En-Route Data Filtering Technique for Maximizing Wireless Sensor Network Lifetime

Hassan Harb[a,‡], Abdallah Makhoul[b,†] and Chady Abou Jaoude[a,∓]
[a]*Computer & communication department, Faculty of Engineering, Antonine University, Baada, Lebanon*
[b]*Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS, Belfort, France*
Emails: [‡]hassan.harb@ua.edu.lb, [†]abdallah.makhoul@univ-fcomte.fr, [∓]chady.aboujaoude@ua.edu.lb

***Abstract*—Today, we can witness wireless sensor networks (WSNs) in action almost everywhere. Their applications are ubiquitous covering environment, medical care, military, surveillance, etc. While the potential benefits of WSNs are real and significant, there remains two major challenges to fully realize this potential: big data collection and limited sensor energy. To overcome these problems, filtering techniques over data routed to the sink should be used in such a way that they do not discard useful information. In this paper, we propose a new filtering technique dedicated to periodic sensor applications. The first filter is applied at the sensor nodes and aims to reduce their raw data based on the Pearson coefficient metric. The second filter is applied at intermediate nodes, called aggregators. It uses K-nearest neighbor clustering algorithm in order to eliminate data redundancy collected by neighboring nodes. In order to evaluate our technique, experiments on real telosB sensors have been conducted while the results showed significant energy savings and high accurate data collection compared to existing approaches.**

Keywords*—*Wireless sensor networks, periodic applications, filtering techniques, Pearson coefficient, K-nearest neighbor algorithm, telosB motes.

I. Introduction

During the last few years, wireless sensor networks (WSNs) have experienced explosive growth and have massively become a part of peoples lives. In such networks, a large number of sensor nodes are usually randomly scattered in different areas and aim to provide low-cost ambient data collection services. The nodes are characterized by a small size (from cubic inches to cubic millimeters), can have multiple sensors on their board (such as for temperature, humidity, pressure, light, etc.), limited power supply and short range radio communication. Thanks to these smart sensors, a lot of real-world applications have been already deployed including environmental monitoring, medical care, military, agriculture and surveillance systems [1]. Additionally, data collected by sensor nodes are forwarded periodically to a specific access point (sink) for analyzing and decision purposes.

The problems in WSNs start from the data acquisition where a big amount of data about the monitored area should be collected for reliability purposes. Hence, what data to keep and what to discard become important in order to take the right decisions. In addition to big data collection problem, data transmission is another challenging task because of the energy-constrained nature of sensor networks. Indeed, transmitting data consumes most of the sensor energy which is mostly limited and not rechargeable, especially in unattended and hostile environments. Therefore, to avoid the above mentioned problems, filtering techniques have been introduced. Filters aim to remove large quantities of redundant data routed on the network, so as to minimize the amount of transmission and save energy.

In this paper, we propose a new filtering technique dedicated to periodic sensor applications. It aims to transmit a summarized scheme of sensed data (without losing meaning and accuracy) to the sink. Thus, it leads to reduce data transmission rate in the network then, optimizing network resource consumption. Our technique composed of two filters. The first filter is applied at the sensor nodes themselves and aims to reduce their big raw data based on the Pearson coefficient metric. The second filter is applied at intermediate nodes, called aggregators. Each aggregator has to eliminate data redundancy collected by neighboring nodes based on K-nearest neighbor clustering algorithm. To evaluate our technique, experiments on real telosB sensors have been conducted while the results showed significant energy savings and high accurate data collection compared to existing approaches.

The rest of the paper is organized as follows. Section II describes the periodic clustering architecture used in our network. Section III overviews various data reduction and filtering techniques existing in the literature for WSNs. in Section IV, we present the first data filtering proposed for the first level, e.g. sensor nodes, in our technique. Section V describes the second filter in our technique proposed for the aggregator level. Experimentations on real sensors are presented in Section VI. Finally, Section VII concludes our paper and gives some perspectives.

II. Our Network Architecture

In this section, we introduce the network architecture used in our technique. Our proposed filtering technique can be applied efficiently by assuming two main concepts for the network: cluster-based architecture and periodic data acquisition. In the next, we describe each of them in more details.

A. Cluster-based Network

In our system, we assume that each set of sensor nodes send their collected data to an intermediate nodes, called aggregators. Each aggregator has an objective to clean data, using a specific filter defined later, coming from neighboring sensor nodes before sending them to the sink. The aggregators can be defined prior to the network deployment and could have more power than normal sensor nodes, depending on the application requirements. Fig. 1 shows our sensor network architecture, where data transmission between sensor nodes and their appropriate aggregators is based on single-hop communication.

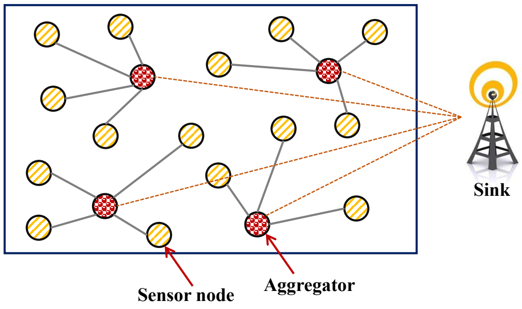
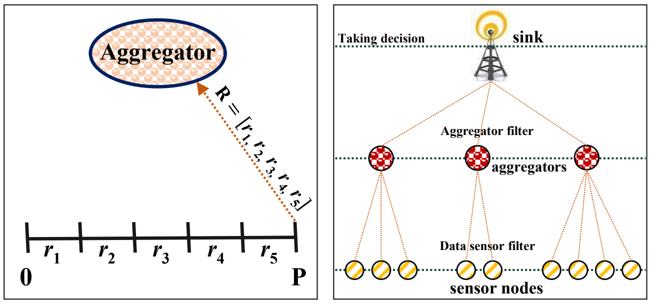


Fig. 1. Two-level data transmission architecture.

B. Periodic Data Acquisition Model

The main mission of WSNs is to forward data packets from event regions to the sink. Unfortunately, sensor nodes are energy-constrained and data transmission task consumes lots of the sensor energy comparing to data processing task. This means that the lifetime of the sensor will shorten if it forwards each sensed data sample to the sink. Hence, periodic data transmission model have been introduced in WSNs in order to reduce the amount of data collected thus, savings sensor energy.

In the periodic acquisition model, data are collected in a periodic basis where each period p is partitioned into time slots. At each slot t, each sensor node N_i captures a new reading r_i. At the end of the period p, N_i collects a vector of τ readings, e.g. $R_i^p = [r_1, r_2, \ldots, r_\tau]$, then it sends it to the sink (Fig. 2(a)). In our system, each sensor node sends periodically (period p) its data to the appropriate aggregator, which in turn sends it to the sink (Fig. 2(b)). Our technique defines two filters: the first one is applied at the sensor level and the second one is applied at the aggregator level.



(a) periodic data acquisition (b) data filtering scheme

Fig. 2. Periodic data filtering scheme.

III. Related Work

The huge amounts of data generated and transmitted by sensors result in increasing energy consumption in WSNs. Hence, a large number of filtering and reduction techniques have been proposed [2], [3], [4], [5] in order to eliminate redundant and meaningless data and consequently ensure energy-aware optimization in WSNs. Furthermore, filtering techniques can intelligently process the raw data to a size that its users can handle while not missing the essential information. In the literature, we can find various data filtering approaches based on data compression, in-network processing or data prediction methods.

The authors in [6] propose an hybridized Least mean squares (LMS) adaptive filter with matrix completion to minimize the necessary information that sensors transmit to sink. The sensor nodes use LMS filter and Bernoulli probability to make a pattern based on which data are sent to the sink. the sink uses matrix completion algorithm to recover missed or lossy data. In [7], the authors propose two data filtering approaches to improve energy efficiency on the agricultural WSNs. The first approach is the simple moving average (SMA) that performs filtering on a sensor node while the second one is based on Threshold Sensitive Energy Efficiency Sensor Network (TEEN) protocol for nodes with only one sensor device attached. A positional prefix-suffix frequency filtering (PPSFF) is proposed in [8]. PPSFF aims to minimize latency of aggregation using a positional filtering that exploits the order of readings both in the prefix and the suffix of a set and leads to upper bound estimations of similarity scores. The authors in [9] propose a data prediction 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.

The authors in [10] propose a supervised linear dimensionality (LDR) reduction technique to reduce the dimensionality of the original data to such that it is well-primed for Bayesian classification. This is done by sequentially constructing linear classifiers that minimise the Bayes error via a gradient descent procedure, under an assumption of normality. In [11], the authors builds a Spanish Inquisition Protocol (SIP) to reduce transmissions in a single-hop wireless sensor system dedicated to monitor temperature in a gas turbine engine application. SIP introduces a selective filtering of sensed data based on state identification, using a skewed double exponentially weighted moving average filter for accurate state predictions. In [12], the authors propose an energy-efficient compressed data reduction frame- work dedicated to underwater sensor network. The proposed framework consists of two layers: the compressed sampling layer, where nodes in clusters are randomly selected to conduct sampling, and the data reduction layer, where full sampling is adopted. The final goal of these layers is to minimize the total energy consumption of transmitting the data sensed by nodes. In [13], 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 diverse WSN data sets. Finally, the authors in [14] propose a prefix frequency filtering (PFF) technique based on clustering architecture of the network. Further to a local processing at the sensor node level, PFF uses Jaccard similarity function to allow aggregator nodes to identify similarities between near sensor nodes at each period and integrates their sensed data into one record.

Although most of the proposed techniques allow efficient data reduction, however they present several disadvantages. They are almost complex, sometimes they generate communication overhead, and the sink may need some transmissions to detect failures. In this paper, we present a novel data filtering method that it is less complex and suitable for limited resources sensor nodes. Then, in order to evaluate our technique, we conducted a set of experiments on a real environment sensor networks based on telosB nodes.

IV. Sensor Data Filtering Model

Usually, the dynamic of the monitored environment in sensor networks can be changed dynamically. Thus, in case of slow down, the sensor node will collect, then sends, more redundant data to the sink leading to drain its available energy. Hence, our objective is to reduce the size of data vector R_i^p collected by a sensor during each period by eliminating redundant readings. We propose to select a subset of readings from R_i^p instead of sending the whole readings in R_i^p to the sink. Our proposed model is based on the Pearson coefficient which is introduced in the next section.

A. Pearson's Coefficient Metric

The Pearson's coefficient indicates the degree of linear correlation between two data sets R_i and R_j, giving a rang of $[1, 1]$, where 1 is total positive correlation, 0 is no correlation, and -1 is total negative correlation. Here each individual sensor node can be regarded as a variable to record the change of observation in the monitored area where the sensor is located.

Consequently, the formula for Pearson's coefficient when applied to two sensor data sets is:

$$\rho_{R_i,R_j} = \frac{n\sum r_i r_j - \sum r_i \sum r_j}{\sqrt{n\sum r_i^2 - (\sum r_i)^2}\sqrt{n\sum r_j^2 - (\sum r_j)^2}} \qquad (1)$$

where $r_i \in R_i$, $r_j \in R_j$ and n is the number of readings in each of R_i or R_j.

Therefore, R_i and R_j are considered to be highly correlated (e.g. redundant) if and only if:

$$\rho_{R_i,R_j} < t_p \qquad (2)$$

where t_p is a threshold determined by the application itself.

B. Sensor Filtering Algorithm

Algorithm 1 shows how each sensor can reduce its vector of collected readings R_i^p at each period. It searches a subset of readings that represent R_i^p by applying iteratively the Pearson's coefficient metric. The main idea behind this algorithm is to divide R_i^p into equal subvectors by applying Pearson's coefficient until the subvectors are highly correlated. This can be made by using the function *divide* which divides a vector of readings into two equal subvectors. Therefore, the process starts by considering that the readings in R_i^p are not correlated (lines 4-7). Then, R_i^p is divided into two subvectors, e.g. $R_{i_1}^p$ and $R_{i_2}^p$ (line 9), and the correlation between them is calculated (line 10). If the correlation is less than the threshold of Pearson's coefficient (line 10) then, the initial vector R_i^p is a final vector of readings. Thus, for each final vector, the mean value of its readings is added to $V_{R_i^p}$ as a representative reading of this vector in addition to the weight of the mean value (lines 11-13). The weight of the mean value indicates the number of readings represented by the mean value (line 12). Otherwise, e.g. the correlation is greater than the threshold, each of the subvectors is divided into equal subvectors and the process is restarted over the readings in the new subvectors (line 16).

Algorithm 1 Sensor Filtering Algorithm.

```
Require: Reading vector: $R_i^p = [r_1, r_2, \ldots, r_\tau]$.
Ensure: Vector of representative readings of $R_i^p$: $V_{R_i^p}$.
 1: $V_{R_i^p} \leftarrow \emptyset$
 2: $V' \leftarrow \emptyset$ // a temporary set of reading vectors
 3: $R_1^p \leftarrow \emptyset$
 4: for each set reading $r_i \in R_i^p$ do
 5:     $R_1^p \leftarrow R_1^p \cup \{r_i\}$
 6: end for
 7: $V' \leftarrow V' \cup \{R_1^p\}$
 8: repeat
 9:     $\{R_{i_1}^p, R_{i_2}^p\} \leftarrow Divide(R_i^p)$
10:     if $\rho_{R_{i_1}^p, R_{i_2}^p} < t_p$ then
11:         find the mean value, $\overline{r}_i$, of readings in $R_i^p$
12:         $wgt(\overline{r}_i) = R_i^p.length$
13:         $V_{R_i^p} \leftarrow V_{R_i^p} \cup \{\overline{r}_i, wgt(\overline{r}_i)\}$
14:         remove $R_i^p$ from $V'$
15:     else
16:         $V' \leftarrow V' \cup \{R_{i_1}^p\} \cup \{R_{i_2}^p\}$
17:     end if
18: until no reading vector $R_i^p \in V'$
19: return $V_{R_i^p}$
```

After applying Algorithm 1, each sensor will send a vector of representative readings $V_{R_i^p} = [\overline{r}_1, \overline{r}_2, \ldots, \overline{r}_k]$ to its proper aggregator, where $k \leq \tau$.

V. Aggregator Filtering Model

At the end of each period, each aggregator will receive a set of representative data sets coming from its sensor nodes. At this level, we propose a second filter allows each aggregator to eliminate redundancy, resulted from temporal correlation between sensed data, among representative sets before sending them to the sink. Our proposed filter is based on data clustering approach. Data clustering is a data exploration technique that allows objects with similar characteristics to be grouped together in order to facilitate their further processing. In this paper, we are interested in K-nearest neighboring (KNN) algorithm adapted to Euclidean distance. In the next section, we explain in more details KNN algorithm as a second filter to clean data at the aggregator level.

A. K-Nearest Neighboring Algorithm

K-nearest neighbors (KNN) [15] is one of the top 10 data mining algorithms used for classification and regression. It is a non-parametric algorithm that does not make any assumptions on the underlying data distribution. KNN algorithm has lots of applications ranging from business[16] and medical [17] to classification of web text [18]. The input of KNN algorithm consists of the entire training dataset. When a comparison is required for an unseen data instance, the KNN algorithm will search through the training dataset for the K-most similar instances (neighbors) and summarizing the output variable for those K instances.

Indeed, distance functions are one of the most similarity measures used in KNN algorithm to search the K-nearest neighbors for a dataset. However, there are huge number of distance functions used in the literature like Euclidean, Cosine, Hamming, Manhattan and so on [19]. We can choose the best distance metric based on the properties/type of the data; for instance, Manhattan distance is a good measure to use if the input variables are not similar in type (such as age, height, etc.); with categorical or binary data, Hamming distance can be used. For real-valued data, e.g. similar in type (all measured temperature or humidity), the most popular distance measure is the Euclidean distance.

B. Euclidean Distance

Computing the distance between a set and all sets in the training datasets is a fundamental process when applying KNN algorithm. In this paper, we are interested in the Euclidean distance that is widely studied and used in different domains. 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, R_i and R_j, then the Euclidean distance (E_d) between them can be calculated as follows:

$$E_d(R_i, R_j) = \sqrt{\sum (r_i - r_j)^2}, \tag{3}$$

where $r_i \in R_i$ and $r_j \in R_j$.

However, the weights of the mean values used at the sensor level makes the computation of the Euclidean distance is not a trivial task. In order to overcome this challenge, we must transform each set of representative readings $V_{R_i^p}$ to a vector as follows:

$$v_{R_i^p} = [\, \underbrace{\overline{r}_1, \ldots, \overline{r}_1}_{wgt(\overline{r}_1)\text{ times}}, \underbrace{\overline{r}_2, \ldots, \overline{r}_2}_{wgt(\overline{r}_2)\text{ times}}, \ldots, \underbrace{\overline{r}_k, \ldots, \overline{r}_k}_{wgt(\overline{r}_k)\text{ times}} \,]. \tag{4}$$

Then, the Euclidean distance between any two representative readings $V_{R_i^p}$ and $V_{R_j^p}$ is calculated based on their readings vectors $v_{R_i^p}$ and $v_{R_j^p}$.

C. Selection of K

The selection of the value of K parameter is very crucial in the KNN algorithm, which is a user-defined constant. In general, a large K value is more precise as it reduces the overall noise on the classification. Heuristic techniques are one of the approaches used to select the proper value of K which is determined by the experts. Another way for the selection of K is by experimenting different values of K (e.g. values from 1 to 20) and see which works best for our problem, i.e. the most accurate results. Historically, the optimal K for most datasets has been between 3-10.

D. KNN Adopted to Euclidean Distance

Algorithm 2 describes the process of KNN algorithm to search the top K similar datasets for a new dataset given as an input for the algorithm. The process starts by computing the Euclidean distance between the new dataset and every dataset in the training set R^p (line 3). Thus, a dataset is added to the final list of top K similar sets of the new set if the list is not yet full (line 4) or its distance to the new dataset is less than the maximum of an existing distance (line 7-10).

Algorithm 2 KNN Adopted to Euclidean Distance Algorithm.

```
Require: List of datasets $R^p = \{R_1^p, R_2^p, \ldots, R_n^p\}$, new dataset $R_j^p$, $K$.
Ensure: List of top $K$ similar datasets to $R_j^p$: $TopK_{R_j^p}$.
 1: $TopK_{R_j^p} \leftarrow \emptyset$
 2: for each dataset $R_i^p \in R^p$ do
 3:     compute $distance = E_d(R_i^p, R_j^p)$
 4:     if $TopK_{R_j^p}.length < K$ then
 5:         $TopK_{R_j^p} \leftarrow TopK_{R_j^p} \cup \{(R_j^p, R_i^p, distance)\}$
 6:     else
 7:         find $R_l^p \in TopK_{R_i^p}$ corresponding to the maximum distance with $R_j^p$
 8:         if $E_d(R_l^p, R_j^p) > E_d(R_i^p, R_j^p)$ then
 9:             replace $R_l^p$ by $R_i^p$
10:         end if
11:     end if
12: end for
13: return $TopK_{R_j^p}$
```

E. Redundant Sets Reduction at the Aggregator

In this section, we show how to integrate the KNN algorithm at the aggregator level in order to search, then eliminate, redundant datasets sent from the sensor nodes at the end of each period (Algorithm 3). First, the aggregator identifies the top K similar sets for each dataset sent by a sensor (lines 3-5) using Algorithm 2. Our objective is to search, for each sensor, the top K nodes that generate highly temporal data correlation in order to reduce the amount of data transmitted to the sink while conserving the integrity of information. Finally, the aggregator removes all pairs of redundant sets that contain $V_{R_i^p}$ or $V_{R_j^p}$ from the set of pairs (which means it will not check them again) (line 8).

VI. Experimental Results

In this section, we describe the experiments we conducted on real sensors deployed in our laboratory in order to evaluate our technique. The hardware platform that we used for data collection was Crossbow telosb motes. Twenty motes have been deployed in our laboratory where each of one monitors temperature data. Data collected by the motes were sent to a specific sink[1] node called SG1000 [20], which can be connected to a laptop machine for retrieving, then analyzing, the collected data. Due to the limited bandwidth of telosB, the

[1] in our experiments, the sink plays the role of an aggregator.

Algorithm 3 Selecting Final Sets Algorithm.

```
Require: List of representative reading sets $V_{R^p} = \{V_{R_1^p}, V_{R_2^p}, \dots, V_{R_n^p}\}$, $K$.
Ensure: List of sent reading sets at period $p$: $V_{L^p}$.
$V_{L^p} \leftarrow \emptyset$
$topk \leftarrow \emptyset$
for each set $V_{R_i^p} \in V_{R^p}$ do
    $topk \leftarrow topk \cup KNN(V_{R^p} - \{V_{R_i^p}\}, V_{R_i^p})$
end for
for each pair of sets$(V_{R_i^p}, V_{R_j^p}) \in topk$ do
    $V_{L^p} \leftarrow V_{L^p} \cup \{V_{R_i^p}\}$ // or $V_{L^p} \leftarrow V_{L^p} \cup \{V_{R_j^p}\}$
    Remove all pairs of sets containing one of the two sets $V_{R_i^p}$ and $V_{R_j^p}$
end for
return $V_{L^p}$
```

period size is set in our experiments to 50 readings where each mote takes a new reading of temperature every 30 seconds. Figure 3 shows the distribution of motes inside the laboratory. Motes are ranged from 1 and 20 respectively while the ID of SG1000 is set to 0. The effectiveness of our technique at the sensor level is tested and compared to a data compression technique (S-LEC) proposed in [13].

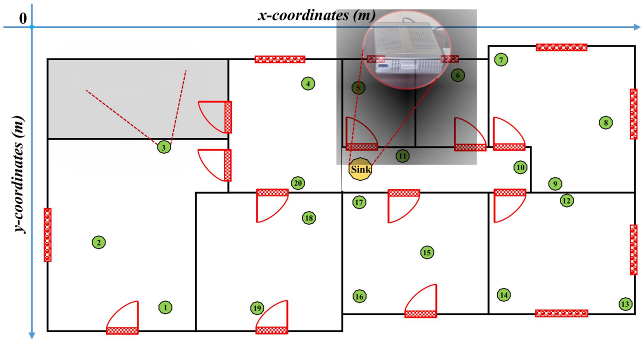


Fig. 3. Distribution of motes in our lab.

Our technique, S-LEC and naïve approach are implemented on the motes as shown in Table I:

Technique	mote IDs
our technique	1, 5, 6, 7, 9, 12, 13, 15, 19, 20
S-LEC	3, 8, 11, 14, 16, 18
naïve	2, 4, 10, 17

TABLE I. TECHNIQUES IMPLEMENTED ON THE MOTES.

Finally, it is important to notice that all methods were implemented on the motes based on the nesC language [21], i.e. the standard programming language of tinyOS [22], while a Java code was implemented on the laptop machine to retrieve data from the sink node.

A. Filtering Ratio at each Mote

Due to the Pearson coefficient used in our technique, each sensor node has the ability to reduce the amount of data collected at each period by eliminating redundant values. Figure 4 shows the average number of temperature readings sent by each mote along the days of deployment, using our technique and S-LEC. The obtained results show that our data filtering model allows motes to significantly reduce its data transmission compared those operating with S-LEC technique. Subsequently, each mote can reduce up to 50% the temperature readings sent to SG1000.

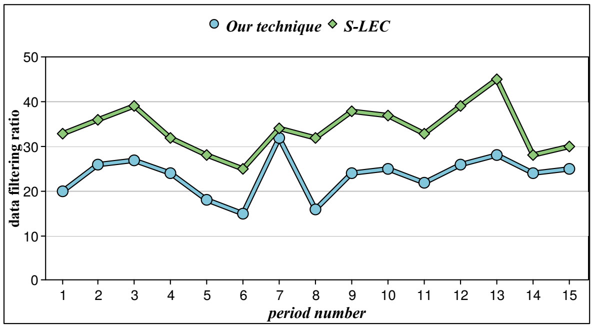


Fig. 4. Filtering ratio at each mote during periods, $\tau = 50, t_p = 0.5$.

B. Filtering Set Ratio at the Sink

In Figure 5, we show the average number of remaining sets after applying KNN algorithm at the sink node, when varying K values to 3, 4 and 5 respectively. The obtained results show that KNN can significantly eliminate redundant data sets generated by neighboring sensors compared to naïve approach, e.g. without any filtering technique. Subsequently, we observe that KNN can reduce up to 85% of the whole received sets at the sink. These results confirm that the clustering is a very efficient approach in terms of eliminating redundant data and providing useful information to the enduser, comparing to other existing approaches. We can also observe that KNN eliminates more sets when K increases; this is because, the temporally correlation between each sensor and it neighboring nodes will increases thus, KNN will consider then eliminate more datasets.

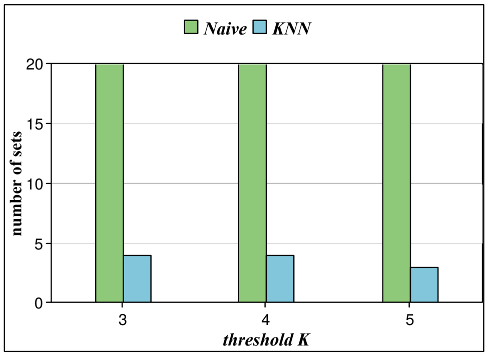


Fig. 5. Filtering set ratio after applying KNN at the sink, $\tau = 50, t_p = 0.5$.

C. Data Accuracy

Data accuracy is an important factor in WSNs which represents the measure loss rate. In our experiments, data accuracy has been calculated by divided the number of loss readings after applying KNN algorithm over the whole readings collected by the naïve sensors. Figure 6 shows the results of data accuracy of KNN compared to S-LEC technique, when varying the threshold K. The obtained results are highly dependent on the number of remaining sets after applying KNN (see results of Figure 5); more the number of remaining sets thus less of readings are lost. Indeed, we observe that both techniques give important results regarding the accuracy of the collected data where the integrity of the information is highly conserved for the end user. Subsequently, we notice that KNN algorithm gives the best results of data accuracy when K is small, e.g. ≤ 4, whilst the information is more conserved using S-LEC when K increases, e.g. > 4.

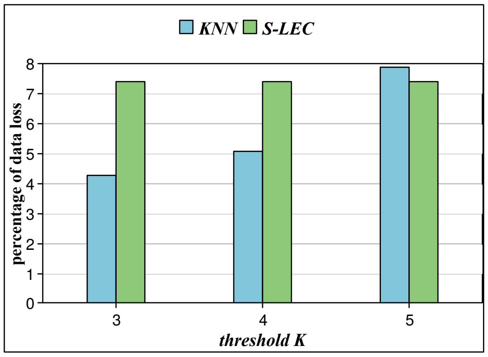


Fig. 6. Percentage of data loss, $\tau = 50, t_p = 0.5$.

VII. Conclusion and Future Work

Wireless sensor networks (WSNs) will play an important role in future internet by collecting surrounding conditions and environment information. Thus, designing new filtering techniques will become essential in order to eliminate meaningless/redundant raw data and make such networks operated as long as possible. This paper proposed energy-efficient filtering technique dedicated to periodic sensor applications. The first filter uses Pearson coefficient metric and aims to reduce the raw data collected by the sensors. The second filter allows aggregator nodes to eliminate redundant data collected by neighboring nodes using K-nearest neighbor clustering algorithm. Or technique has been evaluated based on both simulation and experiments on real telosB sensors. The results obtained with our technique showed significant energy savings and high accurate data collection compared to existing approaches.

We have two major directions for our future work. First, we plan to let aggregators in our technique be able to adjust the sampling rate of the sensors based on the redundancy level with their neighboring nodes. Second, we seek to try another data clustering methods at the aggregator level, like decision trees and neural networks.

References

[1] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, "A comprehensive survey on real-time applications of wsn," *Future Internet*, vol. 9, no. 4, pp. 1–22, 2017.

[2] K. Gupta and A. Singh, "Hybrid data filtering approach for mobile wireless sensor networks," *International Journal of Computer Applications*, vol. 138, no. 8, pp. 37–41, 2016.

[3] R. Bi, L. Sun, X. Zheng, and G. Tan, "Adaptive filter based strategy for data collection in wireless sensor networks," *IEEE International Conferences on Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom), Atlanta, GA, USA*, pp. 1–6, 2016.

[4] T. Maulik, U. Nagaraj, and P. Ganjewar, "Data reduction techniques in wireless sensor network: A survey," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 11, pp. 6621–6632, 2014.

[5] P. Ganjewr, B. S., and S. Wagh, "Threshold based data reduction for prolonging life of wireless sensor network," *International Journal on AdHoc Networking Systems (IJANS)*, vol. 7, no. 1/2/3, pp. 1–8, 2017.

[6] M. El-Telbany and M. Maged, "An energy-efficient wireless sensor networks utilizing lms filter and matrix completion," *International Journal of Applied Engineering Research*, vol. 12, no. 5, p. 591597, 2017.

[7] M. Fajar, J. Litan, A. Munir, and A. Halid, "Energy efficiency using data filtering approach on agricultural wireless sensor network," *International Journal of Computer Engineering and Information Technology*, vol. 9, no. 9, p. 192197, 2017.

[8] H. Harb, A. Makhoul, S. Tawbi, and O. Zahwe, "Energy efficient filtering techniques for data aggregation in sensor networks," *13th International Wireless Communications and Mobile Computing Conference (IWCMC), Valencia, Spain*, pp. 26–30, 2017.

[9] Y. Jon, "Adaptive sampling in wireless sensor networks for air monitoring system," *Thesis at the University of UPPSALA*, pp. 1–142, 2016.

[10] K. Gyamfi, J. Brusey, A. Hunt, and E. Gaura, "Linear dimensionality reduction for classification via a sequential bayes error minimisation with an application to flow meter diagnostics," *Expert Systems with Applications*, vol. 91, pp. 252–262, 2018.

[11] D. J. McCorrie, E. Gaura, K. Burnham, N. Poole, and R. Hazelden, "Predictive data reduction in wireless sensor networks using selective filtering for engine monitoring," *Wireless Sensor and Mobile Ad-Hoc Networks*, vol. Springer New York, pp. 129–148, 2015.

[12] H. Lin, W. Wei, P. Zhao, X. Ma, R. Zhang, W. Liu, and T. Peng, "Energy-efficient compressed data aggregation in underwater acoustic sensor networks," *Wireless Networks Journal*, vol. 22, no. 6, pp. 1985–1997, 2016.

[13] 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.

[14] 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.

[15] N. S. Altman, "An introduction to kernel and nearest-neighbor nonparametric regression," *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[16] S. B. Imandoust and M. Bolandraftar, "Application of k-nearest neighbor (knn) approach for predicting economic events: Theoretical background," *International Journal of Engineering Research and Applications*, vol. 3, no. 5, pp. 605–610, 2013.

[17] H. S. Khamis, K. Cheruiyot, and S. Kimani, "Application of k-nearest neighbour classification in medical data mining," *International Journal of Information and Communication Technology Research*, vol. 4, no. 4, pp. 121–128, 2014.

[18] J. Fuli and C. Chu, "Application of knn improved algorithm in automatic classification of network public proposal cases," *IEEE 2nd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA), China*, pp. 28–30, 2017.

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

[20] Advanticsys, "http://www.advanticsys.com/wiki/index.php?title=sg1000," *Online data*, 2012.

[21] D. Gay, P. Levis, D. Culler, and E. Brewer, "nesc language manual," *https://github.com/tinyos/nesc/blob/master/doc/ref.pdf?raw=true*, 2009.

[22] P. Levis and D. Gay, "tinyos programming," *http://csl.stanford.edu/pal/pubs/tos-programming-web.pdf*, 2009.