DOI: http://dx.doi.org/10.21123/bsj.2021.18.1.0184

Compression-based Data Reduction Technique for IoT Sensor Networks

Suha Abdulhussein Abdulzahra[1] *Ali Kadhum M. Al-Qurabat*[2*] *Ali Kadhum Idrees*[2]

[1] Department of Dentistry, Al-Mustaqbal University College, Babylon, Iraq
[2] Department of Computer Science, College of Science for Women, University of Babylon, Babylon, Iraq
[*]Corresponding author: Suha.Abd@mustaqbal-college.edu.iq, alik.m.alqurabat@uobabylon.edu.iq[*], ali.idrees@uobabylon.edu.iq
[*]ORCID ID: https://orcid.org/0000-0002-3254-3005, https://orcid.org/0000-0002-8522-290X[*], https://orcid.org/0000-0001-9773-0066

Received 30/3/2020, Accepted 22/7/2020, Published Online First 6/12/2020, Published 1/3/2021

Abstract:

Energy savings are very common in IoT sensor networks because IoT sensor nodes operate with their own limited battery. The data transmission in the IoT sensor nodes is very costly and consume much of the energy while the energy usage for data processing is considerably lower. There are several energy-saving strategies and principles, mainly dedicated to reducing the transmission of data. Therefore, with minimizing data transfers in IoT sensor networks, can conserve a considerable amount of energy. In this research, a Compression-Based Data Reduction (CBDR) technique was suggested which works in the level of IoT sensor nodes. The CBDR includes two stages of compression, a lossy SAX Quantization stage which reduces the dynamic range of the sensor data readings, after which a lossless LZW compression to compress the loss quantization output. Quantizing the sensor node data readings down to the alphabet size of SAX results in lowering, to the advantage of the best compression sizes, which contributes to greater compression from the LZW end of things. Also, another improvement was suggested to the CBDR technique which is to add a Dynamic Transmission (DT-CBDR) to decrease both the total number of data sent to the gateway and the processing required. OMNeT++ simulator along with real sensory data gathered at Intel Lab is used to show the performance of the proposed technique. The simulation experiments illustrate that the proposed CBDR technique provides better performance than the other techniques in the literature.

Key words: Data Compression, IoT, LZW, SAX Quantization, Sensor Networks.

Introduction

Currently, the Internet migrates from linking people to linking things, moving to the modern Internet of Things (IoT) concept. The modern concept brings objects or things into the Web and produces new business and applications. Such things, from interior wearable devices to exterior environmental sensors, become new sources, produce data on the Internet, and together make the entities on the Internet more conscious of the real world (1,2). In IoT one of the most important contributors is wireless sensor networks (WSNs). WSN includes a large number of dispersed sensors interconnected wirelessly for environmental and physical surveillance applications. As an IoT branch, wireless sensor networks (WSNs) have been commonly used in a number of smart technologies and services, like smart building, smart home, smart cities, smart industrial automation, smart transport, smart grids, and smart healthcare (3). In general, the sensing devices contain restricted-energy resources (power of battery), storage and processing capability, range of radio communication and reliability, etc., and still, their deployment should be covering a wide range area (4).

In WSN-based IoT, energy-saving is essential since sensor nodes are working by their restricted battery and if a vast number of sensors are spread over wide space or spread in a harsh or hostile area such as in the deep sea or around the volcanoes when the battery expires, it could be uncomfortable or very hard to exchange or recharge it (4,5). At IoT sensor nodes, energy is disposed in too many ways like receiving and transmitting the data, data processing, sensing, etc., Among all these, transferring the data is very costly in terms of power exhausting, while the consumption in data processing is considered to be much fewer (6,7). Transferring a single bit of data almost consumes energy equal to that required to process a thousand operations in a regular sensor node. For that reason, how to decrease the power exhausting of IoT sensor nodes became a critical problem for increasing the duration of life of the IoT network to attain the application demands. There are too many techniques and concepts concentrated for saving the power, specially focalize to decrease the transmission of data (2).

Like that solution is appropriate in applications that do not need data in real-time and specifically useful when sensor nodes need to send regularly their data readings to the gateway (GW) for a very long time. To decrease the quantity of transmitted data, need to compress them inside the network. Relying on the recoverability of data, the data compression schemes can be categorized into three categories: unrecoverable, loss, and lossless (8).

lossless compression means that after accomplishing the decompression operation, can get quite the same data as those before accomplishing the compression operation. A loss compression means that some (usually minor) features of data may be lost because of compression operation. Finally, an unrecoverable compression refers to the irreversible compression operation. In other meaning, the decompression operation is not existing. For instance, a set of numbers can be compressed by using their average value but every one of the original numbers cannot be obtained from this average value (8). Therefore, considerable energy can be saved by decreasing the number of data transmissions (i.e. compressing data) in IoT sensor networks. For that reason, this research target to evolve a lightweight algorithm of data compression.

The contributions made by this research are as follows:

1. A new Compression-Based Data Reduction (CBDR) technique is proposed to compress the IoT sensor data readings in an effective way that saving the power, decreases the volume of transmitted data, and maintain the accuracy of the received data readings at the gateway thus extend the IoT network lifetime. CBDR has composed of two stages of compression: a lossy SAX Quantization stage that decreases the dynamic range of the sensor data readings and greatly increases the amount of reoccurring data patters, the next stage is a lossless LZW compression to compress the lossy quantization output.
2. This research presents an efficient data transmission approach for IoT sensor networks based on data correlation that could help to prolong IoT network lifetime.
3. CBDR technique is evaluated by using extensive simulation experiments provided by the OMNeT++ network simulator. CBDR is compared with two algorithms in the related works: PFF algorithm suggested by Bahi et al. (2014) (9) and ATP protocol suggested by Harb et al. (2015) that proposed in (10).

The remainder of this research is arranged as follows. The next part provides related works. Part III gives a detailed description of our proposed technique. Section IV inspect the results of experiments, Finally, this paper is ended in section V with conclusions.

Related Works

The main aim of this review is to thoroughly examine published works of literature on prolonging the lifetime of IoT sensor networks using data compression approaches. There are many techniques and concepts devoted to save energy and extend the lifetime of IoT sensor networks, mainly focused to reduce data transmissions, like predictive monitoring, clustering, aggregation, routing scheduling, data compression, radio optimization, and battery repletion (11,12,13,14,15,16,17). Please observe that several algorithms of data compression have been used in WSNs.

Although a lot of former results assess compression techniques, few have been assessed from the sensors network viewpoint. In IoT sensor nodes, the concentration should be on energy and other needs of resources rather than merely the compression ratio (2).

The algorithm of compression performed on sensor nodes must have a high compression ratio to decrease both the transmitted bits number and the percentage of power consumption. A lot of compression techniques that aware of resources have been developed and used to decrease data in WSNs (18). To compress local climate data, a lossy temporal compression algorithm called “Lightweight Temporal Compression (LTC)” has been proposed in (19). The researchers explained that the LTC is convenient for devices with low energy, it implements compression in a similar way to the “Lempel-Ziv-Welch (LZW)” and wavelet compression, low CPU consumption and needs a little storage space.

To improve data compression in WBSN, in (20), the researchers suggested the simple delta encoding algorithm, called ”Differential Pulse Code Modulation (DPCM)”. The results cleared that the delta encoding performs better than the ”Huffman encoding” in terms of reducing the amount of data, the complexity of computational, and reduce energy consumption. A technique referred to as LiftingWise has been suggested in (21). The LiftingWise technique is an adjusted version of the original Discrete Wavelet Transform (DWT) Lifting Scheme (LS) algorithm and it can be used on a set of data with varying lengths while the original LS is used on a signal Sn with length 2n. This method has been utilized to process the data spread from objects disseminated in a monitoring environment. It was compared with two other simple compression techniques suitable for utilization in WSNs: The Offset compression and Marcelloni compression (22). The results have revealed the efficiency of this method in decreasing bits’ number of the collected data by considering the finite resources of sensor nodes.

After the aforementioned analysis, it found that presently used data compression methods have not yet established both temporal and spatial similarities inside and between nodes and that the accuracy of the recovered data is so weak that it did not satisfy the implementation requirements. Also, it found that certain suggested methods of compression are highly complex for the IoT sensors and not necessary. It was found that the physical world assumes the gradient distribution; thus, the data obtained by neighboring nodes are roughly equivalent, in keeping with the temperature experiment presented in this article. Therefore, the temporal similarity that occurs between data can be exploited to minimize that data. In this article, a Compression-Based Data Reduction (CBDR) technique has been proposed. It works in the level of IoT sensor nodes to decrease both the total number of data transmitted to the gateway and the computation time is required.

Description of Proposed CBDR

This section is intended to present the design of the proposed technique. In this research, a Compression-Based Data Reduction (CBDR) technique was suggested which works at the IoT sensor nodes level to compress their readings in an efficient way to minimize the amount of data transmitted, save the power, thus prolong the lifetime of IoT network while maintaining received data readings accuracy at the gateway. CBDR includes two stages of compression, a lossy SAX Quantization stage that reduces the dynamic range of the sensor data readings and increases the amount of reoccurring data patterns, followed by a lossless LZW compression to compress lossy quantization output. Quantizing the data readings of sensor nodes down to only the alphabet size of SAX results in a lowering at the benefit of best compression volumes, which lead to producing the best compression from the LZW end of things. Fig. 1 shows the proposed compression system flowchart. A few terms used in this research are listed in Table 1.

Table 1. A few terms used in this research.

Parameter	Description
$SAMP_{Rate}$	The Sampling Rate
R	Data Readings Series $R_i = [y_{i1}, y_{i2}, \dots, y_{iT}]$
α	The Alphabet
a	Alphabet symbols number (e.g. if a = 4 then *alphabet* = [*a*,*b*,*c*,*d*])
δ	Correlation Threshold
E_{Thr}	Energy Threshold

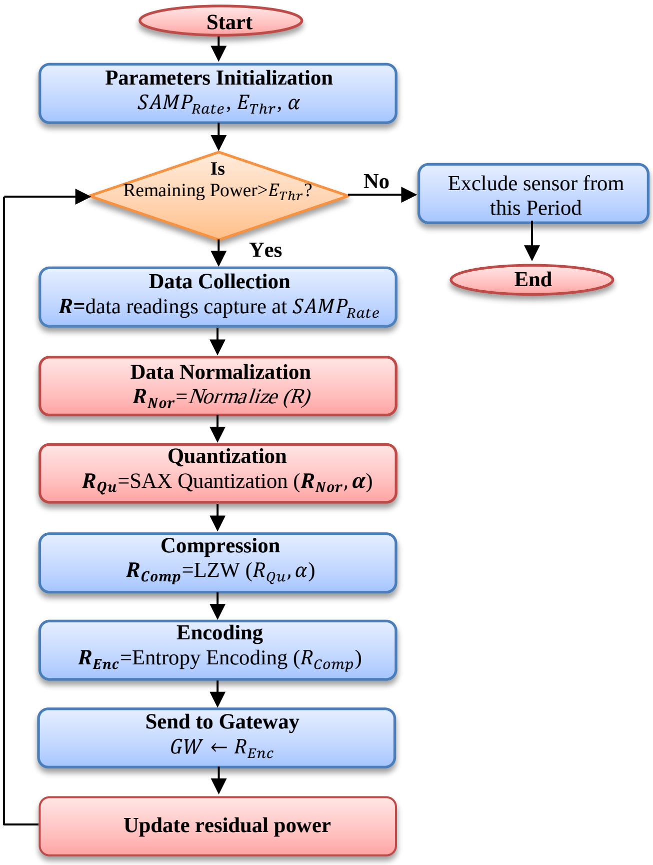
Start
Parameters Initialization
$SAMP_{Rate}, E_{Thr}, \alpha$
Is
Remaining Power>E_{Thr}?
No
Exclude sensor from this Period
End
Yes
Data Collection
R=data readings capture at $SAMP_{Rate}$
Data Normalization
R_{Nor}=Normalize (R)
Quantization
R_{Qu}=SAX Quantization (R_{Nor}, α)
Compression
R_{Comp}=LZW (R_{Qu}, α)
Encoding
R_{Enc}=Entropy Encoding (R_{Comp})
Send to Gateway
$GW \leftarrow R_{Enc}$
Update residual power

Figure 1. Flowchart of proposed CBDR method.

Data Collection

The main objective of IoT is to make human life easier and simpler. The implementation of IoT is often concerned with data collection and communication of information. In the IoT context, the data is often collected from sensors. Based on application requirements, in WSN-based IoT, the collection of data may be event-driven (like forest fire, oil and gas leaks detection) or time-driven (like habitat monitoring, logging temperature and humidity in the plants for precision agriculture) (5,11,12). This research takes into account the time-driven data collection model which is called Periodic.

During periodic data collection, sensor node i captures a new reading y_{is} for every slot of time s. After that, the node i shapes a new vector (i.e. time series vector) of captured readings $R_i = [y_{i1}, y_{i2}, \dots, y_{iT}]$ at each period ρ, while T is the number of overall readings in every period ρ, and transmit it to the suitable GW.

Figure 2 displays a periodic data collection example in which every sensor node capture one reading of data every 10 minutes, e.g. s = 10 minutes, and transmit the set of collected data that include 6 reads, e.g. T = 6, to GW at end of every hour.

As a result, one of the essential design points that should be taken into consideration correlated with the periodic collecting model of data is the conditions of surveillance cases that are dynamic can speed up or slow down. So, it is a potential IoT sensor node that takes identical (or very similar) readings many times, specifically when s is very short, which enables the IoT sensor node to transfers lots of repeated data to the GW at every period (23).

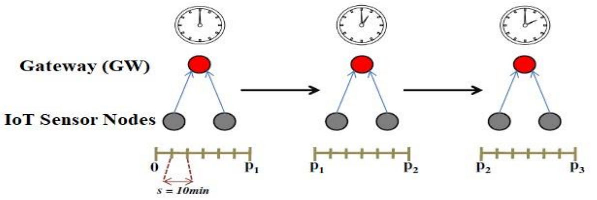

Figure 2. Illustrative example of Periodic data collection.

SAX Quantization

To make the LZW algorithm works on the collected data readings provided by IoT sensor nodes (which represent an ideal paradigm of a time-series data), some type of time series preprocessing is needed. It is desired to convert time series, which represent data readings from several IoT sensor nodes, for some appropriate formats for further analysis. To handle time series, propose to utilize two techniques of their representation: a symbolic representation and a normalization.

Normalization is known as the conversion process of time series in which making mean value equal to zero and a standard deviation one, this conversion is an essential part of the data readings preprocessing (24).

Throughout the broad field of research of time series, particularly on data mining and data management, several methods have been suggested which could be used to create an abstract representation of time series (25). It involves transforms from Fourier, wavelets, piecewise, and symbolic representations. Each of these methods guides to representations of the time series or abstractions that become generally smaller than the original time series. Since they could not be used to reconstruct the data completely, they are considered lossy compression methods (25).

There are several reasons why symbolic representation is used in a wide range, in addition to simplicity, readability and the efficiency of time series representation, it is possible to utilize algorithms from other fields such as text processing, retrieval of information or bioinformatics. One of the most successful symbolic representation techniques is Symbolic Aggregate approXimation (SAX) which proposed by Bondu et al. (26). SAX includes two parts: piecewise aggregate approximation (PAA) transformation and the transformation of the numerical data to symbols set. In this research, the concern with the second part of SAX only.

Symbolic representation of IoT sensor nodes data readings can be obtained from normalized one using the SAX algorithm. To perform this conversion, the SAX quantization utilizes $(N-1)$ breakpoints that division area under the Gaussian distribution into a equal proportional areas. Breakpoints are known as a list of sorted values $B = \beta_1,\ldots,\beta_{a-1}$. The area under an $N(0,1)$ Gaussian curve from β_i to $\beta_{i+1} = 1/a$, where β_0 and β_a indicate to $-\infty$ and ∞ respectively. The breakpoints are in a statistical table by searching for them. For example, Table 2 displays a lookup table of the breakpoints for a range of values from (3 to 10) (26).

Table 2. A breakpoint lookup table for a range of values (e.g. a: 3 to 10).

	α	3	4	5	6	7	8	9	10
β_i	β_1	-0.43	-0.67	-0.84	-0.97	-1.07	-1.15	-1.22	-1.28
	β_2	0.43	0	-0.25	-0.43	-0.57	-0.67	-0.76	-0.84
	β_3		0.67	0.25	0	-0.18	-0.32	-0.43	-0.52
	β_4			0.84	0.43	0.18	0	-0.14	-0.25
	β_5				0.97	0.57	0.32	0.14	0
	β_6					1.07	0.67	0.43	0.25
	β_7						1.15	0.76	0.52
	β_8							1.22	0.84
	β_9								1.28

When the breakpoints are determined, the normalized data set can be quantized as follow. Each normalized value less than the smallest breakpoint will be turned into ”a” symbol, while the normalized values that are equal to or larger than the smallest breakpoint and less than the second smallest breakpoint are turned into ”b” symbol, etc. Let $alpha_i$ refers to the i^{th} alphabet value (i.e., $alpha_1 = a$ and $alpha_2 = b$, etc.). As a result, the transition from a normalized representation R_{Nor} to a symbol R_{Qu} is determined as in Equation 1.

$$R_{Qu_i} = alpha_i \quad iif \quad \beta_{j-1} \le R_{Nor_i} < \beta_j \qquad (1)$$

Examples of original, normalized, and symbolic representation of IoT sensor data are shown in Table 3.

Table 3. Example of original, normalized and symbolic representation of IoT sensor readings.

Time	1	2	3	4	5	6	7	8	9	10
IoT sensor data readings	19.9884	19.3024	19.175	19.1652	19.1162	19.0672	19.0182	18.979	18.9692	18.9202
Normalized data readings	4.71968	1.12900	0.46215	0.41086	0.15438	-0.1020	-0.3585	-0.5637	-0.6150	-0.87152
Symbolic representation	J	I	G	G	F	E	D	C	C	B

Our goal of using SAX is to reduce the range of data and limit the number of alphabets used in the algorithm to increase the patterns of repeated symbols to give good results in the second stage of the proposed method. Algorithm 1 shows the SAX quantization method.

Algorithm 1: SAX Quantization

Input: $\boldsymbol{R}$ (IoT Sensor data readings with T measures); $\boldsymbol{a}$ (Alphabet Length); $\boldsymbol{\alpha}$ (Alphabetic); $\boldsymbol{\beta}$ (Breakpoints)
Output: $\boldsymbol{R_{Qu}}$ (set of symbols)

```
for $i \leftarrow 1$ to $T$ do
    $R_{Nor}(i) \leftarrow R(i) - \mu/\sigma$    // $\mu$:mean of data readings; $\sigma$: standard deviation of data readings
end
for $i \leftarrow 1$ to $T$ do
  for $j \leftarrow 1$ to $a$ do
      if($\beta_i \leq R_{Nor} < \beta_{i+1}$)then
          $R_{Qu} \leftarrow R_{Qu} \cup \alpha_i$    //convert data readings to symbols and add them to the set
      end
  end
end
return $R_{Qu}$
```

LZW Compression

IoT sensor nodes generate a mountain of data. In IoT, the data is like gold. The collected data by the IoT sensor nodes must be processed for the analyses and decision-making. As it is what enables IoT based solutions to deliver new services and opportunities. Since data transmission in IoT sensor nodes consumes a large amount of energy, so it is very costly. Therefore, in this research, the main focus is on reducing the data transmission through compressing data using the LZW algorithm for energy conservation and prolong the lifetime of the network as long as possible.

The Lempel-Ziv-Welch (LZW) algorithm (8,27) is one of the most popular lossless compression algorithms, in which the dictionary is created dynamically to encode new strings based upon strings previously encountered. Where an initiated dictionary contains the strings of a single character corresponding for all potential input characters. For instance, when using the “American standard code for information interchange (ASCII)”, the dictionary will include 256 initial entries. The LZW algorithm then searches each character of the incoming stream of data until a substring that was not in the dictionary can be found. When it detected such a string, the longest identical substring index in the dictionary sends to the output stream of data, while adding the new string into the dictionary with the next of obtainable code. Then, the LZW algorithm continues of checking the input stream of data, it starts with the last character of the preceding string (8,27).

The LZW algorithm is simple in terms of computation and has no overhead transmission. This is because the sender (IoT sensor node) and the recipient (GW) get the same preliminary dictionary entries and all-new dictionary entries can be extracted from existing dictionary entries and the input stream of data, as the result, the receiver can construct the complete dictionary on the fly when compressed data is received.

Even though the above observation, a few deficiencies or limitations related to the encoding of LZW was faced:

1. The LZW algorithm is only appropriate for text files. Therefore, to solve this limitation, this research proposes to use normalization and SAX quantization to convert the IoT sensor data readings from real numbers to symbols.
2. All single characters must be placed in the dictionary at the beginning, although it does not participate in the encoding and decoding process, as a result, the LZW algorithm suffers from space redundancy. Therefore, this research handles this problem by initializing the dictionary to only the characters set in the alphabet of SAX. Because when convert the IoT sensor data readings using the alphabet of SAX (for example, alphabet=10 characters), the range of data readings will be from ‘A’ to ‘J’. Hence, just want to include the characters’ set of SAX alphabet in the dictionary and this will minimize the dictionary size to suit with restricted sensor nodes of IoT.

These minor modifications to the LZW method proposed in this research called SAX-based LZW.

Algorithm 2 describes the process of compressing data readings using SAX-based LZW.

Algorithm 2: SAX-based LZW Compression

```
Input:   R_Qu:(set of symbols); α:(Alphabet Length); α:(Alphabetic)
Output:  CP_R (Compressed set)
Dic ← initialize Dictionary (α)        // create the Dictionary
X ← first symbol of (R_Qu)
while ! endof (R_Qu) do
    Y ← next symbol of (R_Qu)
    if (X + Y) is in the Dic then
        X ← X + Y
    else
        CP_R ← CP_R ∪ index of X in Dic
        Dic ← X + Y
    end
  end
end
return CP_R
```

Finally, after the data is compressed using LZW, its output, which are indexes of locations in the dictionary, will be encoded and sent to the next level (GW).

Since lossless compression and lossy compression are suitable for different situations, it is possible to combine the two kinds of algorithms without disturbing each other, e.g., using a lossy compression algorithm as a filter, in our case to greatly increase the amount of reoccurring data patterns, followed by applying lossless compression to further decrease the amount of data that needs to be transmitted. Figure 3 shows the basic concept of implementing such a series of compression.

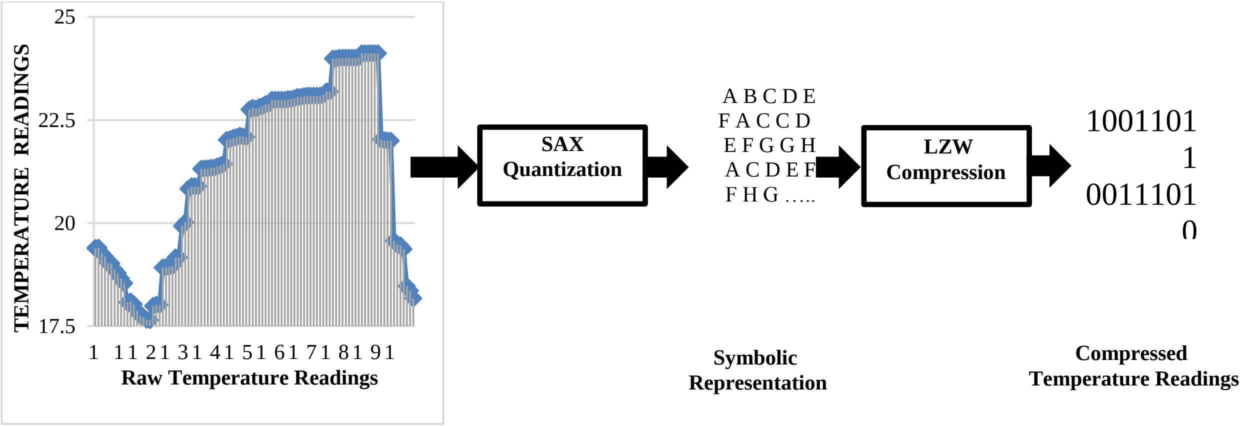

Figure 3. Basic concept of combing compression algorithms.

Dynamic Transmission

For most systems of real physical, the gradient distribution is followed by nature world physical parameters, which leads the sensing data readings for successive periods identical or with a constant difference roughly. It is responsible for the existence of a high proportion of temporal redundancies (28) that can always be called correlation. To save the power of the entire IoT network and also decreasing the number of packets sent to the GW, these redundancies in data need to be eliminated. Literature used the cosine similarity, Euclidean distance, edit distance, Jaccard's similarity, and generalized edit distance of the data to explore the correlation among sensor data readings. These methods are used to discover the similarity among data (13).

For the sake of minimizing the amount of sent data readings to the GW as much as possible, the dynamic transmission stage in the CBDR technique was proposed for more optimization, called (DT-CBDR) as illustrated in Algorithm 3.

The main responsibility of DT is to distinguish pairs of sets whose similarities are higher than a certain threshold. The DT compares between two data sets (the current and the new sets of data) for consecutive periods utilizing correlation function, and send data to the GW. If the two data sets are similar, the DT sends a notification packet only to inform the GW. Otherwise, it forwards all the new data readings to the GW (after processing them using the CBDR technique).

Figure 4 offers the behavior of DT manner. Impose v and $\hat{v}$ are two collected data sets for successive periods

where $v = [v_1, v_2, \dots, v_\rho]$ is a set of data previously collected, and $\hat{v} = [\hat{v}_1, \hat{v}_2, \dots, \hat{v}_\rho]$is a new data set, and ρ is the number of data readings in total. The correlation decision process of v and $\hat{v}$ works as follows:

Algorithm 3: Dynamic Transmission CBDR

Input: $\boldsymbol{R}$:(IoT Sensor data readings with T measures); $\boldsymbol{a}$:(Alphabet Length);

Output: $\boldsymbol{\alpha}$:(Alphabetic); $\boldsymbol{\beta}$ (Break points); $\boldsymbol{ID}$ (Sensor Identification) $\boldsymbol{CP_R}$ (Compressed set)

```
$Period_{Idx} \leftarrow 1$
// Index of Period
$\boldsymbol{while}$ $(Residual\_power) > E_{Thr})$ $\boldsymbol{do}$
    $\hat{V} \leftarrow Gather\ data\ readings\ at\ SAMP_{Rate}$
    $\boldsymbol{if}$ $(Period_{Idx} = 1)$ $\boldsymbol{then}$
        $V \leftarrow Store\ in\ memory\ (\hat{V})$
        $\hat{V}_{SAX} \leftarrow Algorithm\ 1(\hat{V}, a, \alpha, \beta)$
        $CP_R \leftarrow Algorithm\ 2(\ \hat{V}_{SAX}, a, \alpha)$
        $Send - to - GW(CP_R, ID)$
        $Update\ Residual\_power$
        $Period_{Idx} + +$
        $\boldsymbol{elseif}$ $\left(COR_F(V, \hat{V})\right) \geq \delta$ $\boldsymbol{then}$
            $Send - Correlation - Notification$
                    $- GW(ID)$
            $Update\ Residual\_power$
            $Period_{Idx} + +$
            $\boldsymbol{else}$
                $V \leftarrow \emptyset$
                $V \leftarrow Store\ in\ memory\ (\hat{V})$
                $\hat{V}_{SAX} \leftarrow Algorithm\ 1(\hat{V}, a, \alpha, \beta)$
                $CP_R \leftarrow Algorithm\ 2(\ \hat{V}_{SAX}, a, \alpha)$
                $Send - to - GW(CP_R, ID)$
                $Update\ Residual\_power$
                $Period_{Idx} + +$
            $\boldsymbol{end}$
        $\boldsymbol{end}$
    $\boldsymbol{end}$
$\boldsymbol{end}$
$\boldsymbol{return}$ $CP_R$
```

1- Compute the difference between v and $\hat{v}$, which here is the Euclidean distance. It measures the dissimilarity between each data pair in the data set and is computed by Equation 2.

$$EUC_{Dis} = \sum_{i=1}^{\rho} \sqrt{(v_i - \hat{v}_i)^2} \qquad (2)$$

2- Calculate the correlation percentage between v and $\hat{v}$ using Equation 3.

$$COR_F = \left(\frac{1}{1+EUC_{Dis}}\right) \times 100$$
(3)

3- If COR_F greater than threshold δ, then say v and $\hat{v}$ are correlative.

4- Otherwise, v and $\hat{v}$ are not correlative.

Simulation Results and Discussion:

Here, the evaluation of the performance and the results of the simulation are displayed as graphs and discussion for the proposed CBDR technique presented in section 3. The goal is twofold: firstly, to assess CBDR performance via real sensory data using different performance metrics. Proposed CBDR is disseminated in every sensor node, which is dependent on the use of the Intel Berkeley Research Lab dataset. These sensed weather data (like temperature, humidity, and light) are collected periodically every 31 seconds. In our simulations, the sensor nodes utilize a log file that includes 2.3 million readings collected formerly by 47 Mica2Dot sensor nodes in the Lab as shown in Fig. 5. This research only uses one measure of measurements of sensor nodes: temperature.

Some performance metrics are used in the experimental simulations (Table 4), to evaluate CBDR technique efficiency like *remaining data after compression, percentage of sent data to GW, compression ratio, energy consumption, lossy compression vs loss of information*, and *lifetime*. Secondly, to compare the proposed CBDR with competitive methods belong to the same field.

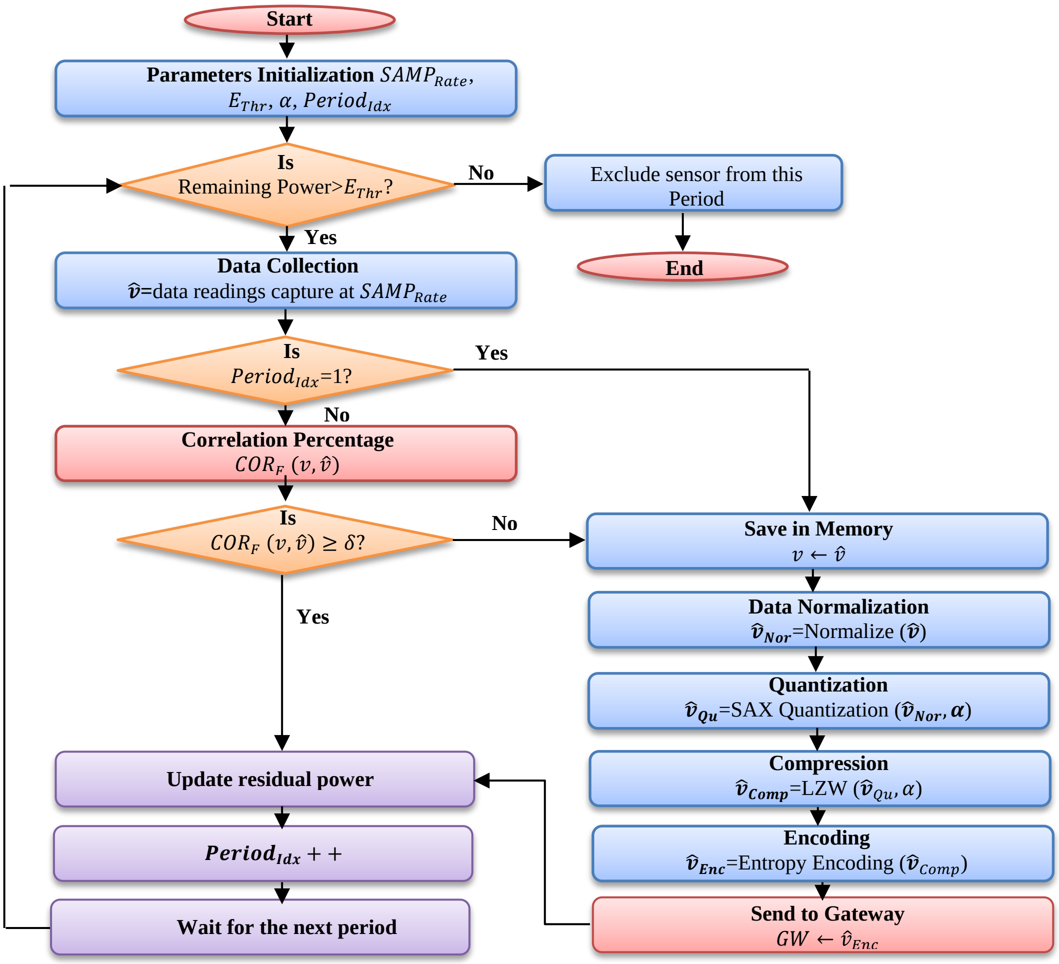
Start
Parameters Initialization $SAMP_{Rate}$, E_{Thr}, α, $Period_{Idx}$
Is Remaining Power>E_{Thr}?
No
Exclude sensor from this Period
End
Yes
Data Collection
$\hat{v}$=data readings capture at $SAMP_{Rate}$
Is $Period_{Idx}$=1?
Yes
No
Correlation Percentage
$COR_F\ (v, \hat{v})$
Is $COR_F\ (v, \hat{v}) \geq \delta$?
No
Save in Memory
$v \leftarrow \hat{v}$
Data Normalization
$\hat{v}_{Nor}$=Normalize ($\hat{v}$)
Quantization
$\hat{v}_{Qu}$=SAX Quantization ($\hat{v}_{Nor}, \alpha$)
Compression
$\hat{v}_{Comp}$=LZW ($\hat{v}_{Qu}, \alpha$)
Encoding
$\hat{v}_{Enc}$=Entropy Encoding ($\hat{v}_{Comp}$)
Send to Gateway
$GW \leftarrow \hat{v}_{Enc}$
Yes
Update residual power
$Period_{Idx}++$
Wait for the next period

Figure 4. The behavior of DT-CBDR technique.

Table 4. The settings of parameters.

Parameter	Value
Network Size	47 sensors
$SAMP_{Rate}$	20, 50, and 100 data readings
δ	0.03, 0.05, 0.07 (threshold of correlation)
α	5 and 10
Eelec	50 nJ/bit
βamp	100 pJ/bit/m^2

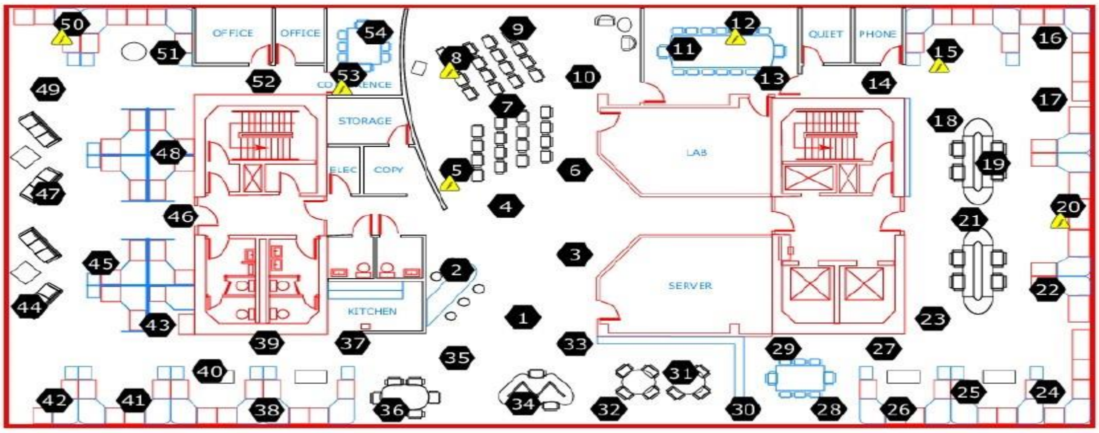

Figure 5. Deployment of Sensors in Intel Berkeley Lab.

Remaining Data After Compression

Through the compression process, every node will perform a search in its dictionary for the longest substring in temperature readings series collected in every period and allocates for each matched substring the index in that dictionary. Therefore, the result of the compression in this stage relies on the alphabet size *α* that is chosen, the changes in the conditions that have monitored, and the number of temperature readings collected in period *T*. In these simulations *α* is changed from 5 to 10 characters, *δ* from 0.03 to 0.07 and *T* from 20 to 100 readings.

The remaining data readings or compressed data are shown in Fig. 6, in every period with and without compression/aggregation and dynamic transmission at each sensor. The obtained results from CBDR and DT-CBDR technique explain that, in every period, every node decreases the amount of data collected by at least 39% and up to 79%, while ATP decreases the amount of data collected by at least 68% and up to 87% after compression/aggregation, whereas PFF sends all data collected, for example, 100%, if not applied it. So, CBDR, DT-CBDR, and ATP can get rid of redundant data readings efficiently in every period and minimize the total number of data sent to the GW.

Also, it is possible to note that in the compression stage, when T or δ increases and α decreases, the data redundancy increases. Because the compression algorithm will be able to find more repetitive patterns to be removed in each period.

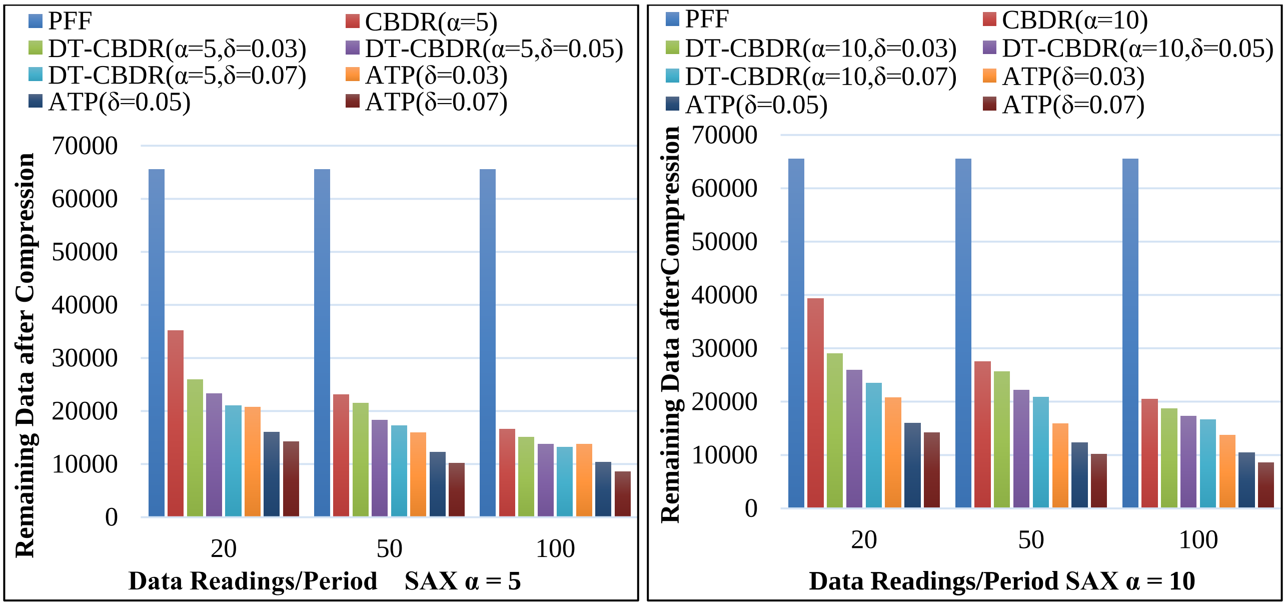

Figure 6. Remaining data after compression.

The Percentage of Sent Data to GW

The communication cost in IoT sensor networks is directly affected by the process of reducing the number of data readings (data compression). Thus, reducing the radio on-time of the transceivers (communication compression) is the result of reducing the number of packets. Figure 7 explains the percentage of sent data readings by a sensor node with the use of compression/aggregation and dynamic transmission and without. In these simulations, *α* is varying between 5 and 10 characters, *δ* between 0.03 and 0.07, and *T* between 20 and 100 data readings.

From Fig. 7 it is easy to observe that the percentage of sent data readings by IoT sensor node decreases when *α* decreases or *T* increases. The reason behind this is that the more reoccurring data patterns are, the more compression ratio results, and hence the fewer data readings transmitted that conserve the energy of the sensors. The gained results indicate that CBDR can decrease up to 74% of the sent data readings using compression only. Additionally, it’s clear that when the dynamic transmission applied, the percentage of sent data readings by IoT sensor node decreases when *α* decreases or *T* and *δ* increases. The obtained results show that DT-CBDR and ATP can reduce up to 80% and 17% respectively the sent data readings, while the percentage of sent data is equal to 100% without applying the compression/dynamic transmission such as the case of PFF.

In other words, the DT-CBDR method helps the IoT network reach a better lifetime through reducing the percentage of sent data readings but in the cost of fewer data readings integrity or fidelity. For all the values of *α*, *δ* and *T* tested, CBDR and DT-CBDR always outperform the ATP and PFF protocols in the percentage of sent data readings.

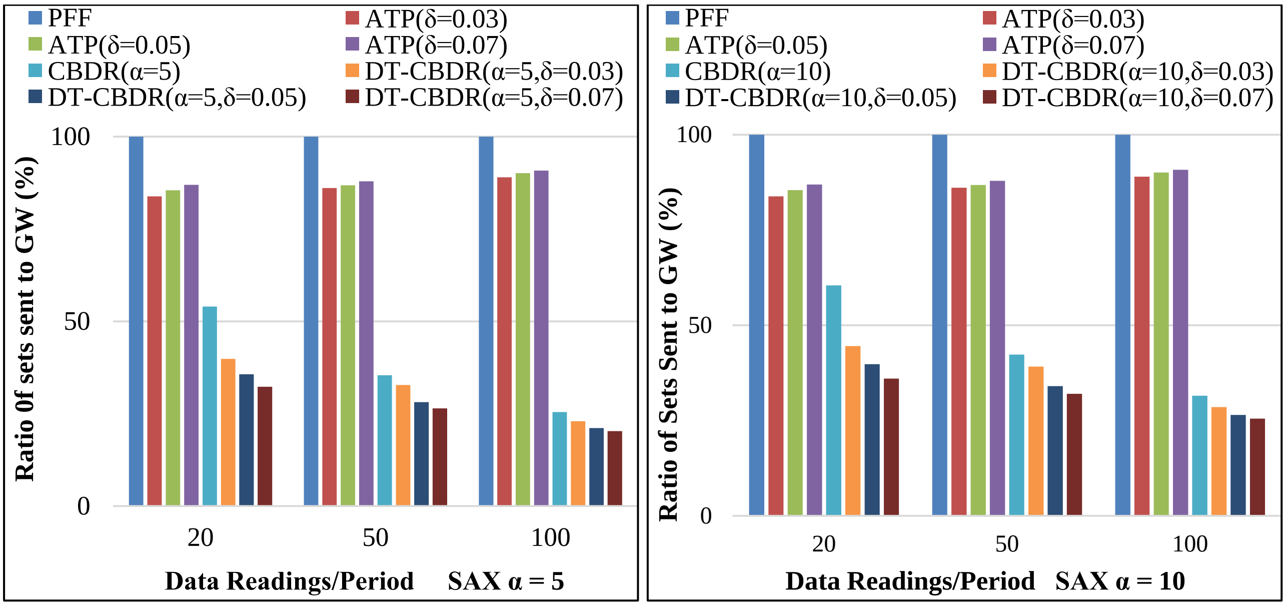

Figure 7. The percentage of sent data readings by a sensor node to GW.

The Compression Ratio

The CBDR technique compresses a specific set of temperature data readings and the logical way to measure the quality of our proposed algorithm is by measuring the percentage of the number of bits need to represent temperature data readings before compression (*Raw Readings*) to the number of bits needed to represent the temperature data readings after compression (*Compressed Readings*). This ratio is called the compression ratio COM_{Ratio} as denoted in Equation 4.

$$COM_{Ratio}(\%) = 100 \times \left(1 - \frac{Compressed_{Readings}}{Raw_{Readings}}\right) \qquad (4)$$

When analyzing the results of the simulation experiment, observe that the performance of both CBDR and DT-CBDR techniques show an interesting phenomenon compared to the ATP and PFF.

From Fig. 8, can see that the compression ratios increase when the T or δ increases, and α decreases. In most of the cases, CBDR and DT-CBDR techniques reach high compression ratios (above 95%). The ATP protocol reaches a high compression ratio of up to 87%. In contrast, the PFF compression ratio is 0% without applying any compression/aggregation techniques. Better compression algorithms have greater compression ratios. For all the values of α, δ and T tested, CBDR and DT-CBDR always outperform the ATP and PFF protocols in the compression ratio.

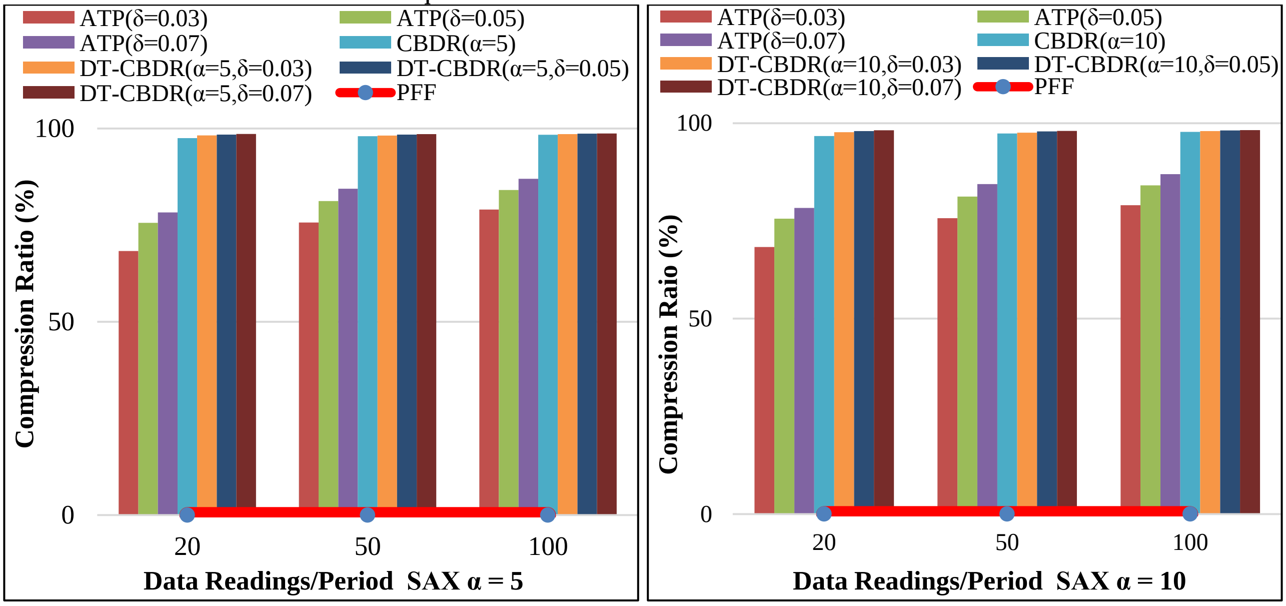

Figure 8. CBDR and DT-CBDR Compression Ratio.

Energy Consumption

The purpose of this section is to demonstrate the ability of our CBDR technique in decreasing the energy consumption. The same radio model as mentioned in (29) is used to assess the energy consumption. It is one of the most used models for energy consumption in WSNs as shown in Fig. 9.

In this model, a radio dissipates E_{elec} = 50 *nJ/bit* to turn on the sender or receiver circuitry and β_{amp} = 100 $pJ/(bit/m^2)$ for the sender amplifier. to find the transmission costs of a $k\text{−}bit$ message and a distance d, equation 5 is used:

$$E_{TX}(k,d) = E_{elec} \times k + \beta_{amp} \times k \times d^2 \qquad (5)$$

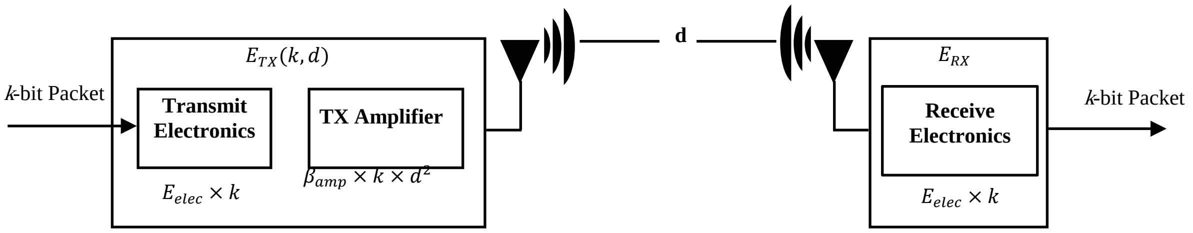

Figure 9. First Order Radio Model.

Figure 10 shows a comparison between our techniques CBDR, DT-CBDR and the ATP and PFF in terms of the amount of energy consumed using different α, δ and T. The results obtained show the superiority of our techniques over ATP and PFF by reducing (above 90%) of the energy consumption in every sensor node for all values of α, δ, and T. This is due to the compression algorithm and dynamic transmission proposed by our techniques, which reduces both the bits number needed to represent the data readings and the amount of data transmitted to the GW, this ultimately contributes to a reduction of the IoT sensor node's energy consumption and increases its lifetime.

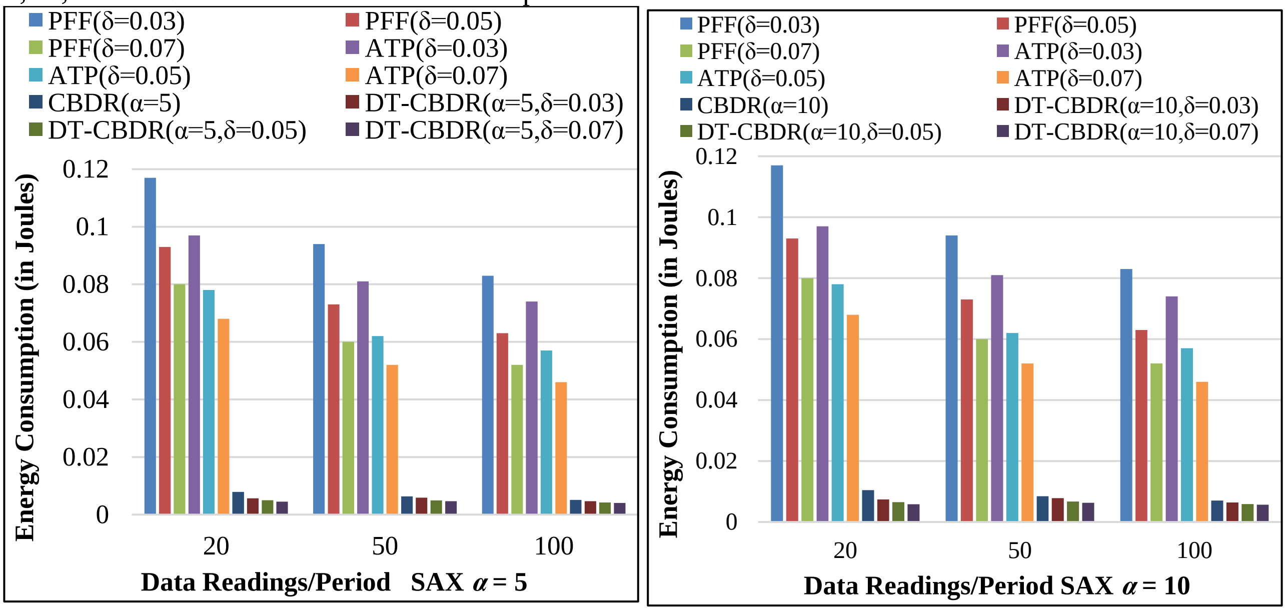

Figure 10. Energy consumption by IoT sensor nodes.

Lossy Compression vs Loss of Information

The SAX quantization in our proposed CBDR technique leads to represent existing readings within a certain range to the same symbol. This is considering a lossy compression because the data cannot completely reconstruct. In lossy compression, the data readings reconstructed at the GW different from the original data readings. A method should be used to find the difference between the original data readings and reconstructed data readings and this called the distortion (i.e. accuracy), to assess our compression algorithm efficiency. Two common measures are used to find the difference between the original and reconstructed data readings [27]: the Root Mean Squared Error (RMSE) as indicated in Equation 6 and the Percent-Root Mean Square Difference (PRD) as denoted in Equation 7.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(X(i)-\hat{X}(i)\right)^2} \quad (6)$$

$$PRD = \sqrt{\frac{\sum_{i=1}^{N}\left(X(i)-\hat{X}(i)\right)^2}{\sum_{i=1}^{N}\left(X(i)\right)^2}} \times 100 \quad (7)$$

Where X and $\hat{X}$ are the original and reconstructed data readings.

Figure 11 explains the results of data distortion (accuracy) comparison between our techniques CBDR, DT-CBDR, and the ATP and PFF while varying α, δ, and T.

The results obtained using two techniques ATP and PFF illustrate a good performance in terms of data accuracy for varying values of parameters compared with our techniques. Can see that in DT-CBDR, in the worst case, the percentage of data readings that are not reached to the GW does not exceed 5.8% (i.e. α = 5, δ = 0.07 and T = 20). This amount is insignificant if compared to the amount sent to the user (the user's decision-making based on the data received is not affected by the amount of data removed). So, our techniques reduce the amount of redundant data transmitted to the GW while maintaining an acceptable level of information accuracy.

Tables 5 and 6 illustrate the Percent-Root Mean Square Difference (PRD) achieved by our proposed technique CBDR and DT-CBDR between the original and reconstructed data readings using two values of α 5 and 10.

Based on the results in Fig. 11, Table 5 and 6, it can be deduced that the higher α, the less the difference between the original and reconstructed data readings. The reason is that the greater the number of symbols of the alphabet will decrease the range of values that convert to the same symbol and thus the smaller the difference.

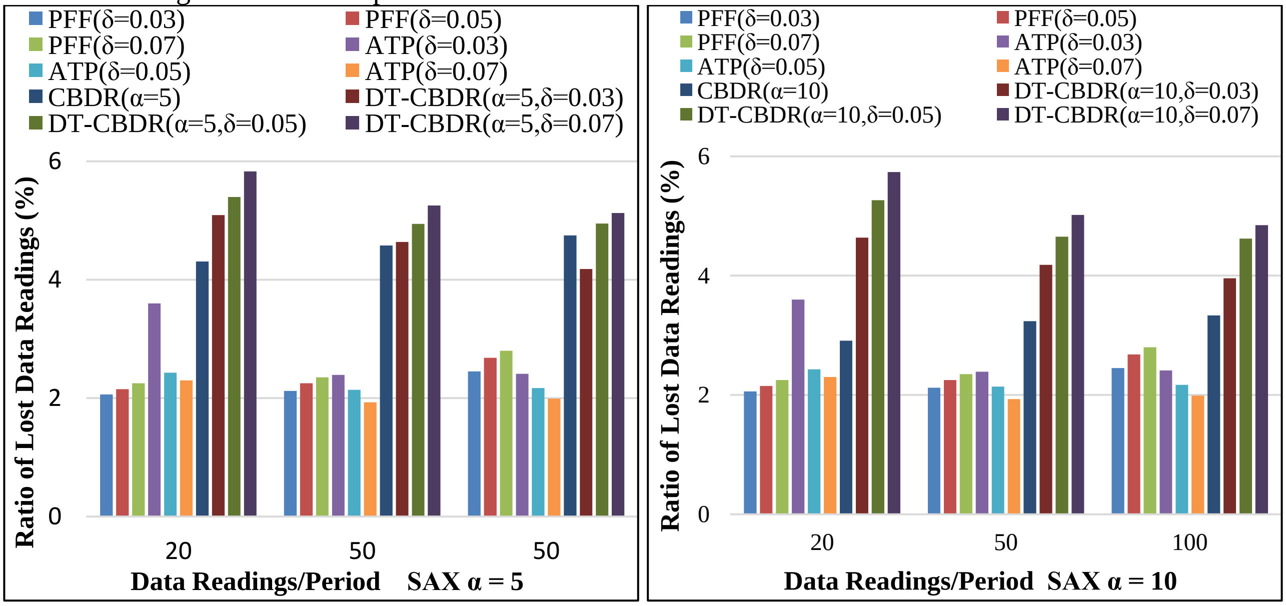

Figure 11. Lossy compression vs loss of information.

Table 5. The Percent-Root Mean Square Difference (PRD) for α=5.

T	CBDR	DT-CBDR (δ = 0.03)	DT-CBDR (δ = 0.05)	DT-CBDR (δ = 0.07)
20	0.021101	0.055133	0.059912	0.063464
50	0.017217	0.06879	0.035348	0.0369262
100	0.011048	0.0433611	0.020692	0.02120

Table 6. The Percent-Root Mean Square Difference (PRD) for α=10.

T	CBDR	DT-CBDR (δ = 0.03)	DT-CBDR (δ = 0.05)	DT-CBDR (δ = 0.07)
20	0.014276	0.0519586	0.0572254	0.061368
50	0.011331	0.0695307	0.038190	0.033415
100	0.0077	0.0436385	0.019105	0.019654

Figure 12 displays the reconstruction process for 1-period data readings using CBDR with α = 5 and 10. It is clear that when CBDR with α = 10 the restored signal matches more the original signal than CBDR with α = 5 reconstructed signals.

Lifetime

Finally, the influence of the amount of collected and sent data readings on the IoT sensor network lifetime was studied. In all the methods in this comparison, every sensor node began its energy to 2 *mJ*. In these simulations, varying α between 5 and

10 characters, δ between 0.03 and 0.07, and T between 20 and 100 data readings.

When analyzing the results of the simulation experiment, observe that both the performance of CBDR and DT-CBDR techniques show interesting phenomenon compared to the normal method (i.e. without compression).

From Fig. 13 it is easy to see that the lifetime of the IoT network increases when α or T decreases. The reason behind this is that the more reoccurring data patterns are, the more compression ratio results, and hence the fewer data readings transmitted that conserve the energy of the sensors. Additionally, can see that when the dynamic transmission applied, the lifetime of IoT network increases when α or T decreases and δ increases due to fewer data readings are transmitted hence the less energy consumed. In other words, the DT-CBDR technique helps the IoT network reach a better lifetime but in the cost of fewer data readings integrity or fidelity.

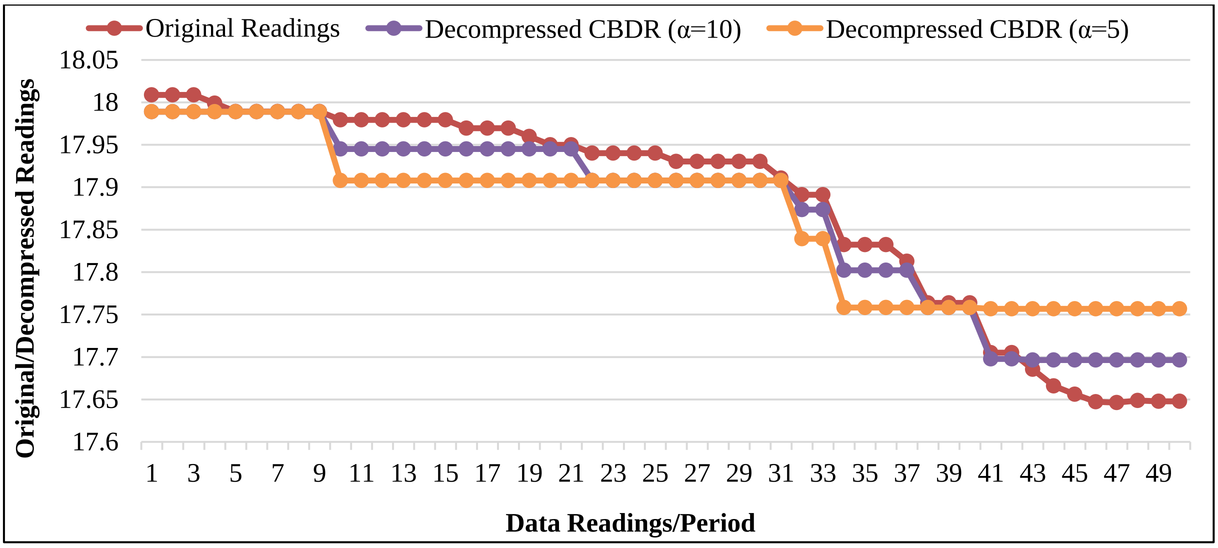

Figure 12. Displays the reconstruction process for 1 period data readings using CBDR.

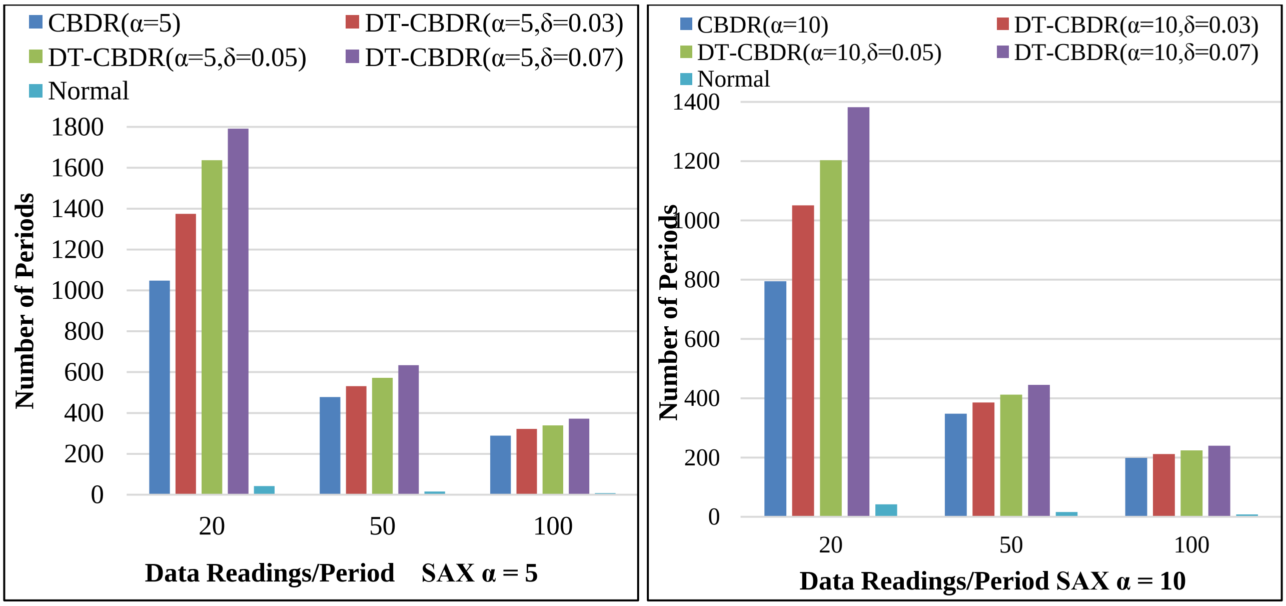

Figure 13. The lifetime of IoT network.

In this paper, the following limitations was encountered: The processor and the memory in the sensor node are limited in capabilities, so, high-complexity compression algorithms did not use. On the contrary, a simple algorithm that does not need complicated processing and large memory was suggested.

Conclusion and Future Work:

For a vast amount of data created by IoT sensor networks, data compression is very beneficial to save energy and provide important information to the end-user. In this research, a Compression-Based Data Reduction technique devoted to applications of big data in IoT networks, called CBDR, has been suggesting which works at the level of IoT sensor nodes. The CBDR includes two compression stages, a lossy SAX Quantization stage that reduces the dynamic range of the sensor data readings, after that a lossless LZW compressor to compress the output of lossy quantization. Quantizing data readings of sensor nodes down to only the alphabet size of SAX results in a decrease at the advantage of best sizes of compression, which tends to produce better compression from the LZW end of things. Also, another improvement was suggested to the CBDR method which is to add a Dynamic Transmission (DT-CBDR) to decrease both the large volume of data sent to the gateway and the processing required. It was displayed, during simulations of real sensor data, that our approaches can be used efficiently to reduce the consumption of energy in IoT networks and prolonging its lifetime by reducing the large volume of sent data readings to the GW. The simulation results show CBDR and DT-CBDR performance relative to PFF and Harb protocols, i.e. a workload decrease of up to 79% and 80% in the amount of data collected, 74% to 80% in the data transmitted, and 78% in the energy used while CBDR and DT-CBDR techniques reach high compression ratios (above 95%).

As future work, will study the possibility of proposing a dynamic compression algorithm that can convert from lossless to lossy based on certain parameters, for example, based on residual energy.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are mine ours. Besides, the Figures and images, which are not mine ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Babylon.

References:

1. Al-Qurabat AK, Idrees AK. Data gathering and aggregation with selective transmission technique to optimize the lifetime of Internet of Things networks. INT J COMMUN SYST. 2020; 33(11); https://doi.org/10.1002/dac.4408.
2. Al-Qurabat AK, Jaoude CA, Idrees AK. Two Tier Data Reduction Technique for Reducing Data Transmission in IoT Sensors. In2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) 2019 Jun 24 (pp. 168-173). IEEE.
3. Xu G, Shi Y, Sun X, Shen W. Internet of Things in Marine Environment Monitoring: A Review. Sensors. 2019 Jan;19(7):1711.
4. Liu X, Sheng Z, Yin C. Routing Protocol for Low Power and Lossy IoT Networks. In From Internet of Things to Smart Cities 2017 Sep 1 (pp. 89-118). Chapman and Hall/CRC.
5. Al-Qurabat AK, Idrees AK. Energy-efficient adaptive distributed data collection method for periodic sensor networks. IJITST. 2018;8(3):297-335.
6. Jon Y. Adaptive sampling in wireless sensor networks for air monitoring system. 2016(Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva-295995
7. Al-Qurabat A, Idrees A. Distributed data aggregation protocol for improving lifetime of wireless sensor networks. QZSJ . 2017;2(2):204-15.
8. McAnlis C, Haecky A. Understanding compression: Data compression for modern developers. " O'Reilly Media, Inc."; 2016 Jul 13.
9. Bahi JM, Makhoul A, Medlej M. A two tiers data aggregation scheme for periodic sensor networks. AD HOC SENS WIREL NE. 2014 Jan 1;21(1-2):77-100.
10. Harb H, Makhoul A, Couturier R, Medlej M. ATP: An aggregation and transmission protocol for conserving energy in periodic sensor networks. In2015 IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises 2015 Jun 15 (pp. 134-139). IEEE.
11. Al-Qurabat AK, Idrees AK. Two level data aggregation protocol for prolonging lifetime of periodic sensor networks. WIREL NETW. 2019 Aug 1;25(6):3623-41.
12. Idrees AK, Al-Qurabat AK. Distributed Adaptive Data Collection Protocol for Improving Lifetime in Periodic Sensor Networks. IAENG Int J Comput Sci. 2017 Sep 1;44(3).
13. Al-Qurabat AK, Idrees AK. Distributed data aggregation and selective forwarding protocol for improving lifetime of wireless sensor networks. J. Eng. Appl. Sci. 2018;13(5 S1):4644-53.
14. Al-Qurabat AK, Idrees AK. Adaptive data collection protocol for extending lifetime of periodic sensor networks. QZSJ. 2017 Apr 10;2(2):83-92.
15. Idrees AK, Al-Qurabat AK, Jaoude CA, Al-Yaseen WL. Integrated Divide and Conquer with Enhanced k-means technique for Energy-saving Data Aggregation in Wireless Sensor Networks. In2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) 2019 Jun 24 (pp. 973-978). IEEE.
16. Qian J, Tiwari P, Gochhayat SP, Pandey HM. A noble double dictionary based ECG Compression Technique for IoTH. IEEE Internet Things J. 2020 Feb 18.
17. Lin CH, Wang WJ, Chen JC, Lin CW. Code Compression for Embedded Systems. In Embedded, Cyber-Physical, and IoT Systems 2020 (pp. 115-147). Springer, Cham.
18. Azar J, Makhoul A, Darazi R, Demerjian J, Couturier R. On the performance of resource-aware

compression techniques for vital signs data in wireless body sensor networks. In2018 IEEE Middle East and North Africa Communications Conference (MENACOMM) 2018 Apr 18 (pp. 1-6). IEEE.

19. Schoellhammer T, Greenstein B, Osterweil E, Wimbrow M, Estrin D. Lightweight temporal compression of microclimate datasets. In 29th Annual IEEE International Conference on Local Computer Networks 2004 (pp. 516-524). IEEE.
20. Arrabi S, Lach J. Adaptive lossless compression in wireless body sensor networks. In Proceedings of the Fourth International Conference on Body Area Networks 2009 Apr 1 (pp. 1-8).
21. Aboelela E. Liftingwise: A lifting-based efficient data processing technique in wireless sensor networks. Sensors. 2014 Aug;14(8):14567-85.
22. Marcelloni F, Vecchio M. An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks. Comput. J. 2009 Nov 1;52(8):969-87.
23. Harb H, Makhoul A, Laiymani D, Bazzi O, Jaber A. An analysis of variance-based methods for data aggregation in periodic sensor networks. In Transactions on large-scale data-and knowledge-centered systems XXII 2015 (pp. 165-183). Springer, Berlin, Heidelberg.
24. Fomina M, Antipov S, Vagin V. Methods and algorithms of anomaly searching in collections of time series. In Proceedings of the First International Scientific Conference “Intelligent Information Technologies for Industry” (IITI’16) 2016 (pp. 63-73). Springer, Cham.
25. Eichinger F, Efros P, Karnouskos S, Böhm K. A time-series compression technique and its application to the smart grid. The VLDB J. 2015 Apr 1;24(2):193-218.
26. Bondu A, Boullé M, Cornuéjols A. Symbolic representation of time series: A hierarchical coclustering formalization. In International Workshop on Advanced Analysis and Learning on Temporal Data 2015 Sep 11 (pp. 3-16). Springer, Cham.
27. Sayood K. Introduction to data compression. Morgan Kaufmann; 2017 Oct 23.
28. Liu C, Luo J, Song Y. Correlation-model-based data aggregation in wireless sensor networks. In2015 12th international conference on fuzzy systems and knowledge discovery (FSKD) 2015 Aug 15 (pp. 822-827). IEEE.
29. Heinzelman WR, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd annual Hawaii international conference on system sciences 2000 Jan 7 (pp. 10-pp). IEEE.

تقنية تقليل البيانات القائمة على الضغط لشبكات أجهزة استشعار إنترنت الأشياء

سهى عبد الحسين عبد الزهرة[1] علي كاظم محمد الغرابي[2] علي كاظم دريس[2]

[1] قسم طب الاسنان، كلية المستقبل الجامعة، بابل، العراق
[2] قسم علوم الحاسوب، كلية العلوم للبنات، جامعة بابل، بابل، العراق

الخلاصة:

في شبكات أجهزة استشعار إنترنت الأشياء ، يعد توفير الطاقة أمرًا مهمًا جدًا نظرًا لأن عقد أجهزة استشعار إنترنت الأشياء تعمل ببطاريتها المحدودة. يعد نقل البيانات مكلفًا للغاية في عقد أجهزة استشعار إنترنت الأشياء ويهدر معظم الطاقة ، في حين أن استهلاك الطاقة أقل بكثير بالنسبة لمعالجة البيانات. هناك العديد من التقنيات والمفاهيم التي تعنى بتوفير الطاقة ، وهي مخصصة في الغالب لتقليل نقل البيانات. لذلك ، يمكننا الحفاظ على كمية كبيرة من الطاقة مع تقليل عمليات نقل البيانات في شبكات مستشعر إنترنت الأشياء. في هذا البحث ، اقترحنا طريقة تقليل البيانات القائمة على الضغط (CBDR) والتي تعمل في مستوى عقد أجهزة استشعار إنترنت الأشياء. يتضمن CBDR مرحلتين للضغط ، مرحلة التكميم باستخدام طريقة SAX والتي تقلل النطاق الديناميكي لقراءات بيانات المستشعر ، بعد ذلك ضغط LZW بدون خسارة لضغط مخرجات المرحلة الاولى. يؤدي تكميم قراءات البيانات لعقد المستشعر إلى حجم ابجدية الـ SAX إلى تقليل القراءات ، مع الاستفادة من أفضل أحجام الضغط ، مما يؤدي إلى تحقيق ضغط أكبر في LZW. نقترح أيضًا تحسينًا آخر لطريقة CBDR وهو إضافة ناقل حركة ديناميكي (DT-CBDR) لتقليل إجمالي عدد البيانات المرسلة إلى البوابة والمعالجة المطلوبة. يتم استخدام محاكي OMNeT ++ جنبًا إلى جنب مع البيانات الحسية الحقيقية التي تم جمعها في Intel Lab لإظهار أداء الطريقة المقترحة. توضح تجارب المحاكاة أن تقنية CBDR المقترحة تقدم أداء أفضل من التقنيات الأخرى في الأدبيات

الكلمات المفتاحية: ضغط البيانات، إنترنت الأشياء، LZW، SAX الكمي ، شبكات الاستشعار