

III. First Phase: Data Aggregation

A. Data Structure

Data aggregation is considered as an efficient technique to save energy consumption in wireless sensor networks. The goal of the filtering approach in [9], [10], [11] is to eliminate similar data. In our approach, $N = \{N_1, N_2, \ldots, N_n\}$ denotes the set of sensor nodes, where n is the total number of nodes in the network. Each node N_i is composed of a set of sensors $S_i = \{S_{i_1}, S_{i_2}, \ldots, S_{i_K}\}$, where each sensor S_{i_k} produces measures related to changes in one physical condition (e.g. temperature). In periodic sensor networks, each period is divided into slots and composed of τ slots. At each slot j each sensor S_{i_k} takes a measure. Subsequently, at each slot j, each node N_i collects a vector of measures $M_{i_j} = \left[m_{i_{j_1}}, m_{i_{j_2}}, \ldots, m_{i_{j_K}}\right]$, 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 matrix of data vectors V_i as follows:

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

Mostly, node N_i takes the same (or very similar) measure vectors several times especially when the slot is too short or the monitored conditions vary slowly. Thus, in order to reduce the number of vectors sent to the aggregator, each node searches the similarities between the generated vectors collected in successive slots. Thus, if the current vector is similar to the one collected in the previous slot, then, the current vector will not be sent to the aggregator in order to save sensor energy. The presented work aims at searching the similarity degree between two vectors based on the Euclidean distance.

B. Euclidean Distance

In mathematics, the Euclidean distance is the ordinary distance, e.g. straight line distance, between two points, sets or objects. Mostly, the Euclidean distance is used in computer vision and face recognition applications, such as the ones used in human computer interaction, ATM machines, prevention of identity theft, etc. [12]. In the first phase of our method, each node uses the Euclidean distance and a frequency function noted $Freq(M_{i_j})$ (defined next) to identify duplicate and similar data vectors collected into two different slots in the matrix V_i.

Definition 3.1 (Euclidean distance): The Euclidean distance (E_d) between two slot vectors M_{i_p} and M_{i_q} is given by:

$$E_d(M_{i_p}, M_{i_q}) = \sqrt{\sum_{k=1}^{K} (m_{i_{p_k}} - m_{i_{q_k}})^2}$$
$$\text{where } m_{i_{p_k}} \in M_{i_{p_k}} \text{ and } m_{i_{q_k}} \in M_{i_{q_k}}$$

Thus, M_{i_p} and M_{i_q} are said to be redundant if $E_d(M_{i_p}, M_{i_q}) \leq t_{E_d}$, where t_{E_d} is a threshold determined by the application.

Definition 3.2 (Slot vector frequency: Freq (M_{i_j})): The frequency of a slot vector M_{i_j} is defined as the number of subsequent occurrences of the same or similar (according to the Euclidean distance) measurements in the same matrix V_i.

C. Distance Normalization

Normalization is a key method for the distance data. The objective is to scale the distance between data sets into the range $[0, 1]$ in order to have the same variation. Consequently, an exact comparison among those sets can be performed. In order to normalize them, let us first define the length of a vector as follows:

Definition 3.3 (Length of the vector M_{i_p}, $length(M_{i_p})$): The length of the vector M_{i_p} is defined as the distance from the origin vector (or zero's vector) to the vector M_{i_p} as follows:

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

Then, the Euclidean distance can be normalized as follows:

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

D. Data Aggregation Algorithm

Algorithm 1 describes the aggregation phase which is run by each node. In the first slot of the period, node N_i takes the first slot vector measures, initializes its weight to 1 and adds it to the final data matrix (lines 2-4). Then, for each new slot vector measurements, N_i searches for similarities of the new taken vector measurements based on the Euclidean distance. If a similar vector measurements is found, it deletes the new one and increments the corresponding weight by 1 (lines 8-12), else it adds the new vector measures to the matrix and initializes its weight to 1 (lines 15-16).

Algorithm 1 Data Aggregation at Node.

Require: Node N_i, new vector measures $M_{i_j} = \{m_{i_{j_1}}, m_{i_{j_2}}, \ldots, m_{i_{j_K}}\}$ collected at slot s_j, period p.
Ensure: reduced data matrix: V_i.

1: $V_i \leftarrow \emptyset$
2: **if** $j = 1$ (s_j is the first slot in p) **then**
3: $\quad Freq(M_{i_j}) \leftarrow 1$
4: $\quad V_i \leftarrow V_i \cup \{(M_{i_j}, Freq(M_{i_j}))\}$
5: **else**
6: $\quad found \leftarrow false$
7: $\quad$ **while** $\left((M_{i_k}, Freq(M_{i_k})) \in V_i\right)$ && $\left(!found\right)$ **do**
8: $\quad\quad$ **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. Second Phase: Data correlation

The term correlation quantifies the extent to which two quantitative features, X and Y, match. In other words, when high values of X are associated with high values of Y, it can be said that a positive correlation exists. Otherwise, when high values of X are associated with low values of Y, a negative correlation can be said to exist. Correlation matrices provide the basis for all classical multivariate techniques. There are many statistical tools to analyze multivariate structures such as: principal component analysis, factor analysis, canonical correlation analysis, and so forth. All of these aim at reducing the high-dimensional multivariate structures to a smaller number of dimensions, so that the relationships among the features will be more readily apprehended.

A. Polynomial Regression

Statistical models can be generated to make predictions or to facilitate understanding. Here the focus is put on model selection primarily in terms of prediction making. Regression analysis is the most useful as it studies the features individually and determines their significance with greater accuracy.

After reducing the number of measures in the aggregation phase by eliminating the similar vectors, less computational complexity is obtained in the second phase of our technique. Now, to further reduce the amount of data transmission from sensor nodes to the aggregator, each sensor fits its correlated parameter's data to a polynomial function. The aim is to try to figure out the relationship between the measures of two features in the dataset $X_{S_{i_p}}$ and $X_{S_{i_q}}$, where the dataset of node N_i is $X_i = \{X_{S_{i_1}}, X_{S_{i_2}}, \ldots, X_{S_{i_k}}\}$ and $1 \leq p < q \leq k$. Then, the existence of a linear relationship between them is considered one of the easiest, most common and effective assumptions to make. However, the truth underlying their relationship is much more complex than that assumption. In this case, polynomial regression can be adapted as a helping method. R is a free software environment for statistical computing and graphics [13]. Our goal is to obtain an R-based 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 called the $intercept$ term. Then, the aim is to assemble our linear model:

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

where $X_{S_{i_p}}$ and $X_{S_{i_q}}$ are correlated parameters obtained from the correlation matrix, according to a correlation threshold α chosen in relation with the application criticality. As the criticality of the application increases, the chosen threshold will be closer to the value 1. The degree of the polynomial at which one stops depends on the degree of precision that is sought. The greater the degree of the polynomial, the greater the accuracy of the model, but the greater the difficulty in the calculations. It is also essential to verify the significance of coefficients that are found. A 10th order polynomial can be fitted and the result is a near-perfect fit. The accuracy of each model depends on the coefficient of determination of the multiple R-squared (R^2) which is a number that indicates how well a data fits a statistical model; sometimes simply a line or a curve. In general, the higher the R-squared, the better the model fits our data. When trying to find out which polynomial degree is better with respect to its accuracy, results show that the third degree gives a significant increase in the accuracy compared to the first and second degrees. This accuracy increases in a negligible way when degrees greater than 3 are tested. In addition, with the ANOVA table, the models are compared, and in our case study, the polynomial of 3rd degree is the best to choose. According to the previous results, it was decided to apply the fitting of a polynomial regression model of the third degree. The centerpiece for linear regression in R is the lm function:

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

Four steps can be identified in fitting distributions [14]: Model/function choice, estimate parameters, evaluate quality of fit, goodness of fit statistical tests. The fitting functions can easily be stored in a WSN node, despite their limited storage space [15]. The omitted parameters during data transmission from the aggregator to the sink can be recomputed at a latter phase using the fitting function and its values.

V. Experimental results

The two phases (aggregation and correlation) proposed in our technique are integrated at the sensor level. It allows each sensor, to eliminate redundant captured measure vectors at the aggregation phase and then to reduce the number of parameters sent to its proper aggregator at the correlation phase of each period. To validate our proposed technique, a R based simulator that is run on the data collected from 54 sensors deployed in the Intel Berkeley Research Lab [16].

Ten nodes were chosen to simulate local aggregation step and it was assumed that the network contains one aggregator located at the center of the lab and a set of sensors that periodically send their data to the aggregator. The goal is to demonstrate that this technique can successfully achieve desirable results in decreasing the power consumption of a heterogeneous WSN. At each slot s = 31 seconds, sensors with weather boards collect humidity, temperature, light and

voltage values together. Each node periodically reads real measures saved in a file while applying the aggregation phase. Each sensor node is assumed to be preloaded with a curve fitting algorithm. At the end of the first phase, each node tests the correlation between features and decides to send the collected records of vectors/frequencies to the aggregator using the correlation phase while calculating the coefficient values for the polynomial function. A record means a set of 4 different features measures captured at a slot s. Our approach was evaluated while taking into account the following metrics: the percentage of aggregated data during the first phase, the percentage of data sent to the aggregator during the second phase, energy consumption and data accuracy. Furthermore, in our experiments our proposed technique is compared to a classical clustering approach without aggregation at the node level and then to the most recent published version of the PFF technique [10].

A. Aggregation using Euclidean Distance

During this phase, each sensor aggregates similar records captured at each slot using the Euclidean distance and assigns to each vector its frequency. The goal of this phase is to reduce the size of the data collected by each node while preserving the frequency of each record of measures in order not to affect the analysis at the sink level. The result depends on the chosen set of thresholds t_{E_d} and the number of collected records by period $\mathcal{T}$.

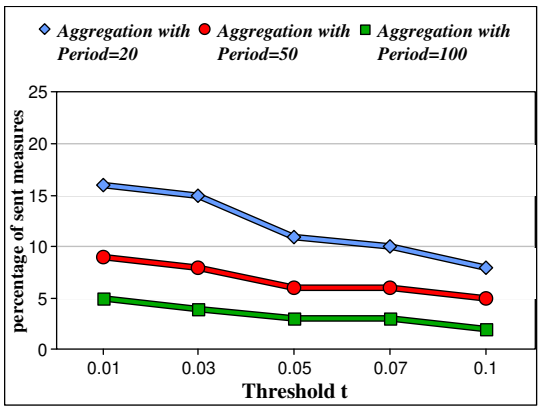

Fig. 2. Percentage of remaining vectors after the first aggregation phase.

In these simulations, t_{E_d} varied between 0.01 and 0.1 (it depends on the collected measures) and $\mathcal{T}$ between 20 and 100. Figure 2 shows the percentage of the remaining measures at each period with and without aggregation at each sensor. The obtained results show that our method can, in the worst scenario, reduce up to 86% during the aggregation phase. Therefore, our technique can successfully eliminate redundant measures at each period and reduce the amount of data sent to the aggregator. We can also observe, that, at the aggregation phase, data redundancy decreases when $\mathcal{T}$ or t_{E_d} increases.

B. correlation and fitting

At this step the aim is to further reduce the number of sent data. At the local aggregation step the number of sent data through rows was reduced, but with the correlation matrix the correlated parameters can also be found. Therefore, less parameters are sent through columns from the nodes to the aggregator. In every studied application, the threshold of correlation α can be chosen according to the criticality of the data. In our data (weather-concerned data), α was chosen as being equal to 0.9.

In all sensor nodes, the correlated parameters are $temp$, hum and $volt$. In our case study the $temp$ feature is the data which is the most correlated by the other parameters. In this way it was decided to get the result of the fitting function in terms of temperature. For example, when α = 0.9, for node 1 (likewise for other nodes), and in order to model the relationship 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 $volt$. Then, the nodes send the coefficients of the functions to the aggregator instead of their original data. Thus, in addition to the coefficients, the $temp$ measures are sent without sending hum and $voltage$, together with the other non correlated parameters in the data. The unsent parameter values can be extracted at the sink from the fitting functions.

We compared our approach to the PFF which is an in-network technique proposed recently to periodic sensor networks that eliminates redundant data over the network, and with other four well-known compression methods (brotli,gzip, bzip2, and xz) existing in the literature for data reduction. Figures. 3 and 4 show the results of the percentage of data sent from some nodes to their proper aggregator, independently of the periods, while varying the threshold of the Euclidean distance t_{E_d}. In Figure 3, the threshold t_{E_d} was fixed at 0.01, and in Figure 4 it is fixed to 0.1. The obtained results show that with our technique each sensor reduces from 2 to 13% of sets sent to the aggregator. It outperforms PFF and the four compression methods. It can be noticed that in the best cases in different periods, PFF reduces from 5 to 20% of sets sent to the aggregator, and gzip reduces from 8 to 52% of sets sent to the aggregator depending on the number of digits taken to the right of the decimal point of data values.

C. Data accuracy

To test the data accuracy of our work, the difference between the values in the correlation matrices was analyzed before and after the aggregation, in addition to the difference of R-squared of the correlated parameters, where R-squared is the metric to evaluate how well our model fits.

Table I shows the R-squared values of the high correlated parameters, before and after the local aggregation phase, where t, h, v respectively refer to the correlated parameters $temperature$, $humidity$, $voltage$. The simple difference of values in all sensors shows that the loss of data due to local aggregation has no effect on the correlation between parameters and the adequacy of our models fitting.

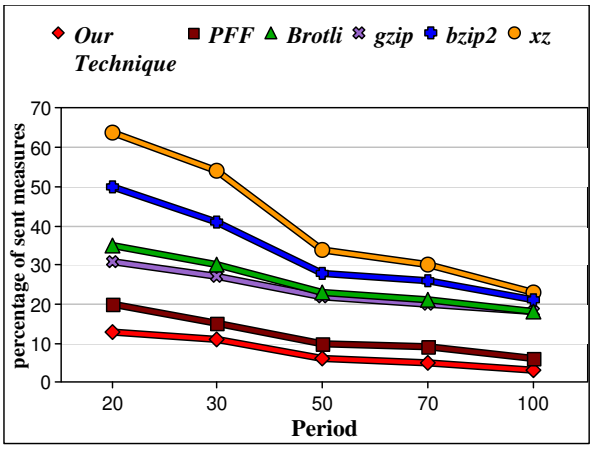

Fig. 3. Percentage of sets sent to the aggregator at each period with t_{E_d}= 0.01.

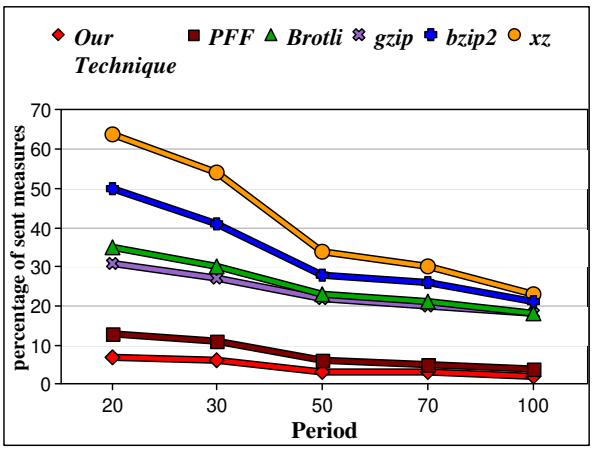

Fig. 4. Percentage of sets sent to the aggregator at each period with t_{E_d}= 0.1.

VI. Conclusions and Future Work

A two-layer data reduction technique is presented in this work to save energy in multivariate wireless sensor networks at the node level. First, each sensor aggregates captured measures based on the Euclidean distance. Second, the high correlated parameters are fitted so that the estimated coefficients representing the values of the slope calculated by the regression are obtained. The omitted parameters during the regression phase are recomputed at the sink using the fitting function and its coefficient values. The efficiency of our approach in terms of data reduction and energy consumption is shown through simulations on real data measurements. Furthermore, it was shown that our technique outperforms the existing PFF technique dedicated to data aggregation in WSN, in addition to four well-known compression methods.

Acknowledgement

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

	before aggregation		after aggregation	
Sensor ID	t and v	t and h	t and v	t and h
1	0.9	0.94	0.9	0.93
2	0.92	0.91	0.93	0.9
3	0.87	0.92	0.87	0.91
4	0.84	0.91	0.85	0.91
5	0.9	0.92	0.91	0.91
6	0.85	0.95	0.89	0.94
7	0.83	0.92	0.86	0.91
8	0.88	0.93	0.89	0.92
9	0.9	0.92	0.92	0.9
10	0.96	0.95	0.96	0.95

TABLE I
R-squared values before/after the aggregation phase

References

[1] Marcos Dias de Assunção, Alexandre Da Silva Veith, and Rajkumar Buyya. Distributed data stream processing and edge computing: A survey on resource elasticity and future directions. *J. Network and Computer Applications*, 103:1–17, 2018.

[2] Murad A Rassam, Anazida Zainal, and Mohd Aizaini Maarof. Principal component analysis–based data reduction model for wireless sensor networks. *International Journal of Ad Hoc and Ubiquitous Computing*, 18(1-2):85–101, 2015.

[3] Hassan Harb, Abdallah Makhoul, Samar Tawbi, and Raphaël Couturier. Comparison of different data aggregation techniques in distributed sensor networks. *IEEE Access*, 5:4250–4263, 2017.

[4] Hassan Harb, Abdallah Makhoul, David Laiymani, and Ali Jaber. A distance-based data aggregation technique for periodic sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 13(4):32:1–32:40, 2017.

[5] Shahid Md. Asif 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.

[6] Hassan Harb, Chady Abou Jaoude, and Abdallah Makhoul. An energy-efficient data prediction and processing approach for the internet of things and sensing based applications. *Peer Peer Netw. Appl.*, 13(3):780–795, 2020.

[7] Leandro A Villas, Azzedine Boukerche, Heitor S Ramos, Horacio ABF de Oliveira, Regina Borges de Araujo, and Antonio AF 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.

[8] Ali Norouzi, Faezeh Sadat Babamir, and Zeynep Orman. A tree based data aggregation scheme for wireless sensor networks using ga. *Wireless Sensor Network*, 4(08):191, 2012.

[9] Jacques M Bahi, Abdallah Makhoul, and Maguy 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.

[10] Jacques M Bahi, Abdallah Makhoul, and Maguy Medlej. A two tiers data aggregation scheme for periodic sensor networks. *Adhoc & Sensor Wireless Networks*, 21(1):77–100, 2014.

[11] Jacques Bahi, Abdallah Makhoul, and Maguy Medlej. Frequency filtering approach for data aggregation in periodic sensor networks. *NOMS 2012, 13-th IEEE/IFIP Network Operations and Management Symposium*, pages 570–573, 2012.

[12] Abin Abraham Oommen, C. Senthil 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.

[13] R project. https://www.r-project.org/.

[14] Vito Ricci. Fitting distributions with r. *Contributed Documentation available on CRAN*, 96:1–24, 2005.

[15] Jehn-Ruey Jiang, Chih-Ming Lin, Feng-Yi Lin, and Shing-Tsaan Huang. Alrd: Aoa localization with rssi differences of directional antennas for wireless sensor networks. *Information Society (i-Society), 2012 International Conference on*, pages 304–309, 2012.

[16] Madden S. Intel berkeley research lab. available at http://db.csail.mit.edu/labdata/labdata.html/, 2004.