Karbala International Journal of Modern Science

Volume 7 | Issue 4 Article 9

Energy Conservation Approach of Wireless Sensor Networks for IoT Applications

Suha Abdulhussein Abdulzahra MSc
Department of Dentistry, Al-Mustaqbal University College, Babylon, Iraq

Ali Kadhum M. Al-Qurabat Ph.D.
Dept. of Computer Science, College of Science for Women, University of Babylon, Babylon, Iraq, alik.m.alqurabat@uobabylon.edu.iq

Prof. Dr. Ali Kadhum Idrees
Dept. of Computer Science, College of Science for Women, University of Babylon, Babylon, Iraq

Recommended Citation

Abdulzahra, Suha Abdulhussein MSc; Al-Qurabat, Ali Kadhum M. Ph.D.; and Idrees, Prof. Dr. Ali Kadhum (2021) "Energy Conservation Approach of Wireless Sensor Networks for IoT Applications," *Karbala International Journal of Modern Science*: Vol. 7 : Iss. 4 , Article 9.
Available at: https://doi.org/10.33640/2405-609X.3160

Energy Conservation Approach of Wireless Sensor Networks for IoT Applications

Abstract

Wireless Sensor Network is one of the most important contributors to IoT and performs significant role in people's lives due to its extensive use in many applications. Energy-saving is essential since sensor nodes are working by their restricted battery. In this article, data reduction method proposed to work at Gateway level of network. In GW, proposed method works as filtering via enabling GW to identify, then remove, sets of data that are redundant and produced by neighboring nodes. Principle idea of method recommended at this level is to exploit the advantage of spatial correlation between sensors to minimize energy depletion.

Keywords

Wireless Sensor Networks (WSNs), IoT, Energy-saving, Leader clustering, Network lifetime

1. Introduction

With a growing need to best understand our environments, the Internet of Things (IoT) is gaining popularity between communication and information technologies [1]. One of the most significant IoT contributors is the Wireless Sensor Network (WSN) which performs a significant role in people's lives due to its extensive use in many applications such as disaster detection/prevention, industry, transportation, agricultural/industrial/home automation, healthcare, military and so on. This type of network occurs because of the rapid growth of embedded computing technology, wireless devices and electronics [2].

A huge amount of lightweight, cheap, and low-energy sensors are included in the WSN that are distributed spatially to track the physical or environmental phenomena of a particular region of attention like temperature, humidity, sound, vibration, strain, motion, etc. [3]. Sensor nodes that are characterized by limited resources such as energy (battery power), processing and storage capacity, reliability and range of wireless communications, can sense, process, store and communicate, moreover, their dissemination must cover a large geographical region. The sensors capture data from the tracked region, manipulate it, and send it to one or more collection points called gateway (GW) (i.e., which acts as an interface between the sensor nodes and users) for further analysis [4].

The energy supplied by the battery is the most critical resource in the sensor node affecting WSN's lifetime. The key role of the power unit (battery) was to provide the energy necessary to accomplish the mission of the sensor node [5]. Therefore, energy-saving is essential since sensor nodes are working by their restricted battery and if a vast number of sensors are spread over a wide space or spread in a harsh or hostile area such as in the deep sea or around volcanoes when the battery expires, it could be uncomfortable or very hard to exchange or recharge it [6].

The key cause of the waste of the sensor nodes' energy is the radio system. Therefore, there are several strategies and concepts that are based on saving power, particularly focusing on reducing the sending of data, like the compression of data, battery replenishment, scheduling, aggregation, predictive tracking, routing, optimization of radio and clustering [7]. Data compression is excellent for applications that really don't require real-time data, and it is particularly beneficial when sensor nodes must continually communicate data measurements to the GW over a lengthy time period [8].

Based on application requirements, data collection in WSN could be event-driven (for instance detection of forest fires and gas or oil leaks) and time-driven (such as ecosystem monitoring, temperature and humidity logging in a canopy of precision growing potato plants). This article considers the model of a time-driven collection of data called Periodic. Each sensor node periodically sends the sensed data from the surveillance area to the GW [1]. The principal contributions in this article focus on designing and implementing energy-efficient data reduction techniques to improve the lifetime of WSNs.

The contributions made by this article are as follows:

1. At the gateway level, we proposed a method that works as filtering via enabling the GW to identify, then remove, sets of data that are redundant and produced by neighbouring nodes, to cut down on the number of total sets that the sink will receive.
2. The suggested approach is evaluated with the aid of the OMNeT++ network simulator by extensive simulation experiments. The efficiency of the proposed technique is evaluated with two related works: the PFF protocol proposed in Ref. [9] and the AVMDA protocol proposed in Ref. [10].

For the rest of this article, the following are structured: the related data reduction works are presented in Section 2. The proposed data reduction at the GWs is shown in Section 3. Section 4 depicts the simulation experiments and discusses the outcomes, while Section 5 summarizes the findings and suggests further research.

2. Related works

The main aim of this review is to thoroughly examine published works of literature on extending the life span of IoT sensor networks using data compression/reduction approaches. Although many of the previous works have evaluated the results of the compression techniques used in their work, only a few have evaluated these methods from the perspective of sensor networks. Most methods focus on the compression ratio, while sensor nodes in the IoT

https://doi.org/10.33640/2405-609X.3160

require a focus on other metrics in addition to the compression ratio such as energy, accuracy and other resource requirements [11]. There should also be a high compression ratio for the compression algorithm operating on sensor nodes that decreases both the number of bits sent and the ratio of energy consumption. Many resource-conscious compressions techniques have been developed and used in WSNs to minimize data [12].

In [13] a technique known as Lifting Wise was suggested. The Lifting Wise technique is an adjusted version of the original Discrete Wavelet Transform (DWT) Lifting Scheme (LS) algorithm and it can be used on a variable-length range of data whereas the original LS is used on a signal Sn with length 2^n. This method has been utilized to process the data spread from objects disseminated in a monitoring environment. It was compared to two other basic techniques of compression suitable for utilization in WSNs: Offset compression and Marcelloni compression [14]. This method is implemented on the sensor node and it cannot be implemented efficiently on the gateway, where the spatial correlation is not considered in this paper. The findings demonstrate the efficacy of this approach in decreasing the number of bits obtained from the data gathered by considering the finite resources of sensor nodes. In Ref. [15], the authors suggested a compression algorithm for spatial−temporal data from one data form of a sensor node in a WSN installed in an underground tunnel. The suggested algorithm operates effectively as it considers the sensor data's both temporal and spatial characteristics. A recovery method is used for data recovery with a near approximation to the initial node's data. This method provides a high complexity during compressing the data; therefore, it is difficult to consider it on the limited resources nodes in the IoT network.

In [9], the authors explored the problem of filtering collection using the Prefix-Frequency Filtering (PFF) technique. The aggregation is executed in this technique at two phases: the sensor phase utilizing local processing of data as well as the aggregator phase employing PFF with the Jaccard similarity mechanism to consolidate similar data from the nearby nodes in one record. This method did not provide competitive results in reducing the redundant data before sending it to the base station. In Ref. [16], the authors proposed a lightweight compression algorithm for WSNs that tracks low-resolution sensor environmental parameters. In consideration of the general understanding of the parameters to be tracked, when calculating the same parameter at several places and times, contemporary Huffman compression may be utilized effectively to describe it. The Huffman dictionary, calculated using statistics derived from public databases, frequently reaches the entropy of the system when the data obtained by the sensor nodes comprises integer measurements. The results provided in this method show that this method cannot reduce the huge data efficiently. In Ref. [17], the authors introduced an adaptive lossless data compression algorithm (ALDC) for WSNs. The proposed ALDC algorithm incorporates several coding alternatives to achieve lossless compression. The adaptive compression method allows dynamic adjustment of the compression to a changeable source. To compress the data series, it's first divided into blocks, and for each block, the optimal compression method is implemented. This method increases the time complexity and it is not suitable to work on the gateway.

A simple lossless compression technique for real-time sensor networks was proposed by the authors in Ref. [18]. The proposed linear adaptive filtering compression (ALFC) algorithm uses linear adaptive filtering to forecast sample values, as well as entropy coding of prediction residuals, to compact a variable number of samples into fixed-length packets. Adaptive prediction reduces the need to evaluate prediction coefficients a priori and, more importantly, allows compression to be dynamically adjusted to an evolving source. The algorithm only uses integer arithmetic operations, making it compatible with sensor systems that don't support floating-point calculations. This proposed approach cannot provide a competitive data reduction rate compared with the new methods. In Ref. [19], the authors introduced a new, effective and reliable lossless data compression algorithm named Sequential Lossless Entropy Compression (S-LEC), which can greatly boost the efficiency of temporal lossless compression for data collections in WSNs. The developed algorithm improves the LEC to resolve its limitation due to inadequate robustness, allowing extremely robust compression efficiency for specific sensor streaming data to be achieved. This algorithm can be implemented on resource-restricted WSN nodes, but it is not efficient on the gateway nodes.

In our previous work [20], we proposed the use of a new “Data Reduction Based on Compression Technique (DRCT)" to compress IoT data readings at the sensor level, effectively saving power, reducing data transfer volume, and maintaining the accuracy of the data readings received on the gateway, thereby prolonging the duration of the IoT network. DRCT is a two-stage compression system that starts with a lossy quantization based on the SAX technique for decreasing the measurements range of the sensor node while increasing the number of frequent patterns of data and then moving on to an LZW compression stage to compress the SAX output in a lossless way.

3. The proposed data reduction technique at the GW

In this article, a new data reduction technique works at the GW was proposed which complements the previously proposed method [20]. This technique was developed based on cluster topology. In the proposal, the formation of the cluster topology is out of scope, it is assumed that there is already a topology and deliberately skipping to discuss the formation of the topology. The proposed techniques can be applied to these clusters produced by any clustering protocol. Focusing basically on designing energy-efficient data reduction techniques. More precisely, the objective of this technique is to reduce the sensed data at the GW level to prolong the WSN's lifetime. At the level of the sensor node, for restricted IoT sensor nodes, we proposed in our previous work [20] an easy and appropriate method of data compression. At the gateway level, the proposed method works as filtering via enabling the GW to identify, then remove, sets of data that are redundant and produced by neighbouring nodes, to cut down on the number of total sets that the sink will receive. Fig. 1 depicts the flow chart of the proposed approach.

Any cluster in a cluster-based model has many cluster members (sensor nodes) and a unique cluster head (i.e., GW). Usually, when the sensor nodes are scattered in close regions, they can generate the same readings or similar readings, and therefore spatial correlation can be exploited. After each period, the sensor node's data set is delivered to the GW that corresponds to it. In its role, the GW tries to minimize the sets of data received before they are transmitted to the sink. Despite the likelihood that the first level of compression has decreased the duplication of the data collected at each sensor node, there is still the likelihood of overlap (i.e., similarity) between sets of data from different sensor nodes. In the following sections, a detailed description of the proposed methods is given.

3.1. Data sets duplication

In our previous work [20], when the period ends, the nodes transmit their compressed sets of data in addition to the corresponding standard deviation and average of each set to the GW to which they belong. Where the corresponding standard deviation and average are used in the process of decompressing the data set. We must determine whether the data sets are close to each other or not, based on the returned values from Euclidean distance. If the similarity is strong (i.e., below or equal threshold δ), we'll consider pairs of sets as repeated and reduce the number of total sets of data we send to the sink.

One of the simplest ways to locate all pairs of similar sets of data is to list and compare each of the sets. This approach is viewed as a naïve solution. Nonetheless, it is quite costly computationally to apply the distance measure based on Euclidean to every pair of sets because the comparison number it produces is $O(n^2)$, in which n denotes the number of obtained sets. Moreover, the calculation for massive data sets would be more complicated, as in the case of large sensor networks. As a result, checking the pairings of duplicate sets is required to minimize comparisons number and speed up the removal of redundancy.

This check will take place in two steps. In step one, we suggest dividing the sets of data into clusters depending on the corresponding average of each set. Where each cluster will contain a list of “candidates” pairs to be similar. Some conditions if fulfilled will make the pair a candidate and this implies that the two sets that make up this pair are perhaps similar. A pair isn't a candidate, though, meaning it's not similar certainly. In step two we are using a similarity measure to confirm that we select the best set of data per each cluster.

3.2. Clustering candidate sets

Conducting pair comparisons in enormous amounts of data (like in the situation of WSN) is time-consuming since the comparison number rises linearly with the data volume. The cause of comparisons reduction is that many of these comparisons may not lead to a match and are therefore considered superfluous. The major purpose of this step is to group the data sets into clusters based on the average of them in order to decrease the number of comparisons and speed up the redundancy removal process. The goal of this clustering is, for each cluster, to contain the most likely comparable sets of data. In this method, the GW can avoid making distinctions for all sets that come to it in search of candidate pairings. For clustering the receiving data sets from level one, we recommend utilizing the Leader method [21] in the GW, as illustrated in Algorithm 1.

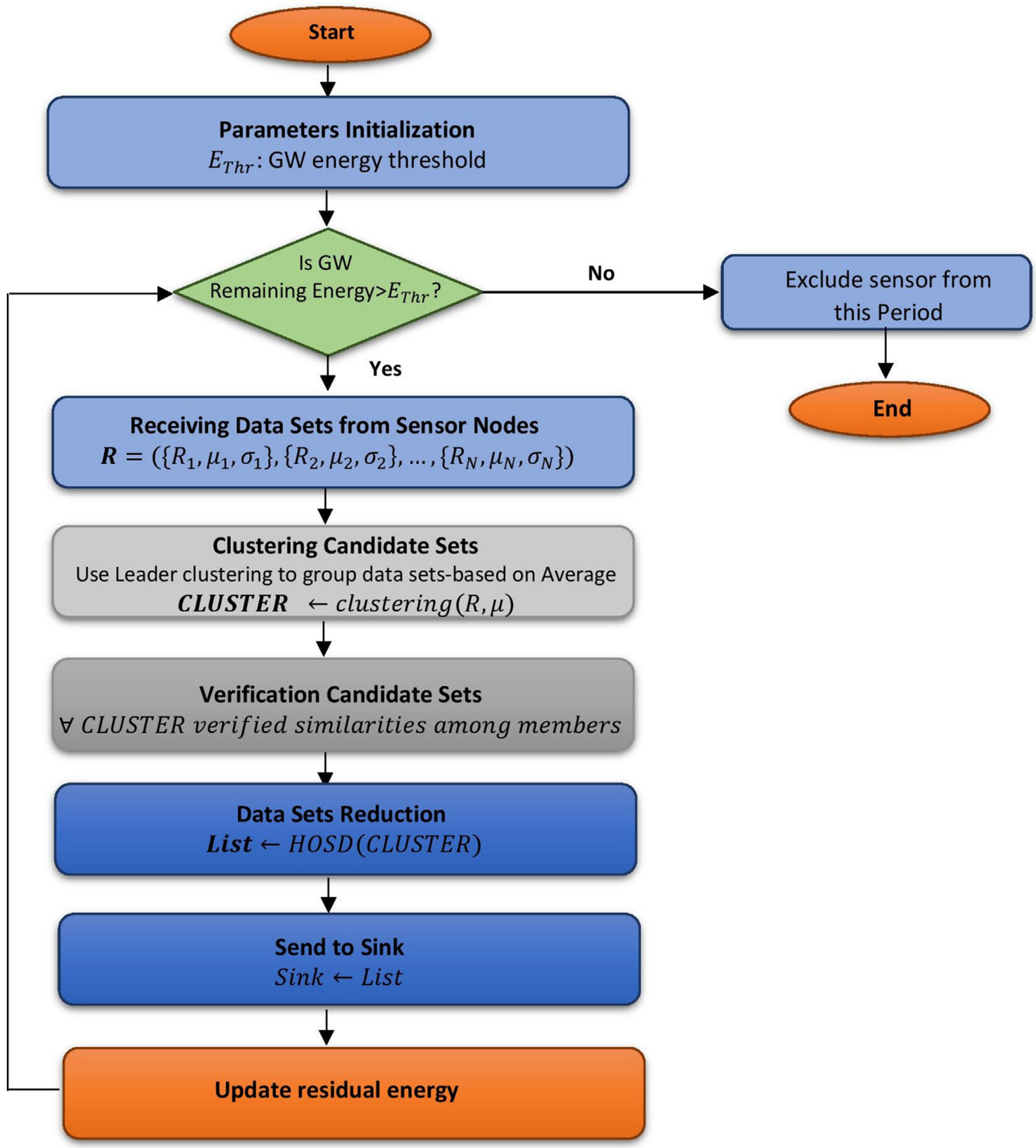

Fig. 1. Flow chart of the proposed approach.

A leader is an incremental clustering procedure that is presented in the following way:

1. Cluster number one contains data set number one.
2. Calculate the distance between the data set number two and the cluster number one (i.e., between averages).
   i. If the distance is below or equal to the threshold, data set number two is placed in cluster number one.
   ii. Or alternatively, create and insert the number two data set into a new cluster.
3. We'll keep measuring the distance between the data set number three and all of the current clusters.
   i. The data set number three is set in the cluster number i^{th}, when the distance between them is the shortest and that distance is less than or equal to the threshold.
   ii. Or alternatively, create and insert the number three data set into a new cluster.

This approach will proceed till all the incoming data sets in the GW have been taken into consideration.

The time complexity of Algorithm 1 is $O(n\ .\ sizeof(Sj))$, where $sizeof(Sj)$ refers to the size of the data set Sj and n is the number of data sets in S.

Algorithm 1: Leader-based Data sets Clustering

Input: N sensor nodes: $S = \{S_1, S_2, \ldots, S_n\}$, set of averages: $\mu = \{\mu_1, \mu_2, \ldots, \mu_n\}$, distance threshold Th_d

Output: Cluster: set of clusters

```
j = 1
while (j < N) Do
   if (j = 1) then
      Add S_j to Cluster[1]
   else
      Calculate EUC_D(S_j(μ), All C_i(μ) from Cluster) //C_i:the i-th cluster
      if (EUC_D ≤ Th_d) then
         Add S_j to Cluster[i]
      else
         Create New Cluster[No]          // No: index of current cluster
         Add S_j to Cluster[No]
      endif
   endif
   j++
endwhile
return Cluster
```

3.3. Verification candidate sets

During this step, each GW determines the pairs of sets (set of data readings after converted to the original representation (i.e., after decompressed each data set) that are "candidate for similarities". We assume that when the distance between the average two sets of data readings is below the Th_d threshold the distance between the two sets of data readings is a nominee to be below Th_d. So, we prove in our study that two compressed sets of data reading D_i and D_j, if the difference over their corresponding averages is smaller than Th_d, they are candidates as demonstrated by the lemma below:

Lemma: Take into account two compressed sets of data readings D_i and D_j and their respective averages μ_i and μ_j. Assume that R_i and R_j are D_i and D_j's original data readings (i.e., raw data without compression). As a result, if the Euclidean distance between R_i and R_j is less than Th_d, the distance between their respective averages should likewise be less than Th_d. Therefore:

$$EUC_D(R_i, R_j) \leq Th_d = \delta \times \sqrt{n} \tag{1}$$

where δ is a user-defined value and n is the length of R. Then

$\boldsymbol{D_i}$ and $\boldsymbol{D_j}$ are considered candidates $\Leftrightarrow |\mu_i - \mu_j| \leq Th_d$.

Proof: First, we define a threshold between the readings in R_i. Two readings are considered similar if their difference is less than the defined threshold as follows:

$$\|r_i - r_j\| \leq \delta \tag{2}$$

Let two sets of data readings R_i and R_j with the same length n. The Euclidean distance between them is calculated as follows:

$$EUC_D(R_i, R_j) = \sqrt{\sum_{i=1}^{n} (r_i - r_j)^2} \leq Th_d \tag{3}$$

where $r_i \in R_i$ and $r_j \in R_j$. Consider that R_i is similar to R_j after searching similarity readings in R_i. Thus, we have:

$$EUC_D(R_i, R_j) = \sqrt{\sum_{i=1}^{n} (r_i - r_j)^2} \leq \sqrt{\sum_{i=1}^{n} \delta^2} \tag{4}$$

$$EUC_D(R_i, R_j) = \sqrt{\sum_{i=1}^{n} (r_i - r_j)^2} \leq \sqrt{n \times \delta^2} \tag{5}$$

$$EUC_D(R_i, R_j) = \sqrt{\sum_{i=1}^{n} (r_i - r_j)^2} \leq \delta \times \sqrt{n} \tag{6}$$

Thus R_i and R_j should be considered similar only if the Euclidean distance between them does not exceed $Th_d = \delta \times \sqrt{n}$.

3.4. Data sets reduction

After, we completed the clustering of compressed data sets and verified their similarity in each cluster using the similarity scale, as shown in the previous steps. Now we will show how the GW reduces the total amount of sets of data that have been transmitted to the sink to reduce energy consumption and maintain the life of the network for the longest possible time in addition to ensuring acceptable data accuracy.

Here, we propose a strategy called Hold One Set of Data and Delete Other Sets (HOSD). The GW removes duplicate sets of data sent from neighbouring nodes in each cluster after recognizing all pairs of duplicated sets, to reduce the number of transmitted data to the sink while preserving information integrity. In this strategy, the GW chooses the one having the smallest distance with all the other sets in the same cluster. To do this, the GW makes the following:

1. The GW computes the distances among all similar pairs of sets for each cluster using the distance matrix as shown in Fig. 2, Where, $EUC_D(R_i, R_j)$ represents the Euclidean distance between data sets R_i, R_j of two sensor nodes S_i and S_j. Fig. (2-A) represents the distances among all the sensors' data sets of a specific cluster.
2. The GW finds the sum of distances for each row in the distance matrix, where this sum represents the distance of a specific sensor node with all other nodes in the same cluster as shown in Fig. (2-B).
3. The GW selects the sensor node with the lowest total sum of distances, which represents the node closest to all other nodes within the cluster.
4. The GW holds this sensor node and sends it to the sink with similar sensor node numbers within the cluster where this node is representative of the rest of the nodes that will be deleted.

Using this strategy, the GW will work to reduce the number of data sets sent, thus maintaining its energy and extending its life for the longest possible period while ensuring acceptable data accuracy. We call this strategy HOSD. Algorithm 2 explains how this strategy works.

The time complexity of Algorithm 2 is $O(K.((C_i.Size)^2 + (DIS_{MTX})^2 + Total_{Sum})$, where K is the number of clusters, $C_i.Size$ refers to the size of cluster C_i, DIS_{MTX} refers to the length of matrix DIS_{MTX}, and the $Total_{Sum}$ is the length of the vector $Total_{Sum}$.

4. Simulation experiments and results

In this section, we show the performance evaluation and simulation results as graphs and discussion for the proposed technique is outlined in Section 3. The goal is twofold: first, evaluating the performance of the technique with different performance metrics via real sensor data. Second, comparing the technique proposed with recent existing protocols belongs to the same field.

4.1. Simulation environment

To evaluate proposed techniques, extensive simulation experiments are carried out using the OMNeT++ simulator and are dependent on actual data from sensors. A network of N sensors and a single-hop topology was considered and installed in the laboratory during these simulations. The middle of the laboratory comprises a single GW node. This installation is shown in Fig. 3.

Periodically, sensors measure the local measurements at a set frequency (e.g., temperature). Proposed techniques are disseminated in every sensor node, which is dependent on using the Intel Berkeley

	S_1	S_2	.	.	S_n
S_1	0	$EUC_D(R_1, R_2)$	.	.	$EUC_D(R_1, R_n)$
S_2	$EUC_D(R_2, R_1)$	0	.	.	$EUC_D(R_2, R_n)$
.	.	.	0	.	.
.	.	.	.	0	.
S_n	$EUC_D(R_n, R_1)$	$EUC_D(R_n, R_2)$	.	.	0

A

S_1	$\sum EUC_D(S_1)$
S_2	$\sum EUC_D(S_2)$
	.
	.
S_n	$\sum EUC_D(S_n)$

B

Fig. 2. Distance matrix.

Algorithm 2: HOSD

Input: K set of clusters: $Cluster = \{C_1, C_2, \ldots, C_K\}$, received data sets: $R = \{R_1, R_2, \ldots, R_n\}$
Output: List: list of representative data sets sent to the Sink

```
List ← ∅
for each cluster C_i ∈ Cluster Do
    if (C_i.Size() = 1) then
        List ← List ∪ R_Idx
        Cluster ← Cluster − {C_i}          \\ remove C_i from Cluster
    else
L ← P ← 0                                  \\ Sensor number
DIS_MTX ← ∅, Total_Sum ← ∅
    for x ← 1 to C_i.Size() Do
        L ← C[x]
        for y ← 1 to C_i.Size() Do
            P ← C[y]
            if (x = y) then
                DIS_MTX[x, y] ← 0
            else
                DIS_MTX[x, y] ← EUC_D(R_L, R_P)
            endif
        endfor
    endfor
    for x ← 1 to length ( DIS_MTX) Do
        for y ← 1 to length ( DIS_MTX) Do
            Sum ← Sum + DIS_MTX[x, y]
        endfor
            Total_Sum[x] ← Sum
            Sum ← 0
    endfor
    MIN ← ∞
    for x ← 1 to length ( Total_Sum) Do
        if (Total_Sum[x] < MIN) then
            MIN ← Total_Sum[x]
            Idx ← x
        endif
    endfor
    List ← List ∪ R_Idx
    Cluster ← Cluster − {C_i}              \\ remove C_i from Cluster
        endif
endfor
return List
```

Research laboratory data set [22]. These sensed weather data (like light, humidity and temperature) are collected periodically every 31 s. The nodes of the sensor utilizing a log file containing 2.3 million previously received readings in the laboratory by 47 Mica2Dot sensor nodes as shown in Fig. 3. Only one measurement for sensor nodes is used in this paper: temperature. Each sensor node shown with a yellow sign in Fig. 3 is not included in our experiments because its data may be incomplete or truncated. Then the temperature readings for 47 sensor nodes are collected and processed. The findings are 47 sensor nodes averaging.

4.2. Gateway node performance analysis

In this section, the efficiency of our proposed technique at the GW level is evaluated using various experiments. The purpose of these experiments is to prove that our technology successfully generates successful energy-saving outcomes at the IoT GW node.

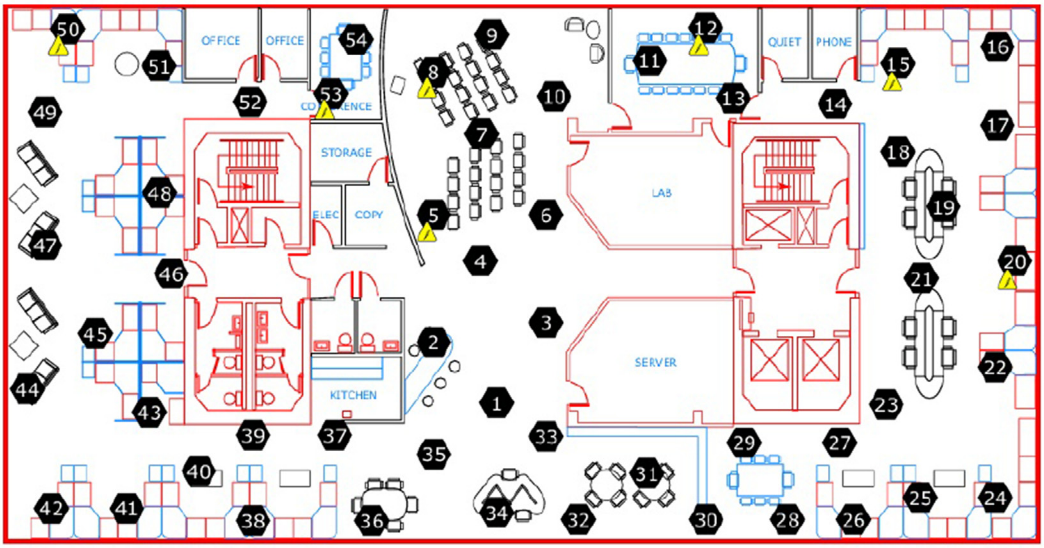

Fig. 3. Deployment of sensors in Intel Berkeley Lab.

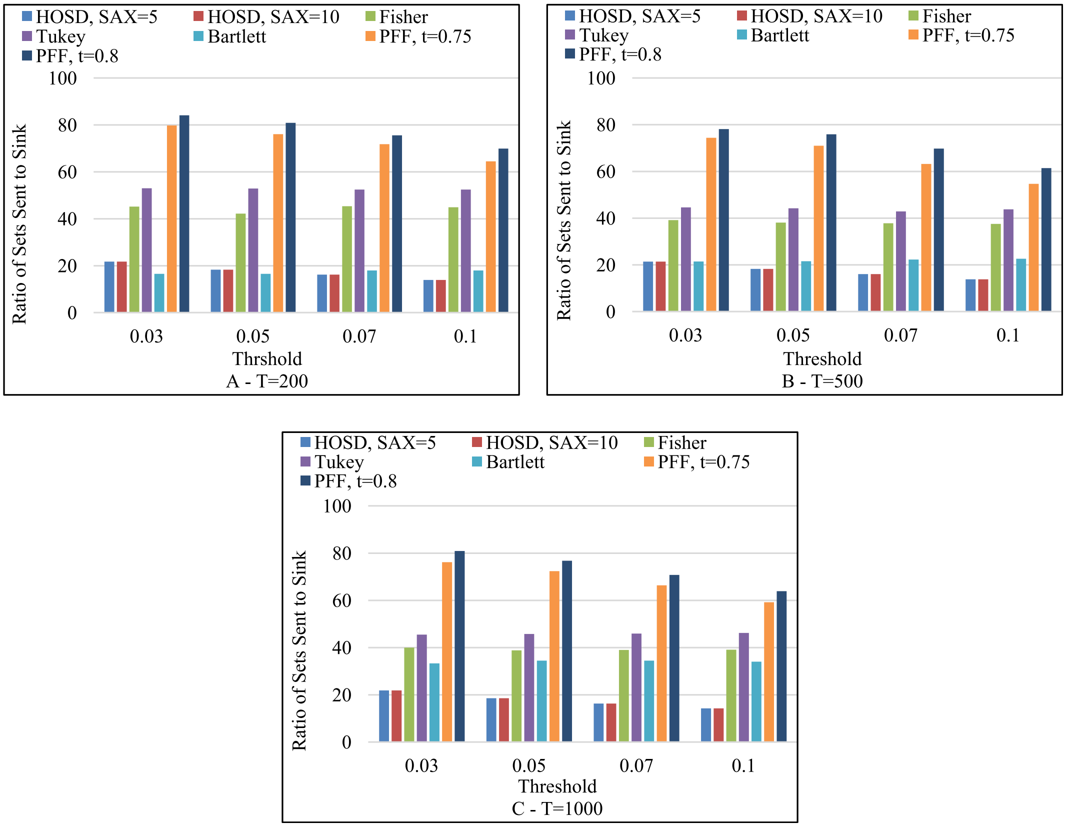

Fig. 4. The ratio of sets sent to the sink.

4.2.1. Ratio of sets sent to the sink

Within this experiment, the aim is to demonstrate how the proposed technique can be used by the GW to remove redundant sets at each period. Fig. 4 shows the percentage of residual sets that are sent to the sink after the redundancy is eliminated. We may easily see that, with the various parameters, the proposed method sends a very small number of sets to the sink throughout each cycle.

Based on the findings obtained, we can infer the following:

- The strategy suggested by HOSD at GW transmits to the sink the lower proportion of sets because it finds more redundant sets particularly in comparison with AVDOM and PFF.
- For the strategy HOSD, the proportion of sets transmitted to the sink is almost constant while T is unchanged and α is increased. This is because we deal with the average of sets so α does not affect the clustering formation.
- In the HOSD strategy, the GW reduces further duplicate sets when δ increased. This is because the number of created clusters will be reduced as δ rises and pairs of redundant sets increase within each cluster.

4.2.2. Data accuracy at GW

A significant problem for the WSN is to delete redundant data without compromising accuracy. The accuracy of data reflects the "loss rate" of readings captured by sensor nodes while the sink does not receive it. Fig. 5 indicates the proportion of accuracy (i.e., data loss rate) which will not be delivered to the sink once the redundancy sets have been removed. We may easily see that, with the various parameters, our technique gives acceptable results compared to other methods.

Based on the findings obtained, we can infer the following:

- In the proposed technique the reduction of the redundant sets has a relation to the accuracy, the loss rate increased when δ increased.
- The length of alphabet α plays an influential role in the accuracy of the data, as increasing the length of the alphabet α leads to an increase in the accuracy of the data.
- Also, the data size captured can affect the accuracy where increasing the volume of data collected T will make the rate of data loss big.
- We can see in the techniques proposed, the worst-case scenario is that the number of data sets not delivered to the sink does not pass 9.07% (i.e., Hybrid, $\alpha = 5$, $\delta = 0.1$ and $T = 1000$). This amount is acceptable if compared with the number submitted to the recipient (the total of the deleted data does not influence the decision-making of the recipient based on the data obtained). So, our techniques minimize the number of sets of

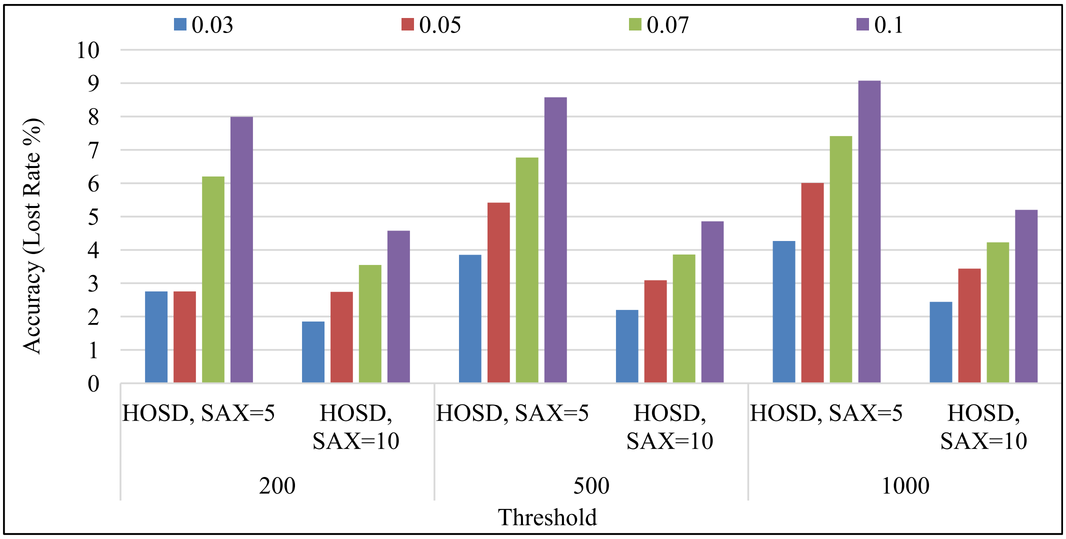

Fig. 5. The Accuracy (loss rate %) of sets at GW.

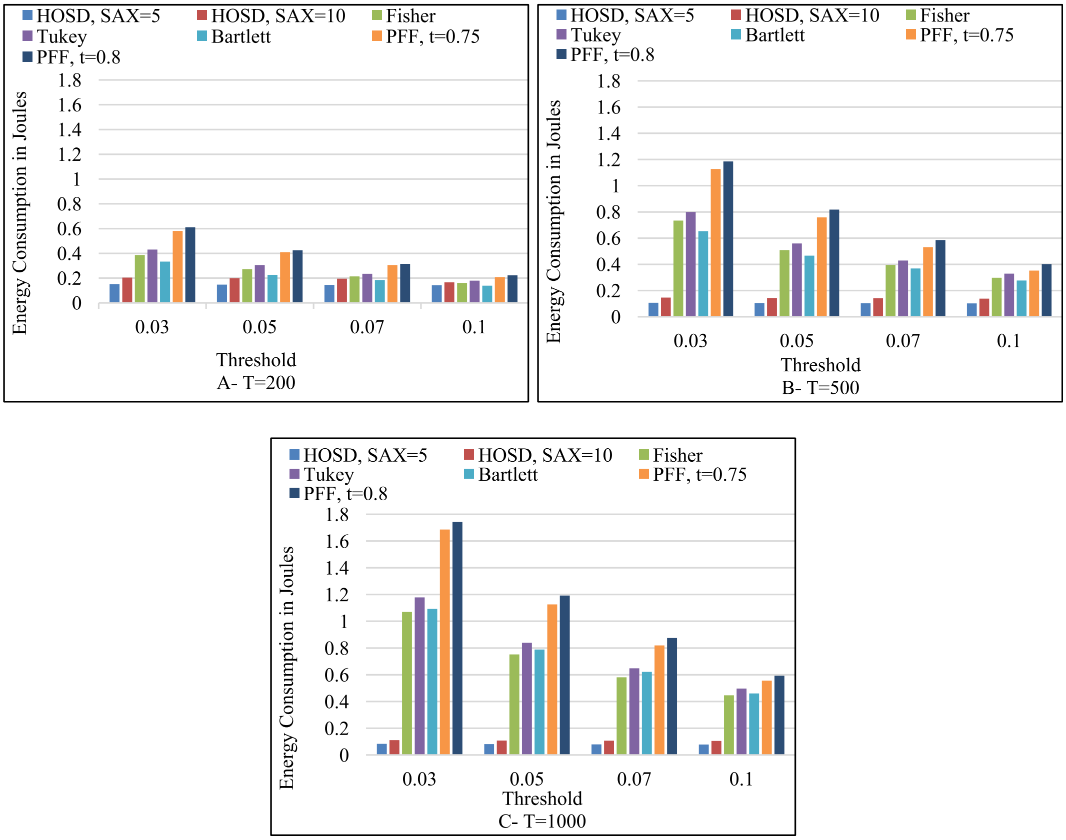

Fig. 6. The energy consumption at GW.

redundancy sent to the sink whilst maintaining an acceptable level of information accuracy.

4.2.3. *Energy consumption at GW*

Our aim in this experiment is to research the cost of energy at the GW level. At GW, the energy consumed represents the energy consumed in receiving and sending data. We use the same model for energy indicated in Ref. [23]. Fig. 6 shows a comparison between our technique HOSD and the AVMDA and PFF in terms of the amount of energy consumed using different α, δ and T. The findings obtained indicate our technique's dominance over AVMDA and PFF by reducing GW energy consumption for all values of α, δ and T. Based on the findings obtained, we can infer the following:

- In the proposed technique, the reduction of redundant sets has a very large impact on reducing energy consumption by reducing the operation of the radio unit (i.e. transmission and reception operations).
- The length of the alphabet α plays an influential role in energy consumption, as increasing the length of the alphabet α leads to an increase in the energy consumption, the reason is due to sending more packets.
- Also, the data size captured T and similarity threshold δ can affect the energy consumption where increasing T and δ will reduce the energy consumption.

4.2.4. *Compression ratio at GW*

In these experiments, we will show the effect of the proposed method on data sets as a compression ratio. The compression ratio indicates how many bits are necessary to encode the data sets that arrive at the GW to the ratio of the bits required for the data sets sent to the sink after reducing the number of sets sent according to our method.

Fig. 7 shows a comparison between our technique HOSD and the AVMDA and PFF in terms of the compression ratio using different α, $\delta$and T.

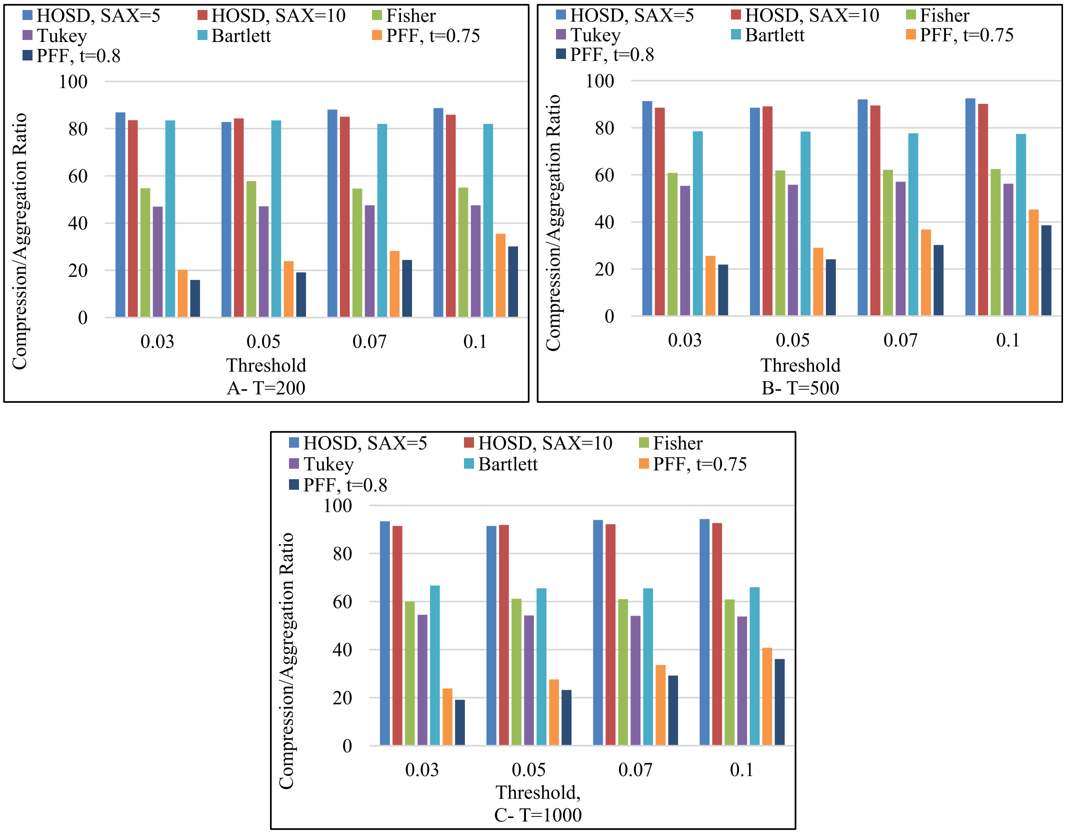

Fig. 7. The Compression ratio at GW.

Based on the findings obtained, we can infer the following:

- From the obvious results, we see that the length of the alphabet α has a fluctuating effect on the compression ratio and the reason is due to the nature and correlation of data.

- The findings obtained indicate our technique's dominance over AVMDA and PFF in terms of compression ratio in GW for all values of α, $\delta$and T.

5. Conclusion

The energy generated by the battery is the most important resource in the sensor node that effects on WSN's lifetime. To improve the WSN lifetime in this article, the principal idea is to exploit the advantage of the temporal and the spatial data correlation between the sensor nodes to minimize the energy depletion by reducing sensed data before sending them to the sink. At the gateway level, the proposed methods work as filtering via enabling the GW to identify, then remove, sets of data that are redundant and produced by neighbouring nodes, so that the final sets to be transmitted to the sink are reduced. The method recommended at this level for minimizing the number of data sets being transmitted is to hold one data set and delete the other. First, the data sets received from the first level are clustered using the Leader method. Then, our method will recognize all pairs of member nodes which produce redundant sets so that redundancy can be removed before being sent to the sink. The simulation results based on real data of the sensor network using OMNeT++ simulator show that the proposed data reduction approach outperforms some recent existing approaches in terms of several performance metrics like remaining data after compression, the ratio of sets sent to the sink, data accuracy at GW, the compression ratio at GW and energy consumption at GW. Comparisons between AVMDA/PFF and recommended methods resulted in a reduction in the volume submitted to the sink of 59%−74% and a reduction in energy use of 90%−93%, a compression ratio of up to 83%−94% was obtained, while accuracy was 91%−97%.

For future studies, we'll look towards developing a dynamic compression method that can transition from lossless to lossy based on specific criteria, such as residual energy-related factors. In order to forecast the missing data at the GW and the sink node, we also expect to implement a prediction approach and combine it with our work.

Funding

Not applicable.

References

[1] A.K.M. Al-Qurabat, A.K. Idrees, Data gathering and aggregation with selective transmission technique to optimize the lifetime of Internet of Things networks, Int. J. Commun. Syst. 33 (2020), e4408, https://doi.org/10.1002/dac.4408.

[2] R. Maivizhi, Y. Palanichamy, Q-learning based routing for in-network aggregation in wireless sensor networks, Wireless Netw. 27 (2021) 2231−2250, https://doi.org/10.1007/s11276-021-02564-8.

[3] H. Ramezanifar, G. Mahdieh, S. Maryam, A new data aggregation approach for WSNs based on open pits mining, Wireless Netw. 27 (2021) 41−53, https://doi.org/10.1007/s11276-020-02442-9.

[4] A.K. Idrees, A.K.M. Al-Qurabat, Energy-efficient data transmission and aggregation protocol in periodic sensor networks based fog computing, J. Netw. Syst. Manag. 29 (2021) 1−24, https://doi.org/10.1007/s10922-020-09567-4.

[5] M. Younan, E.H. Houssein, M. Elhoseny, A. El-mageid, Performance analysis for similarity data fusion model for enabling time series indexing in internet of things applications, Peer J. Comput. Sci. 7 (2021), e500, https://doi.org/10.7717/peerj-cs.500.

[6] N.B. Jarah, Technique pair of node to provide power in WSNs, Karbala Int. J. Mod. Sci. 6 (2020) 2, https://doi.org/10.33640/2405-609X.1193.

[7] K. Das, S. Das, A. Mohapatra, A novel energy-efficient sensor cloud model using data prediction and forecasting techniques, Karbala Int. J. Mod. Sci. 6 (2020) 5, https://doi.org/10.33640/2405-609X.1650.

[8] G. Xu, Y. Shi, X. Sun, W. Shen, Internet of things in marine environment monitoring: a review, Sensors 19 (2019) 1711, https://doi.org/10.3390/s19071711.

[9] J.M. Bahi, A. Makhoul, M. Medlej, A two tiers data aggregation scheme for periodic sensor networks, Ad Hoc Sens. Wirel. Netw. 21 (2014) 77−100.

[10] H. Harb, A. Makhoul, D. Laiymani, O. Bazzi, A. Jaber, An analysis of variance-based methods for data aggregation in periodic sensor networks, in: A. Hameurlain, J. Küng, R. Wagner (Eds.), Transactions on Large-Scale Data- and Knowledge-Centered Systems XXII. Lecture Notes in Computer Science, vol. 9430, Springer, Berlin, Heidelberg, 2015, https://doi.org/10.1007/978-3-662-48567-5_6, 165 − 183.

[11] S. Arrabi, J. Lach, Adaptive lossless compression in wireless body sensor networks, in: BODYNETS 2009 - 4th Int. ICST Conf. Body Area Networks, 2009, pp. 1−8, https://doi.org/10.4108/ICST.BODYNETS2009.6017.

[12] J. Azar, A. Makhoul, R. Darazi, J. Demerjian, R. Couturier, On the performance of resource-aware compression techniques for vital signs data in wireless body sensor networks, in: 2018 IEEE Middle East North Africa Commun. Conf. MENACOMM 2018, 2018, pp. 1−6, https://doi.org/10.1109/MENACOMM.2018.8371032.

[13] E. Aboelela, LiftingWiSe: a lifting-based efficient data processing technique in wireless sensor networks, Sensors 14 (2014) 14567−14585, https://doi.org/10.3390/s140814567.

[14] F. Marcelloni, M. Vecchio, An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks, Comput. J. 52 (2009) 969−987, https://doi.org/10.1093/comjnl/bxp035.

[15] B. He, Y. Li, H. Huang, H. Tang, Spatial−temporal compression and recovery in a wireless sensor network in an underground tunnel environment, Knowl. Inf. Syst. 41 (2014) 449−465, https://doi.org/10.1007/s10115-014-0772-9.

[16] H.P. Medeiros, M.C. Maciel, R. Demo Souza, M.E. Pellenz, Lightweight data compression in wireless sensor networks using Huffman coding, Int. J. Distrib. Sens. Netw. 10 (2014) 672921, https://doi.org/10.1155/2014/672921.

[17] J.G. Kolo, S.A. Shanmugam, D.W.G. Lim, L.M. Ang, K.P. Seng, An adaptive lossless data compression scheme for wireless sensor networks, J. Sensors 2012 (2012) 539638, https://doi.org/10.1155/2012/539638.

[18] A. Kiely, M. Xu, W.Z. Song, R. Huang, B. Shirazi, Adaptive linear filtering compression on realtime sensor networks, Comput. J. 53 (2010) 1606−1620, https://doi.org/10.1093/comjnl/bxp128.

[19] Y. Liang, Y. Li, An efficient and robust data compression algorithm in wireless sensor networks, IEEE Commun. Lett. 18 (2014) 439−442, https://doi.org/10.1109/LCOMM.2014.011214.132319.

[20] S.A. Abdulzahra, A.K.M. Al-Qurabat, A.K. Idrees, Data reduction based on compression technique for big data in IoT, in: 2020 Int. Conf. Emerg. Smart Comput. Informatics, ESCI, 2020, pp. 103−108, https://doi.org/10.1109/ESCI48226.2020.9167636.

[21] G.A. Mauser, J.A. Hartigan, Clustering algorithms, J. Mark. Res. 14 (1977) 124−125, https://doi.org/10.2307/3151073.

[22] B. Peter, H. Wei, G. Carlos, M. Sam, P. Mark, T. Romain, Intel Berkeley Research Lab, 2004. http://db.csail.mit.edu/labdata/labdata.html/. (Accessed 2 April 2021).

[23] W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, Proc. Hawaii Int. Conf. Syst. Sci. 2 (2000) 10, https://doi.org/10.1109/hicss.2000.926982.