FKmeans: A Fast Data Classification Technique to Handle Big Data Collected in Sensor Network

Ola Majed[a,‡], Hassan Harb[b,†], Mohamad Hamze[a,‡,], Ali Jaber[a,†]

[a] *Computer Science department, Lebanese University, Beirut, Lebanon*

[b] *Computer Science department, American University of Culture and Education (AUCE), Tyre, Lebanon*

Emails: olamajed0@gmail.com, hassanhareb@auce.edu.lb, {mohammad.hamze,ali.jaber}@ul.edu.lb

***Abstract*— In recent times, The development of wireless sensor netwoks (WSNs) assumes a noteworthy job in the ascent of big data as a large number of their applications gather enormous amounts of data that require processing. Therefore, WSN faces two noteworthy difficulties. To begin with, it handles the big data collection, and second, the energy of sensors will be drained rapidly because of the immense volume of data gathering and transmission. Subsequently, flow look into has been centered around data classification as a proficient technique to decrease big data accumulation in WSNs along these lines upgrading their lifetime. This paper proposes a fast data classification technique called FKmeans, i.e. Fast Kmeans, committed to periodic applications in WSNs. FKmeans comprises of two phase algorithm to improve the time cost of separation estimation of conventional Kmeans calculation accordingly, guarantee ensure fast data delivery to the sink node. The main stage, i.e. center selection stage, chooses a little segment of datasets with the end goal to locate the most ideal area of the centers. The second stage, i.e. cluster formation stage, uses the conventional Kmeans algorithm adopted to the Euclidean distance where the underlying centers utilized are taken from the primary stage. Our proposed technique is validated via simulations on real sensor data and comparison with the conventional Kmeans algorithm. The gotten outcomes demonstrate the adequacy of our technique regarding enhancing the energy utilization and data conveyance delay, without loss in data fidelity.**

Keywords—Wireless sensor networks, Periodic applications, Data Clustering, FKmeans Algorithm, Real Sensor Data.

I. Introduction

Wireless sensor networks (WSNs) have become a highly active research today. Their applicability can be seen diverse domains such as environment, human, medical, industrial, etc. Typically, a WSN consists of a large number of sensor nodes which are densely deployed over the monitored area in order to collect, then transmit, the data periodically to the remote sink node.

Such type of data collection, usually referred to periodic sensor network (PSN), generates a huge number of data which makes data studying and analyzing a difficult task for the enduser. Furthermore, sensor nodes have a non-renewable power supply and, once deployed, must work unattended. Thus, data collection and transmission in WSNs should be minimized in order to reduce the energy consumption and increase the network lifetime.

Hence, to dodge the previously mentioned issues,data clustering and classification techniques have been presented. Clustering/Classification means to amass similar data together to evacuate huge amounts of redundant data steered on the networks, consequently to limit the amount of transmission and conserve energy.

In this work, we propose a fast kmeans, abbreviated by FKmeans clustering technique used for wireless sensor networks in order to reduce the amount of transmitted data in the network, and thus save energy consumption. Two stage algorithms is proposed in this work which strongly outperform the conventional Kmeans regarding time cost of distance calculation among data and center clusters. The main purpose of the first stage (center selection stage) is to find the ideal location of the centers by selecting a small part of datasets instead of selecting the whole datasets. After getting the center clusters from stage one, the second stage will use the conventional Kmeans algorithm which is adopted to the Euclidean distance for assigning each dataset to its corresponding clusters. Hence, FKmeans will show a remarkable reduce of time cost when compared with traditional Kmeans, this is caused by small amount of training data that is used in the first stage which resulted in few iteration loops in the second one.

The rest of the paper is talks about: Works related to data clustering in WSNs are shown in section II, next, section III shows the architecture used in our network, after that, section IV talk about data clustering model which displayed at the aggregator level, section V show the real data sensors simulation, and finally section VI derives our paper and presents some perspectives.

II. Related Work

Data clustering, as defined by some researchers [1, 2, 3, 4, 5], is the process of grouping similar packets coming from different sensors into groups or clusters thus, to eliminate redundancy before sending final datasets to the sink; so that the number of transmissions through the network is consequently reduced. The main goal behind classifying data is to eliminate redundant data transmission in order to enhance the lifetime of the network and send only the information desired by the end user. Nowadays, data aggregation and clustering techniques are the most data classification methods in order to minimize the data redundancy in WSNs.

Recently, clustering-based data reduction techniques have been used due to their importance in reducing the energy consumption in WSNs [6, 7, 8]. In [6], the authors propose EBDSC scheme to enhance the lifetime of the network where the different devices balance the power consumption among them. The life duration of the node is calculated if it is selected as a cluster head. The next cluster head is the node that has the highest lifetime in the same cluster. The authors in [9] propose DMLDA, an efficient data aggregation technique that works on three tasks: activating nodes, clustering nodes, and filtering messages.

The authors propose in [10] a semi-structured aggregation protocol based on multi-objective tree in WSNs. Their main objective is to increase the aggregation probability and then extend the network lifetime. The authors present in [11] a Cycle-Based Data Aggregation Scheme (CBDAS) for energy saving. In this technique, a grid of cells is created in the WSN, each with a head. These heads are linked together to form a cyclic chain and then the data transmission is reduced and the network lifetime is prolonged. The authors suggest in [12] SFEB, which is a Structure-Free and Energy-Balanced data aggregation protocol. This technique relies on two-phase aggregation process and a selection mechanism for dynamic aggregator that realizes the data gathering and reduces the number of transmissions.

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 fast kmeans, abbreviated FKmeans, clustering technique for wireless sensor networks to decrease the data transmission in the network thus save the energy consumption. Then, in order to evaluate our technique, we conducted a set of simulations followed by experiments on a real environment sensor networks.

III. Cluster-based Periodic Network

After being deployed in the field of interest, sensor nodes organize themselves in the network with the sink node. The network's topology plays an important role in WSNs because of its impact on energy consumption and the network reliability. Indeed, There are two major topology have been proposed for WSNs: tree-based and cluster-based. However, due to its ability to reduce transmission distance or hops between sensors and the sink as well as to perform aggregation processing at intermediate nodes, cluster-based scheme has been more used compared to tree-based network. Thus, in this paper, we are interested in the cluster-based topology with the periodic data collection model in sensor networks.

From one hand, with cluster-based topology, 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.

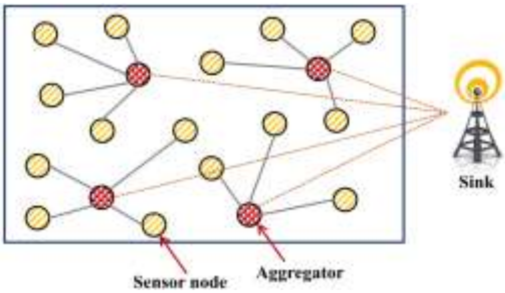
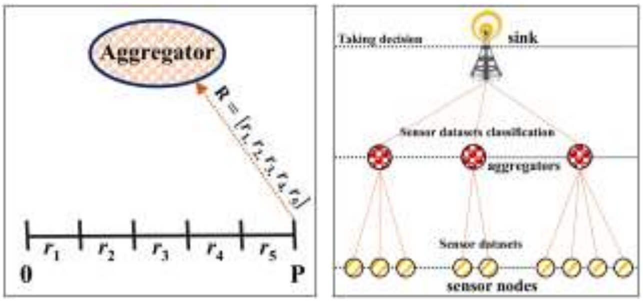


Fig. 1. Distribution of sensors deployed in Intel lab network.

On the other hand, in the periodic collection 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, i.e. $R_i = \{r_1, r_2, ..., r_\tau\}$, 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 objective is to allow aggregator to classify datasets coming from the sensors into groups of similar data then to send only one useful information to the sink node.



(a) periodic data acquisition

(b) data filtering scheme

Fig. 2. Periodic data filtering scheme.

IV. Our Technique

At the end of each period, the aggregator will receive datasets coming from all sensors. Mostly, the spatial-temporal correlation between nodes leads to high redundancy between the received datasets. Hence, the redundancy should be eliminated by the aggregator in order to save its energy and reduce the big size of data sent to the sink.

In order to find redundant data sets received by the aggregator, we propose to use data clustering approach. Data clustering is a data classification technique aims to group object having similar values with each other in order to simplify their processing. In the literature, one can find a huge number of data clustering algorithms like Kmeans, TopK neighboring, etc. However, Kmeans is one the most popular algorithms used in data classification/clustering. Unfortunately, traditional Kmeans suffer from its huge calculation time cost needed to find final datasets clusters. In order to overcome this problem, we propose a new version of Kmeans called FKmeans, Fast Kmeans, which highly enhances the time cost of traditional Kmeans. Our FKmeans consists of two stages of calculation, center selection and cluster formation stages, and uses Euclidean distance to assign datasets to their proper clusters. In the next sections, we first recall traditional Kmeans and Euclidean distance then we details the two stages of our technique.

A. Recall of Kmeans Algorithm

Kmeans algorithm is based on the concept of classifying/grouping data sets into K clusters using the means of sets. As a result, the similarity between sets in the same cluster is high while the similarity between those in different clusters is low. Kmeans clustering is a well-known and well-studied exploratory data analysis technique. The number of clusters is defined by *K* which is a positive integer number. The main idea of Kmeans is to define *K* centroids, one for each cluster. The process of Kmeans starts by taking, each time, a data set from a given data sets then assigns it to the nearest cluster centroid (Fig. 3).

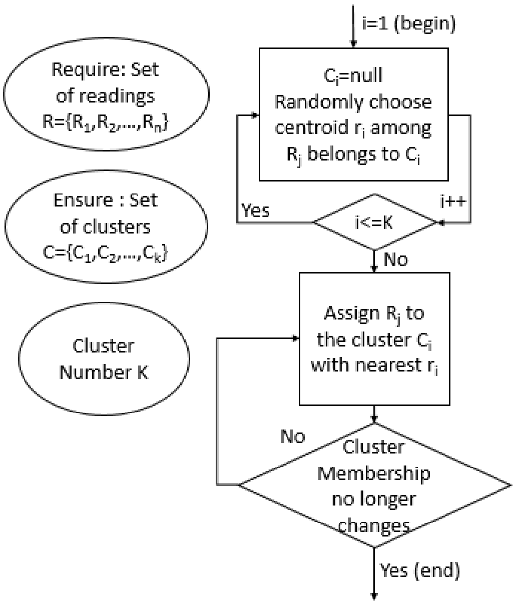

i=1 (begin)
Require: Set of readings R={R1,R2,…,Rn}
Ci=null
Randomly choose centroid ri among Rj belongs to Ci
Ensure : Set of clusters C={C1,C2,…,Ck}
Yes
i++
i<=K
No
Cluster Number K
Assign Rj to the cluster Ci with nearest ri
No
Cluster Membership no longer changes
Yes (end)

Fig. 3. Flow Chart of Kmeans Algorithm.

The first step is done when all points are assigned and an early clusters is forming. Then, we recalculate *K* new centroids as centers of the new clusters. Once the new centroids are calculated, a new assigning has to be computed between the datasets and the nearest new centroid. A loop has been formed. This loop results to change the location of the *K* centroids step by step until no more changes are noticed.

B. Euclidean Distance

Assigning data sets to the nearest cluster centroid is a fundamental process when applying Kmeans algorithm. To do this, we propose to use distance functions as an important way of calculating the distance between two data sets. Indeed, one can find a huge number of distance functions that have been used in the literature like Hamming, Cosine, Euclidean, etc. 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, i.e. 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:

$$E_{d(R_i,\ Rj)} = \sqrt{\sum (r_i - rj)^2} \qquad (1)$$

Where r_i in R_i and r_j in R_j.

C. FKmeans: Fast Kmeans Algorithm

In data classification, the data latency represents one of the main constraint that, in one hand, consumes energy of the aggregator due to the computational power and on other hand, affects timely delivery of data to the sink node. One drawback of the traditional Kmeans algorithm is the time it puts to generate the final clusters. This is due to the computation of Euclidean distance between all the received datasets and the centers of the clusters. Furthermore, this phase of cluster formation can be very complex, especially when it comes to sensor networks where readings' sets can have ten hundreds or thousands elements.

On the other hand, selecting randomly the centroids of the clusters at the initial step consumes a lot of time calculation at the aggregator level. Thus, it also affects the delivery time packet to the sink. Therefore, in order to minimize the data latency for the cluster formation, we propose a new version of Kmeans called FKmeans, Fast Kmeans, which is dedicated to periodic sensor applications having critical issue about the time delivery of packets to the sink node. Our FKmeans consists of two stages: center selection and cluster formation. In the next sections, we detail each of the proposed stage.

1) Center Selection Stage

Mostly, the efficiency and performance of the Kmeans algorithm is greatly affected by initial cluster centers as different initial cluster centers often lead to different clustering. Thus, calculation time cost for the distance between the centroids and the datasets will be high. Hence, selection of the initial center clusters is becoming a challenge for Kmeans algorithm. To overcome this problem, researchers have proposed many techniques like density based, graph based, random based, etc. Unfortunately, most of these methods are very complex and not suitable to the WSN case that is characterized by small processing capacity.

The first stage of our adapted Kmeans is called center selection and aims to solve the above problem. We propose to select a subset/training from the datasets coming to the aggregator node in order to find the approximate final cluster centers. Our intuition is to reduce the number of iterations needed in the traditional Kmeans to obtain the final clusters, thus to enhance the processing time of the Kmeans.

Obviously, the efficiency of the selection center stage is highly related to the percentage, represented by T_s (i.e. training size), of training datasets. Subsequently, increasing the value of T_s leads to increase the calculation time of FKmeans so no profit will be noticed compared to traditional Kmeans. On the other hand, the lowest the value of T_s is the better processing time, and data delivery could be made but the error in the final obtained clusters will increase thus the data accuracy at the sink node. Therefore, selecting the appropriate value of T_s is very essential in the first stage of our technique. Indeed, we believe that T_s should be determined by the decision makers or experts depending on the application requirements. For instance, in health monitoring applications T_s must be lower than weather monitoring applications. Therefore, this parameter is based on the application criticality and the studied phenomenon. After selecting its value, the decision makers assign the threshold T_s accordingly into all sensors nodes prior to deployment or they can adjust it online in function of the application requirement.

2) *Cluster Formation Stage*

After having the cluster centroids in the first stage, our objective now is to use such centers in the second stage in order to form the final clusters. We believe that the obtained centers will help in minimizing the within cluster sum of squares error in the final clusters. Now, our objective in the second stage is to reduce the error with clusters, i.e. between datasets of each cluster. Figure 4 describes the procedure of the second stage of our technique.

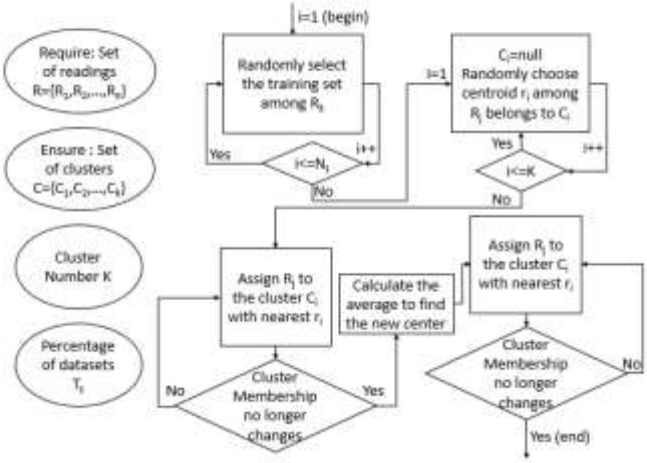

Require: Set of readings R={R1,R2,...,Rn}
Ensure : Set of clusters C={C1,C2,...,Ck}
Cluster Number K
Percentage of datasets Ts
i=1 (begin)
Randomly select the training set among Ri
Yes
i++
i<=Ni
No
i=1
Ci=null Randomly choose centroid ri among Rj belongs to Ci
Yes
i++
i<=K
No
Assign Rj to the cluster Ci with nearest ri
No
Cluster Membership no longer changes
Yes
Calculate the average to find the new center
Assign Rj to the cluster Ci with nearest ri
Cluster Membership no longer changes
No
Yes (end)

Fig. 4. Flow Chart of FKmeans Algorithm.

First, we determine the number of sets needed to find the cluster centers in the first stage of our technique. Based on this number, we randomly select the training sets among the whole datasets *R*. The datasets in the training set represent now the approximate centers of the clusters. Then, we assign each set in training set to the nearest centers. This process is repeated until no more changes in the cluster centers. At

this moment, the first stage is accomplished and the initial centers are determined. After that, the second stage is running where the process starts by considering the centers obtained in the first stage as the initial centers to the clusters. Then, we assign each dataset in the whole datasets to the nearest cluster center. Again, the loop is repeated until the cluster centers become fix in two successive loops.

V. Performance Evaluation

We introduce, in this section, the setup used to validate the relevance and the efficiency of our proposal. We conducted multiple series of simulations using a custom Java based simulator. This section shows the simulation we conducted on data collected in the Intel sensor network [13]. In such network, 46 sensors are deployed in the Intel Berkeley Research Lab for approximately three months collecting more than 3 millions of readings about weather conditions (temperature, humidity and light). Sensor sampling rate is fixed to 1 readings every 31 second. The positions of sensors inside the lab are shown in Fig. 5 (yellow sign indicates the dysfunction of some sensors). For simplicity reason, we show in this section the results of temperature condition. The objective of our simulations was to confirm that our technique can successfully achieve intended results for reducing the energy consumption in sensor nodes and extending network lifetime. In order to evaluate the performance, we compare our results to the traditional Kmeans.

In our simulations, we evaluated the performance using the following parameters:

- the period size, τ, takes the following values: 50, 100, 150 and 200.
- the percentage of data chosen, T_s, takes the following values: 5, 10, 15 and 20.
- the clusters number, *K*, takes the following values: 4, 5, 6 and 7.

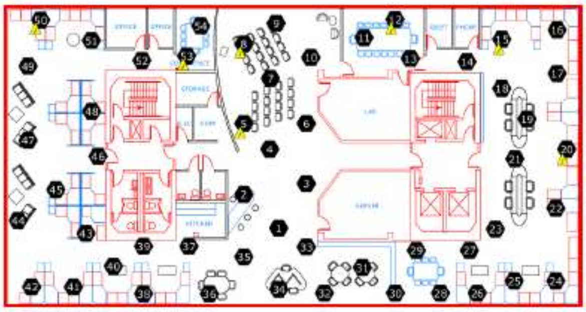
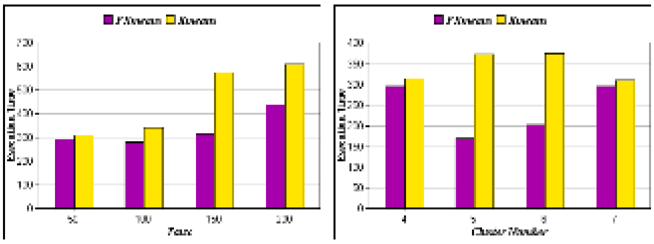


Fig. 5. Distribution of sensors deployed in Intel lab network.

A. *Execution Time*

Sometimes, delivering data as fast time as possible to the enduser is a crucial operation especially in e-health and military applications. Fig. 6, show the execution time at each sensor node for both FKmeans and the traditional Kmeans when varying the period size and the cluster number. The results show that FKmean can optimize the execution time, comparing always to the Kmeans, from 10% (while varying taux from 50 to 100 measures) to 37% (while varying taux from 150 to 200 measures). Obviously, the execution time of FKmeans will be highly affected by the selection of the cluster centroids as well as the number of iteration loops to obtain the final clusters.

Therefore, FKmeans outperforms the normal Kmeans where the processing time at the aggregator is twice accelerated when using FKmeans, compared to Kmeans algorithm to ensure fast data delivery to the sink node and thus save energy.

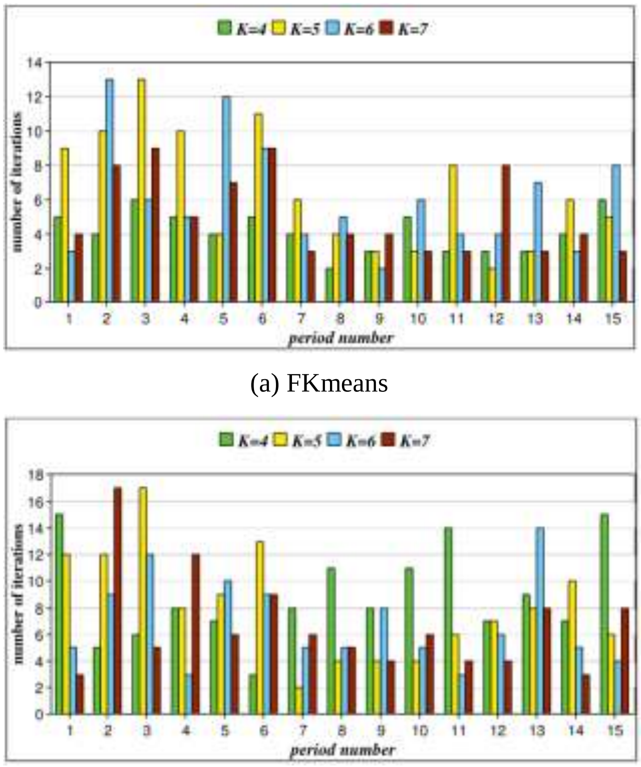


(a) T_s=20, *K*=4 (b) T_s=20, *T*=50

Fig. 6. Processing time at the aggregator level.

B. *Iteration Loop*

One of the factor that can delay the delivery of message is the number of iterations. In Fig. 7, we show how many iterations are generated by the sensor at each period to find the final clusters for both FKmeans and the Kmeans. It is important to know that a high number of iterations can increase the complexity of the proposed algorithm as well as the data latency at the sensor. The obtained results show that, The number of iterations is reduced by at least 20% as shown in these figure when applying FKmeans on the data source. Therefore, FKmeans enhance the data latency by reducing the number of iterations.



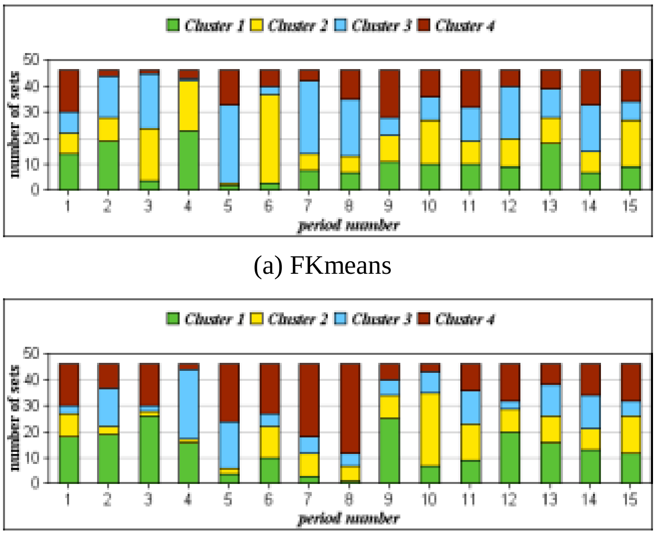
(a) FKmeans

(b) Kmeans

Fig. 7. Iteration loop number for FKmeans and Kmeans, τ = 50, T_s = 20.

C. *Variation of Sets Number Among Cluster*

In this section, we study the distribution of sets between the clusters after applying both FKmeans and Kmeans algorithms along with the period number (Fig. 8). The obtained results show that the sets are distributed in an unequal way into the clusters. This confirms the behavior of our FKmeans algorithm by classifying data sets based on their dissimilarity and not in an equal way.



(a) FKmeans

(b) Kmeans

Fig. 8. Number of sets in each cluster during periods, τ = 50, T_s = 20, *K* = 4.

D. *Illustrative Example of Clusters*

In this section, we show an illustrative example to which sensors are spatio-temporally correlated and how they are clustered using our FKmeans algorithm, during a taken period. We fixed the parameters as shown in Fig. 9. Based on the figure, we can see that data generated by the sensors in the lab are highly spatio-temporally correlated. Furthermore, we can notice that a sensor is more correlated to its nearest neighboring than the other nodes in the cluster. However, sometimes, correlation between distant nodes can be also seen due to the temporal correlation between their generated data.

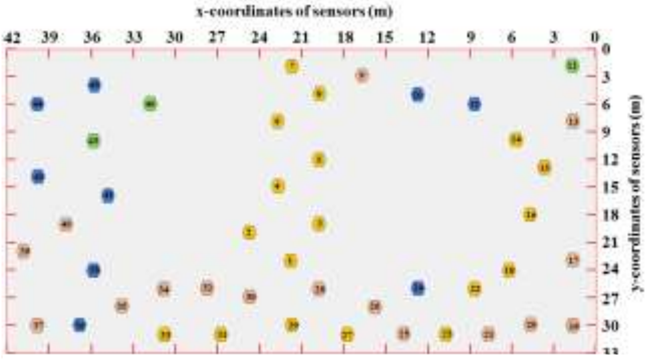


Fig. 9. Example of FKmeans correlated sensors for each node during a period, τ = 50, T_s = 20, *K* = 4.

E. Energy Consumption

In our technique, the aggregator will periodically receive sets of readings coming from all sensor nodes. After clustering redundant ones using FKmeans algorithm, the aggregator selects centers of clusters to be sent to the sink node, as a representative set of the cluster. In our simulation, we implemented the same energy model that used in [14] to calculate the energy consumption in the aggregator level. The proposed model computes the energy consumption in the aggregator when it receives data from sensors as well as sending them to the sink. In addition, we compared our results to those obtained with naïve approach where all datasets are sent from the aggregator to the sink without any clustering. Fig. 10 shows the energy consumed in aggregator depending on the period size. The obtained results show that the energy consumption increases with the increasing of the period size while it is optimized, using FKmeans, up to 60% compared to the naïve approach. Therefore, our proposed technique can be considered very efficiently in terms of reducing the network energy consumption, thus, increasing its lifetime.

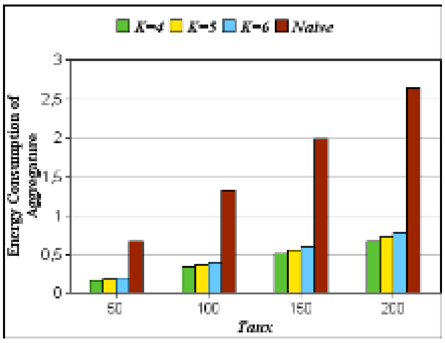


Fig. 10. Energy consumption in aggregator.

VI. CONCLUSION

Wireless sensor networks (WSNs) are one of the most advanced technologies nowadays. Therefore, researchers have paid great attention in the past years to this hot field by exploring the different challenges that cover it, while presenting different solutions. Unfortunately, WSN suffers from two major challenges: The big data collection that complicate the decision and data analysis, and the energy

REFERENCES

[1] Jacques M Bahi, Abdallah Makhoul, and Maguy Medlej. An optimized in-network aggregation scheme for data collection in periodic sensor networks. In International Conference on Ad-Hoc Networks and Wireless, pages 153–166. Springer, 2012.

[2] Abdallah Makhoul, Hassan Harb, and David Laiymani. Residual energy-based adaptive data collection approach for periodic sensor networks. Ad Hoc Networks, 35:149–160, 2015.

[3] Hassan Harb, Abdallah Makhoul, Rami Tawil, and Ali Jaber. Energy-efficient data aggregation and transfer in periodic sensor networks. IET Wireless Sensor Systems, 4(4):149–158, 2014.

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

[5] Ramesh Rajagopalan and Pramod K Varshney. Data aggregation techniques in sensor networks: A survey. 2006.

[6] Xiaoyan Kui, Jianxin Wang, Shigeng Zhang, and JLANNONG CAO. Energy balanced clustering data collection based on dominating set in wireless sensor networks. Adhoc & Sensor Wireless Networks, 24, 2015.

[7] Pinghui Zou and Yun Liu. A data-aggregation scheme for wsn based on optimal weight allocation. JNW, 9(1):100–107, 2014.

[8] M Shanmukhi and OBV Ramanaiah. Cluster-based comb-needle model for energyefficient data aggregation in wireless sensor networks. In Applications and Innovations in Mobile Computing (AIMoC), 2015, pages 42–47. IEEE, 2015.

[9] Tao Du, Zhe Qu, Qingbei Guo, and Shouning Qu. A high efficient and real time data aggregation scheme for wsns. International Journal of Distributed Sensor Networks, 11(6):261381, 2015.

[10] Yao Lu, Ioan Sorin Comsa, Pierre Kuonen, and Beat Hirsbrunner. Dynamic data aggregation protocol based on multiple objective tree in wireless sensor networks. In Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2015 IEEE Tenth International Conference on, pages 1–7. IEEE, 2015.

[11] Yung-Kuei Chiang, Neng-Chung Wang, and Chih-Hung Hsieh. A cycle-based data aggregation scheme for grid-based wireless sensor networks. Sensors, 14(5):8447–8464, 2014.

[12] Chih-Min Chao and Tzu-Ying Hsiao. Design of structure-free and energy-balanced data aggregation in wireless sensor networks. Journal of Network and Computer Applications, 37:229–239, 2014.

[13] Samuel Madden. http://db.csail.mit.edu/labdata/labdata.html. Intel Berkeley Research lab, 2004.

[14] Rupali Rohankar, CP Katti, and Sushil Kumar. Comparison of energy efficient data collection techniques in wireless sensor network. Procedia Computer Science, 57:146–151, 2015.