Real-Time Approach for Decision Making in IoT-Based Applications

Hassan Harb[1] [a], Diana Abi Nader[1], Kassem Sabeh[1] and Abdallah Makhoul[2] [b]

[1] *Faculty of Sciences, Lebanese University, Hadat, Lebanon*

[2] *FEMTO-ST institute, Univ. Bourgogne Franche-Comté, CNRS, 1 cours Leprince-Ringuet, 25200, Montbéliard, France*

hassan.harb.1@ul.edu.lb, abdallah.makhoul@univ-fcomte.fr

Keywords: IoT, wireless sensor networks, real-time applications, energy saving, data reduction techniques, score system, K-means algorithm.

Abstract: Nowadays, the IoT applications benefit widely many sectors including healthcare, environment, military, surveillance, etc. While the potential benefits of IoT are real and significant, two major challenges remain in front of fully realizing this potential: limited sensor energy and decision making in real-time applications. To overcome these problems, data reduction 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 energy efficient and real-time based algorithm to improve the decision making in IoT. At first data reduction is applied at the sensor nodes to reduce their raw data based on a predefined scoring system. Then, a second data reduction phase is applied at intermediate nodes, called grid leaders. It uses K-means as a clustering algorithm in order to eliminate data redundancy collected by the neighbor nodes. Finally, decision is taken at the sink level based on a scoring risk system and a risk-decision table. The evaluation of our technique is made based on a simulation from data collected on sensors at Intel Berkeley research lab. The obtained results show the relevance of our technique, in terms of data reduction and energy consumption.

1 INTRODUCTION

Wireless sensor network (WSN) is an indispensible part of Internet of Things (IoT). It allows monitoring, data collection, processing and transmission in several IoT based applications (Kadiravan et al., 2021) (Mostafa et al., 2018), such as smart cities and smart farms, health care, agriculture, business innovation, environmental monitoring, and so on (Atzori et al., 2017). In a typical scenario, data collected by sensor nodes are transmitted to a middleware for processing before being forwarded to a specific access point (sink) for further analyses and decision making (Kaur and Sood, 2015).

Following the increasing number of IoT users, the quantity of data generated and collected is continuously increasing. For that, differentiating between what data to keep and what data to dismiss is crucial for data accuracy and decision making optimization especially in real-time cases. However, data transmission is the most expensive in terms of sensor energy consumption. Additionally, IoT devices, especially sensor nodes, are limited in their lifetime energy due to the battery power limitation which affects the device's longevity (Ruan et al., 2019; Harb and Makhoul, 2018). Thus, many data reduction techniques have been introduced in order to solve this problem. It aims at removing similar routed data in such a way of minimizing the amount of data transmitted and thus, saving energy (Harb et al., 2018) (Harb et al., 2020).

In this paper, we propose a novel approach for energy saving and decision making in real-time IoT-based applications. Its purpose resides in reducing the quantity of transmitted data to the sink without loosing information and guaranteeing data integrity. By this mean, data transmission rate will be reduced, leading to the optimization of the network resources. Our technique is composed of two data reduction levels and one decision level. The first data reduction technique is applied at the sensor level where raw data are reduced by mean aggregation based on a predefined score system. The second data reduction technique is applied at intermediate nodes, called grid leaders. Each grid leader works on removing data redundancy collected by neighboring nodes based on K-means clustering algorithm. The final decision phase is achieved at the sink level. The dataset obtained will

[a] https://orcid.org/0000-0003-3066-9189

[b] https://orcid.org/0000-0003-0485-097X

be used to calculate the risk ratio based on the score system, then it will determine the right decision according to a predefined risk-decision table. To evaluate our approach several simulations have been conducted while using real dataset collected at the Intel Berkeley research lab. The obtained results show the relevance of our technique in terms of data reduction and energy consumption.

The rest of the paper is organized as follows. Section 2 overviews various existing data reduction techniques for IoT. Section 3 describes the real-time grid architecture used in our scenario. In Section 4, we present the first layer of our data reduction technique applied at the sensor nodes level. Section 5 describes the second layer of data reduction applied at the grid leaders level. In Section 6 we detail the decision making algorithm at the sin level. Simulation results are presented in Section 7. Finally, Section 8 concludes our paper and gives some perspectives.

2 Related Work

Many approaches have been suggested for data reduction in WSNs. The importance of reducing data collected in IoT based sensors relies in avoiding packets loss in network or delay in packets delivery which will affect decision making.

Many energy efficiency techniques based on machine learning techniques have been proposed for effective routing decisions in WSN and IoT. For instance, in (Thangaramya et al., 2019), neuro-fuzzy based cluster formation protocol (FBCFP) was proposed. The network is trained with convolutional neural network with fuzzy rules for weight adjustment, learning four components: current energy level of the CH (Cluster Head), distance of the CH from the sink node, change in area between the nodes present in the cluster and the CH due to mobility and the degree of the CH. Then using Mamdani Inference System for adjusting new CHs depending on energy. In (Preeth et al., 2018), another fuzzy energy efficient scheme is proposed called adaptive fuzzy multi criteria decision making (AF-MCDM) approach, where fuzzy Analytic Hierarchy Process (FAHP) and TOPSIS methods are combined together for selection of cluster head.

In (Santiago et al., 2018), an energy aware load balancing algorithm is proposed for IoT network by selecting parent nodes using event rate for cluster formation and neural network predictor for ELT (Expected Life Time) prediction. In (Elappila et al., 2018), a congestion and interference aware energy efficient routing technique in WSN for IoT is presented. It consists of selecting the signal to interference and noise ratio of the link, the survivability factor of the path from the next hop node to the destination, and the congestion level at the next hop node. In (Khan et al., 2018), a new routing protocol named Modified-Percentage LEACH Protocol is introduced based on existing protocol namely Percentage LEACH. Energy wastage is reduced by reducing communication between Cluster Heads and sink through threshold calculation on each CH for cluster selection and taking into consideration the distance of nodes from sink.

Authors in (Kaur and Sood, 2015) proposed to switch to sleep the IoT sensors based on three conditions: if it is not necessary to sense the target environment in a given period of time; if the coverage area can be compromised for battery life; and if the battery level is critically low. In (Hong et al., 2018), TCEB, a topology control algorithm with energy balance, addresses the problem of how to find a reasonably reduced topology and the packets forwarding route for underwater wireless sensor networks for energy conservation. Non-cooperative game theory is introduced to the the cluster-head selection in order to optimize the set of the cluster-heads selection and to ensure the energy consumption of the whole network balance. Then, the intra-cluster and the inter-cluster topology construction are used in order to find the optimum reliable nodes to perform the forwarding task.

On bio-smart levels, (Roy et al., 2019) a bio-inspired distributed event sensing and data collection framework was proposed based on the gene regulatory networks (GRNs) in living organisms. This is achieved by customizing a heuristic for the Maximum Weighted Independent Set problem encompassing both quality and quantity of sensed data, where the first depends on the device energy levels while the second on the number of events sensed. A sub-optimal device will be proposed depending on the residual energy.

In (Muhammad et al., 2019), an energy-efficient Data Prioritization framework is proposed by intelligent integration of the Internet of Things, artificial intelligence, and big data analytics for green smart cities. In (Ejaz et al., 2017; Tomasoni et al., 2018), a unifying framework for energy-efficient optimization and scheduling of IoT-based smart cities. Finally, in (Wang et al., 2017),an Integer Linear Programming (ILP) formulation as well as two effective polynomial-time heuristic algorithms are proposed for energy-efficient task scheduling problem on smartphones in mobile crowd sensing systems.

Despite that most of the proposed techniques allow efficient energy saving, they fail to reach all aspects of IoT applications. Indeed, they are very complex and require huge processing. In this paper, we present an energy efficient data reduction method that it is less complex and suitable for limited resources sensor nodes.

3 The Network Architecture

In this section, we introduce the network architecture used in our technique. Two main concepts have been adopted in our scenario: grid-based architecture and periodic data acquisition. In the following, we describe each of them in more details.

3.1 Grid-based Network

We consider a grid-based distribution, where each square can be defined based on the dimensions of the whole area of interest and the density of the sensor nodes. In each square, a grid leader is elected or selected using an appropriate method. This grid leader is considered as intermediate node. It can be an ordinary node or a node with more capabilities. We consider that the sensor nodes in the same square will send their data directly to their grid leader, which in his turn summarizes and aggregates the received data coming from neighboring nodes. The idea is to remove useless redundant data, While following specific rules and before sending them to the sink. Figure 1 shows our sensor network architecture, where data transmission between sensor nodes and their appropriate grid leader is based on single-hop communication.

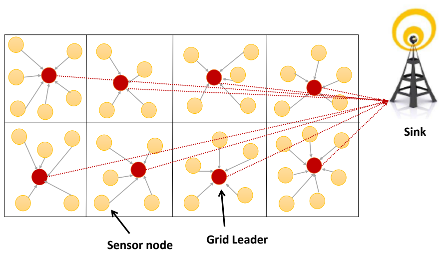

Figure 1: Two-level data transmission architecture.

3.2 Periodic Data Acquisition Model

Data collection from Iot applications can be made by data streaming where, data flows primarily from the sensor node to the grid leader and finally to the sink. We distinguish two kinds of data streaming the periodic sampling and the event driven. In our approach we are interested in ”periodic sampling” data model, where the acquisition of sensor data is forwarded to the grid leader and the to the sink on a periodic basis.

We consider that 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. In our system, each sensor node sends periodically (period p) its data to the appropriate grid leader, which in turn sends it to the sink (Fig. 2) (Harb et al., 2018).

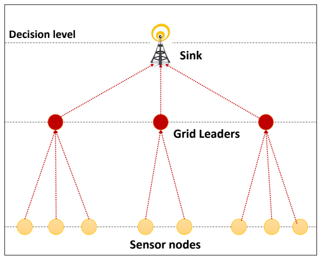

Figure 2: Periodic data reduction scheme.

4 First phase data reduction

In this section we describe how the sensor node itself reduces the amount of its collected data before sending it to the grid leader. Indeed, data collected in vector R contains redundant measurements especially when the slot interval between two collected measures is short. The idea is to reduce the size of the the vector R_i^p by selecting similar and consecutive measures from it and computing their mean to send to the grid leader instead of sending the whole readings. Our proposed model is based on a measurement score aggregation system used to identify similar measures.

4.1 Score System

The score system is a guide used to determine similar measures within a vector R_i^p by assigning scores to each measure. Based on a score system, all measures belonging to the same interval will be assigned the same score and aggregated as similar measures. Table 1 gives an example of such a score system.

4.2 Sensory data reduction Algorithm

In this section we introduce the algorithm proposed to reduce the number of readings collected periodically by each sensor. The idea is to reduce the number of readings in the vector R_i^p to send to the grid leader. The algorithm works as follows: first we find the

Table 1: Score System

Measure	Score
≤16	0
]16,17]	1
]17,18]	2
]18,19]	3
]19,20]	4
]20,21]	5
]21,22]	6
]22,23]	7
]23,24]	8
>24	9

score of each reading present in R_i^p using the function *score*, which takes the reading as an input and returns the score of the reading according to a well defined score system. Then, if consecutive readings in R_i^p have the same score, we calculate the mean value of those readings. The basic idea is that we consider readings having the same score as similar and redundant and instead of sending all of them we send only their mean. Therefore, the process starts by finding the score for the first reading in R_i^p, then it checks the score of the next values until it reaches a reading whose score is different from the current one. Then, we calculate the mean of all previous readings, and the process iterates again to find readings with same score as current value. Hence, the final vector $V_{R_i^p}$ contains the mean value of the readings having same score, as well as 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. Note that only consecutive values having same score are aggregated by their mean, and not all values within R_i^p having same score. The algorithm is represented in algorithm 1.

Algorithm 1 Sensor Mean Aggregation Algorithm.

Require: Reading vector: $R_i^p = [r_1, r_2, \dots, r_\tau]$.
Ensure: Vector of representative readings of R_i^p: $V_{R_i^p}$.

1: $V_{R_i^p} \leftarrow \emptyset$
2: $checkbefore = Score(r_0)$ // score of first reading
3: $frequency = 0$
4: **for** each set reading $r_i \in R_i^p$ **do**
5: $\quad check = Score(r_0)$ // score of current reading
6: $\quad$ **if** $checkbefore = check$ **then**
7: $\quad\quad frequency = frequency + 1$
8: $\quad$ **else**
9: $\quad\quad$ find the mean value, $\bar{r}_i$, of readings in R_i^p
10: $\quad\quad wgt(\bar{r}_i) = frequency$
11: $\quad\quad V_{R_i^p} \leftarrow V_{R_i^p} \cup \{\bar{r}_i, wgt(\bar{r}_i)\}$
12: $\quad\quad frequency = 1$
13: $\quad$ **end if**
14: $\quad checkbefore = check$
15: **end for**
16: return $V_{R_i^p}$

After applying Algorithm 1, each sensor will send a vector of representative readings $V_{R_i^p} = \left[(\bar{r}_1, f_1), (\bar{r}_2, f_2), \dots, (\bar{r}_k, f_k)\right]$ to its proper grid leader, where $k \leq \tau$ and f_i represents the weight or frequency of $\bar{r}_i$.

5 Second phase data reduction

The grid leader receives a vector of reduced data from all neighboring sensors at the end of each period. At this stage, we propose an algorithm to reduce the number of data collected at grid leader level by eliminating similar or redundant vectors. Our objective is to group the readings observed by the grid leader in clusters using the K-means algorithm, then eliminate redundant data within each cluster by applying a mean based technique similar to the one applied at sensor level. In the next sections we explain in more details how we combined the *K*-means algorithm with a mean aggregation technique to clean data at grid leader level.

5.1 *K*-Means Clustering Algorithm

K-means clustering is a simple unsupervised learning algorithm that is used to solve clustering problems. *k*-means tries to divide a set of samples in k disjoint groups or clusters using the mean value of the members as the main indicator. The clusters are partitioned as points and all observations or data points are associated with the nearest cluster, computed, adjusted, and then the process starts over using the new adjustments until a desired result is reached. Although it can be proved that the procedure will always terminate, the algorithm is also significantly sensitive to the initial randomly selected cluster centers.

The performance of a clustering algorithm may be affected by the chosen value of *K*. Therefore, instead of using a single predefined *K*, a set of values might be adopted. It is important for the number of values considered to be reasonably large, to reflect the specific characteristics of the data sets. At the same time, the selected values have to be significantly smaller than the number of objects in the data sets, which is the main motivation for performing data clustering. The optimal value of K for many studied applications varied in the interval $[3,5]$.

5.2 Absolute Value Distance

One of the fundamental steps when applying the K-means algorithm is computing the distance between a reading and a mean value of a cluster. In this paper, we considered the absolute value distance which is a simple yet effective method to find the distance between two values. In mathematics, the distance between x and y is defined by a subtracting relationship: $\mid x-y \mid$. Let us consider a data value in R_i and mean value of a cluster μ , then the absolute value distance (A_d) between them can be calculated as follows:

$$A_d(r_i,\mu) = \mid \mu - r_i \mid, \qquad (1)$$

where $r_i \in R_i$ and μ is the mean value of a cluster.

5.3 *K*-Means and Absolute Value Distance

Algorithm 2 describes the procedure of K-Means algorithm to divide the n vectors R_i received by the grid leader from the n neighboring sensors into a set of k clusters. First, the algorithm assigns k random values from the readings as centroids and allocates a cluster for each centroid. Then, we calculate the absolute value distance between each value r_i and the k centroids of all clusters and assign the readings to the clusters according to the minimum distance. In the next step, we calculate the mean value for each cluster and use it as the new centroid for that cluster. Then, we calculate the distance between the new centroid and the old one. If the distance is zero for all clusters, the algorithm stops and we return the clusters with the corresponding centroid of each. Else, we iterate and recalculate the distance between the readings r_i and all centroids to allocate them to the one with the minimum distance. The algorithm stops when the distance between the mean and current centroid is zero in all clusters.

Algorithm 2 K-Means Adopted to Absolute Value Distance Algorithm.

Require: List of datasets $R^p = \{R_1^p, R_2^p, \dots, R_n^p\}, K$,
  where $R_i^p = \left[(\bar{r}_1,f_1),(\bar{r}_2,f_2),\dots,(\bar{r}_k,f_k)\right]$
Ensure: List of k clusters $C^p = \{C_1^p, C_2^p, \dots, C_k^p\}$
  where $C_i^p = \left[(\bar{r}_1,f_1), value(\bar{r}_2,f_2),\dots,(\bar{r}_k,f_k)\right]$.

for $j \leftarrow 1$ to k **do**
  $C_j \leftarrow \phi$
  and randomly choose centroid x_j belongs to C_j
end for
repeat
  for each value $\bar{r}_i \in R_i^p$ **do**
    Assign $\bar{r}_i$ to the cluster C_j with nearest x_i
  end for
  for each cluster C_j, where $j \in 1,\dots,k$ **do**
    Update the centroid x_i to be the centroid of all values currently in C_j so that $x_j = \frac{1}{|C_j|}\sum \bar{r}_i$ for i $\in C_j$
  end for
until
clusters' centroids no longer changes
return C^p

5.4 Cluster Reduction at the Grid Leader

After applying the K-Means algorithm to the data received by the grid leader we will have K-clusters. Next, we will reduce the data present in each cluster by utilizing the score system we defined earlier (cf. Section 4.1).

For the K clusters found in (5.3) we will aggregate the values having the same score using their mean value. This approach is similar to the one we initially performed in (4.2) but this time we use the frequency to calculate the mean value. So for each cluster $C_i^p = \left[(\bar{r}_1,f_1),(\bar{r}_2,f_2),\dots,(\bar{r}_k,f_k)\right]$, if n consecutive values $\bar{r}_i, \bar{r}_{i+1}, \dots, \bar{r}_{i+n}$ have the same score according to our score system, we aggregate them by calculating the mean value:

$$\overline{m}_j = \frac{\bar{r}_i * f_i + \bar{r}_{i+1} * f_{i+1}, + \cdots + \bar{r}_{i+n} * f_{i+n}}{|\overline{m}_i|} \qquad (2)$$

At the end of this phase we will have a set of K clusters $C^p = \{C_1^p, C_2^p, \dots, C_k^p\}$ where $C_i^p = \left[(\overline{m}_1,f_1),(\overline{m}_2,f_2),\dots,(\overline{m}_i,f_i)\right]$, where f represents the frequency value used to compute the mean and i is ≤ 10 (10 being the number of scores available). This data will be sent to the sink for a decision to take place as illustrated in the next section.

6 Decision making method at the Sink

The sink receives the data from the grid leader as clusters and stores this data after aggregating it, in a single cluster form $D = \left[(\overline{m}_1,f_1),(\overline{m}_2,f_2),\dots,(\overline{m}_n,f_n)\right]$. Then it makes a decision based on this data and the evaluated risk as shown in Table 2.

Table 2: Risk Decision

Risk	Decision
≤ 0.2	d_1
$]0.2, 0.4]$	d_2
$]0.4, 0.6]$	d_3
$]0.6, 0.8]$	d_4
> 0.8	d_5

To calculate the risk, first we have to find the frequency associated with each score. We then transform the aggregated data set D into the following form according to our score system: $D_c = [(0, freq_0), (1, freq_1), (2, freq_2), \ldots, (9, freq_9)]$.

Where the numbers $0, 1, 2, \ldots, 9$ represent scores from our score system in Table 1 while $freq_i$ represents the frequency of values having a score i.

Furthermore, the weight parameter is calculated according to the following formula:

$$weight_i = \frac{freq_i * i}{|D_c|} \tag{3}$$

where $|D_c|$ is the total frequency of the set D_c.

Finally, the risk is obtained according to the formula:

$$\frac{0 * weight_0 + 1 * weight_1 + 2 * weight_2 + \cdots + 9 * weight_9}{6}. \tag{4}$$

Based on the risk level, the decision will be taken according to Table 2.

7 Simulation Results

In this section, we show the relevance of our proposed technique after performing a simulation on real sensor data nodes deployed in Intel Berkeley research lab. 47 sensors were deployed where each of them collected temperature, humidity, light and voltage data from 28 February to 5 April 2004. A total of 2.3 millions readings were collected (Harb et al., 2015). In these series of simulation, we tested our algorithm while considering the temperature values. The sensors send their collected data to a grid leader where the K-means algorithm will be applied. The value of k is set first to 3, then to 4 and finally to 5. The period size in its turn is set first to 50 readings, then 100 readings and finally to 150 readings. Figure 3 shows the distribution of sensors inside the laboratory.

In these simulations Table 1 is adopted for the score system.

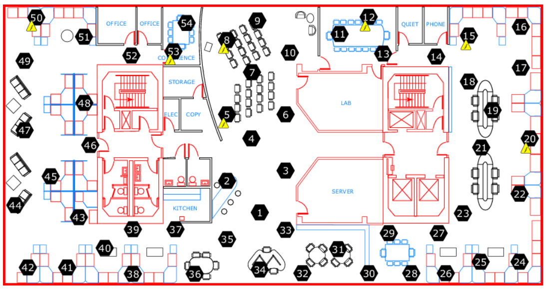

Figure 3: Distribution of sensors in Intel Berkeley lab.

7.1 Data Aggregation Ratio at Sensor nodes

As mentioned before, the mean aggregation according to the score system, allows each sensor node to minimize the size of its sensed data by removing similar readings. Figure 4 shows the average number of temperature readings implemented on different size periods, using our technique. The obtained results show that our data filtering model allows sensors to significantly reduce its data transmission in a redundant zone, up to 92% the temperature readings for $\tau = 50$.

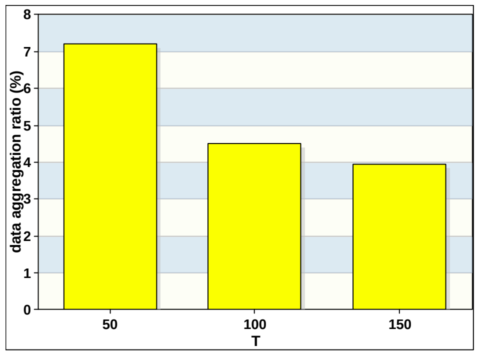

Figure 4: Aggregation ratio at each sensor during different periods.

7.2 Data Aggregation Ratio at Grid Leader

In Figure 5, we show the average number of remaining sets after applying K-means algorithm at the grid leader, when varying K values to 3, 4 and 5 respectively. The obtained results show that K-means can significantly eliminate redundant data sets generated by neighboring sensors. Subsequently, we observe that K-means can reduce up to 36% of the whole received sets at the grid leader. These results confirm that the clustering is a very efficient approach in terms of eliminating redundant data and providing useful information to the end user, comparing to other existing approaches.

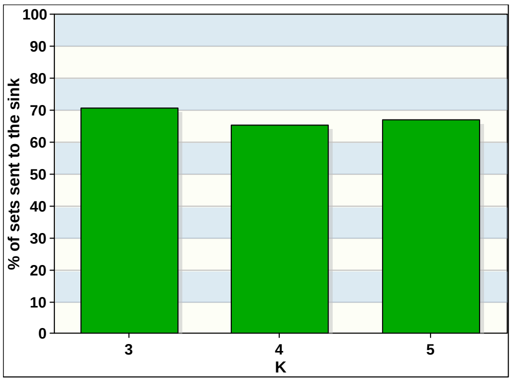

Figure 5: Filtering set ratio after applying K-means at the grid leader.

7.3 Energy Consumption at Sensors

Energy consumption is the main metric being studied in this research. Figure 6 shows the average consumption of energy at sensor level by studying the decrease in the average energy remaining in sensors as a function of rounds. We suppose that the initial energy for each sensor is $50mJ$ and run the simulation until the energy becomes zero. A round consists of 15 periods. We can observe that the sensors survive for approximately 160 rounds when $\tau = 50$, and 100 rounds when $\tau = 150$.

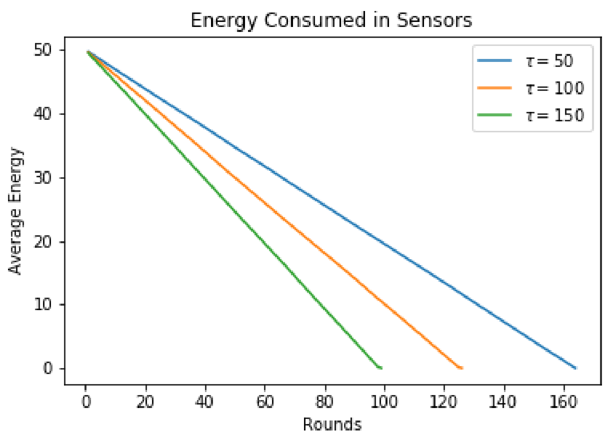

Figure 6: Average energy consumption at sensors level.

7.4 Energy Consumption at Grid Leader

Figure 7 shows the results of energy consumption of K-means, when varying K. The obtained results are highly dependent on the number of remaining sets after applying K-means (see results of Figure 5); less the number of remaining sets thus less of energy is lost. Indeed, we observe that our technique gives important results regarding reducing the energy consumption of the sent data.

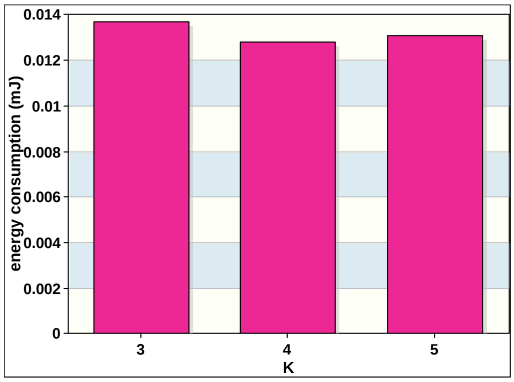

Figure 7: Percentage energy consumption.

8 Conclusion and Future Work

Internet of Things (IoT) will play an important role in the future by collecting surrounding conditions and environment information. Thus, designing new energy efficient techniques while preserving data integrity for decision making appears to be crucial in order to eliminate redundant raw data and make such networks operate as long as possible. This paper proposed a real-time framework for energy-efficient and decision making in IoT-based applications. The first phase uses a score aggregation system and aims to reduce raw data collected by the sensors. The second phase allows grid leader nodes to eliminate redundant data collected by neighboring nodes using K-means clustering algorithm. The third phase will be dedicated to decision making based on a risk-decision table. Our technique has been evaluated based on a simulation on real sensors data. The results obtained with our technique showed significant energy savings.

Many future directions for our work can be traced. We seek to try another data clustering methods at the grid leader level, like decision trees and neural networks.

REFERENCES

Atzori, L., Iera, A., and Morabito, G. (2017). Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. *Ad Hoc Networks*, 56:122–140.

Ejaz, W., Naeem, M., Shahid, A., Anpalagan, A., and Jo, M. (2017). Efficient energy management for the internet of things in smart cities. *IEEE Communications Magazine*, 55(1):84–91.

Elappila, M., Chinara, S., and Parhi, D. R. (2018). Survivable path routing in wsn for iot applications. *Pervasive and Mobile Computing*, 43:49–63.

Harb, H., Jaoude, C. A., and Makhoul, A. (2020). An energy-efficient data prediction and processing approach for the internet of things and sensing based applications. *Peer-to-Peer Netw. Appl.*, 13(3):780–795.

Harb, H. and Makhoul, A. (2018). Energy-efficient sensor data collection approach for industrial process monitoring. *IEEE Trans. Ind. Informatics*, 14(2):661–672.

Harb, H., Makhoul, A., and Couturier, R. (2015). An enhanced k-means and anova-based clustering approach for similarity aggregation in underwater wireless sensor networks. *IEEE Sensors Journal*, 15(10):5483–5493.

Harb, H., Makhoul, A., and Jaoude, C. A. (2018). En-route data filtering technique for maximizing wireless sensor network lifetime. pages 298–303.

Hong, Z., Pan, X., Chen, P., Su, X., Wang, N., and Lu, W. (2018). A topology control with energy balance in underwater wireless sensor networks for iot-based application. *Sensors*, 18(7):2306.

Kadiravan, G., Sujatha, P., Asvany, T., Punithavathi, R., Elhoseny, M., Pustokhina, I., Pustokhin, D. A., and Shankar, K. (2021). Metaheuristic clustering protocol for healthcare data collection in mobile wireless multimedia sensor networks. *Computers, Materials & Continua*, 66(3):3215–3231.

Kaur, N. and Sood, S. K. (2015). An energy-efficient architecture for the internet of things (iot). *IEEE Systems Journal*, 11(2):796–805.

Khan, F. A., Ahmad, A., and Imran, M. (2018). Energy optimization of pr-leach routing scheme using distance awareness in internet of things networks. *International Journal of Parallel Programming*, pages 1–20.

Mostafa, B., Benslimane, A., Saleh, M., Kassem, S., and Molnar, M. (2018). An energy-efficient multiobjective scheduling model for monitoring in internet of things. *IEEE Internet of Things Journal*, 5(3):1727–1738.

Muhammad, K., Lloret, J., and Baik, S. W. (2019). Intelligent and energy-efficient data prioritization in green smart cities: Current challenges and future directions. *IEEE Communications Magazine*, 57(2):60–65.

Preeth, S. S. L., Dhanalakshmi, R., Kumar, R., and Shakeel, P. M. (2018). An adaptive fuzzy rule based energy efficient clustering and immune-inspired routing protocol for wsn-assisted iot system. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13.

Roy, S., Ghosh, N., and Das, S. K. (2019). biosmartsense: A bio-inspired data collection framework for energy-efficient, qoi-aware smart city applications. pages 1–10.

Ruan, J., Wang, Y., Chan, F. T. S., Hu, X., Zhao, M., Zhu, F., Shi, B., Shi, Y., and Lin, F. (2019). A life cycle framework of green iot-based agriculture and its finance, operation, and management issues. *IEEE communications magazine*, 57(3):90–96.

Santiago, S., Kumar, A. D. V., and Arockiam, L. (2018). Ealba: energy aware load balancing algorithm for iot networks. pages 46–50.

Thangaramya, K., Kulothungan, K., Logambigai, R., Selvi, M., Ganapathy, S., and Kannan, A. (2019). Energy aware cluster and neuro-fuzzy based routing algorithm for wireless sensor networks in iot. *Computer Networks*, 151:211–223.

Tomasoni, M., Capponi, A., Fiandrino, C., Kliazovich, D., Granelli, F., and Bouvry, P. (2018). Profiling energy efficiency of mobile crowdsensing data collection frameworks for smart city applications. pages 1–8.

Wang, J., Tang, J., Xue, G., and Yang, D. (2017). Towards energy-efficient task scheduling on smartphones in mobile crowd sensing systems. *Computer Networks*, 115:100–109.