International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 11, No.1 (Jan-2022)

http://dx.doi.org/10.12785/ijcds/110109

A Lightweight Huffman-based Differential Encoding Lossless Compression Technique in IoT for Smart Agriculture

Ali Kadhum M. Al-Qurabat[1]

[1]*Department of Computer Science, College of Science for Women, University of Babylon, Babylon, Iraq*

Received 2 Feb. 2021, Revised 16 May 2021, Accepted 8 Jul. 2021, Published 15 Jan. 2022

Abstract: Major advantages occur in modern agriculture, including effective position and space needs, sufficient meteorological management, water efficiency, and controlled nutrient use. The Internet of Things (IoT) definition suggests that different "things," such as communication devices as well as all other physical objects in the world, can be connected and regulated over the Internet. Wireless Sensor Networks (WSNs), in particular, may be thought of as important data collection and transmission systems. It is possible to build automated systems for improved agricultural environmental control using the IoT and WSN. But WSN is suffering from the motes' limited energy supplies, which decreases the total network's lifetime. Each mote periodically collects the tracked feature and transmits the data to the sink for additional study. This method of transmitting massive volumes of data allows the sensor node to use high energy and substantial bandwidth on the network. In this article, we suggest a lightweight lossless compression algorithm based on Differential Encoding (DE) and Huffman techniques that is particularly beneficial for IoT sensor nodes that monitor the features of the environment, especially those with limited computing and memory resources. Instead of trying to formulate innovative ad hoc algorithms, we demonstrate that, provided general awareness of the features to be monitored, classical Huffman coding can be used effectively to describe the same features that are measured at various time periods and locations. Results utilizing temperature measurements indicate that it outperforms common methods developed especially for WSNs, even though the suggested system does not reach the theoretical maximum.

Keywords: Data Compression, Differential Encoding, Huffman Encoding, Smart Agriculture, Energy Consumption, IoT, WSN

1. Introduction and Overview

The Internet of Things (IoT) may be defined as a platform where virtual and physical objects are interconnected and communicate with each other [1]. IoT systems consist of different technologies like cloud computing, wireless sensor networks and embedded intelligence. IoT systems consist of different technologies like cloud computing, wireless sensor networks, and embedded intelligence. IoT systems capture environmental data using RFID (Radio Frequency Identifier), cameras, sensors, and so on [2]. These systems offer advanced services such as real-time remote monitoring, online analytics, and remote management. IoT is applied in many remote monitoring applications in vast domains from healthcare to smart factories, including smart homes, smart cities, smart agriculture, improving productivity and reducing costs [3], [4].

The current agricultural fields need new and enhanced methods. With the lack of water production and the abundance of demand for it and the worrying climate change, other external problems arise [5]. Also, in order to perform activities such as watering or fertilizing, farmers need to visit their plants frequently (e.g., every day or every few days, depending on the plant and trees). In some cases, farmers need to stay close to their remote farms in order to protect the crop and their resources. When the farmed areas are large, it becomes increasingly difficult and more human resources are required to perform these tasks. This can cause a significant increase in operational costs with a limited impact on productivity [6]. In the era of the IoT, a solution is to deploy a WSN-based IoT as a low-cost remote monitoring and management system for these remote farms. Farms adopting IoT technologies are often referred to as "smart farms." Some of the benefits of the IoT can be utilized to improve the quality of services for automated and remote farming systems.

We assume that a WSN-based IoT used for smart farming gathers environmental data regularly from various sensor nodes and transmits the data to a sink for additional study [7]. This periodic method produces massive data redundancy being passed to the sink, particularly when

E-mail address: alik.m.alqurabat@uobabylon.edu.iq

there are no alterations to the monitoring function (for example, if the temperature remains constant). At the sensor node level, this vast quantity of periodic data is much more overwhelming. Typically, these sensors have limited computational, energy, and storage capacity and cannot handle or store this amount of data.Hence, issues have emerged at this stage surrounding computing resources, storage space, and data mining. And last but not least, one of the main challenges is to minimize energy usage when sending large amounts of data via the IoT network [8].

To resolve these problems and to minimize the quantity of data obtained from the sensor nodes, we suggest a simple lossless compression algorithm based on Differential Encoding (DE) and Huffman techniques. This is especially helpful for IoT sensor nodes, particularly those with restricted computing and memory resources. The compression approach benefits from the high correlation that typically happens in smart farming between consecutive measurements collected by IoT sensor nodes.

By utilizing the principle of entropy in compression along with the correlation attribute, we prove that a compression ratio higher than that achieved by state-of-the-art algorithms can be accomplished by simply using Huffman encoding. We demonstrate that if we create a fixed dictionary based on Huffman used for encoding the successive measurements differences for a large data set, the achieved ratio of compression of the similar phenomenon's test sensor data set at various locations and times is quite similar to what would be carried out if a separate dictionary were designed for every test sensor data set.

The rest of this research will be as follows: Section 2 discusses similar works, while Section 3 outlines our proposed lossless compression algorithm. A variety of experimental findings support the proposed method in Section 4. Section 5 presents a discussion about the results the method has produced. Finally, the article concludes in Section 6.

2. Related Works

WSN-based IoT data reduction has gained a lot of interest in recent years. Traditionally employed core methods of data reduction may be categorized as prediction, data aggregation, data compression, multi-channel multi-paths, compressive sensing, scheduling and clustering [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20]. We will mention and clarify some of those solutions in this section.

Both lossless and lossy solutions that leverage the strong temporal similarity of sensor node data can be found in the literature on compression techniques for WSNs. One of WSN's earliest data-compression lossy techniques, LTC (Lightweight Temporal Compression) [21], used a series of lines to estimates the data obtained by every node in a WSN. K-RLE [22] is a variant of the data compression process called run-length encoding (RLE) in WSN. It uses the range of values $[K - d, K + d]$ to approximate a series of N readings as the pair (N, d), where K determines the system's precision.

While lossy compression approaches may usually obtain high ratios of compression at the cost of modest precision losses, it might not be obvious in certain WSN implementations, until collecting data, how much information should actually be overlooked before losing the system's ultimate intent. Event-based contact strategies aim to address this issue by restricting the transition of sensor data to client query responses [23]. In certain instances, though, without first examining the raw sensory data, the client will be unable to formulate queries. Consequently, in WSN a variety of approaches for compressing data losslessly were suggested

Even so, in this research, we take into account strategies that aim to perform effective lossless compression of data by exploiting only the data temporal correlation obtained by every sensor node and locally conducting all the computations, without depending upon the input from another node. Lossless entropy compression (LEC), S-LZW and ND-Encoding are among the most popular and successful solutions in this area.

By assuming minimal sensor node resources, the researchers strengthened Lempel-Ziv-algorithm Welch's in S-LZW [24]. Using a 256-character dictionary, the algorithm effectively segments the information to be packed into fixed-size blocks, and then compresses each block individually.

In the LEC method [25], the difference between the values of the successive sensor samples is calculated, and then these differences are converted into segments that increase exponentially in size. Every segment correlate to the needed number of bits to show the differences in values. After that, the entropy with a fixed compression table (as shown in Table I) is used to code these segments according to the JPEG algorithm. The symbols being compressed are expressed by associating segment number v_i with the element index f_i inside the segment. The researchers used real data gathered by WSNs about the climate and recorded high compression ratios.

ND-Encoding [26] is a compression algorithm that achieves high-ratio compression as the data slowly different from the normal distribution. In this technique, the structured dictionary provided in Table II is used to encode the residues of the data characterized by normally distributed and very small variance.

A predictive coding method is another kind of compression algorithm. Generally speaking, this category of algorithms is inspired by the presumption that only certain residues are sufficiently encoded in certain situations, resulting from a discrepancy between the real and expected value. This is often referred to as two-modal transmission (TM). The TM approach introduced in [27] employs a second-order linear predictor. The only problem with this method is that, due to the limited computing and storage resources

TABLE I. LEC Dictionary.

d_i	v_i	f_i	l_i
0	00	0	2
-1,+1	010	1	4
-3,-2,+2,+3	011	2	5
-7,...,-4,+4,...,+7	100	3	6
-15,...,-8,+8,...,+15	101	4	7
-31,...,-16,+16,...,+31	110	5	8
-63,...,-32,+32,...,+63	1110	6	10
-127,...,-64,+64,...,+127	11110	7	12
-255,...,-128,+128,...,+255	111110	8	14
-511,...,-256,+256,...,+511	1111110	9	16
-1023,...,-512,+512,...,+1023	11111110	10	18
-2047,...,-1024,+1024,...,+2047	111111110	11	20
-4095,...,-2048,+2048,...,+4095	1111111110	12	22
-8191,...,-4096,+4096,...,+8191	11111111110	13	24
-16383,...,-8192,+8192,...,+16383	111111111110	14	26

TABLE II. ND-Encoding Dictionary.

d_i	v_i	f_i	l_i
0	00	0	2
-1,+1	01	1	3
-3,-2,+2,+3	10	2	4
-5,-4,+4,+5	110	2	5
-7,-6,+6,+7	1110	2	6
All others data	1111	ρ_i	4+ρ_i

available in IoT sensor nodes, the predictive algorithms are extremely complex, and only the sink will run the right estimator.

While some of the aforementioned approaches depend on the temporal similarity of data gathered by WSNs to reach a good compression ratio, they ignore the fact that it is normally fairly simple to predict the specifics of the phenomenon to be observed by a certain WSN before the sensors are deployed. The researchers suggested the Aggregation and Transmission Protocol (ATP), a two-phase adaptive protocol, in [28], which functions independently on each sensor node in order to reduce the transfer of data and preserve power. To extract consistency from raw data, the proposed protocol searches for associations between data obtained within a p-period during the aggregation process. Although the sensor node during the sending process is searching for a periodic data link, the Fisher test uses the one-way ANOVA model.

In [29], a new method of filtering prefixes was proposed by the writers to stop computing identical values for all possible pairs of sets. For content consistency, they describe a current filtering technique. They were interested in pursuing a new part of the filtering aggregation problem by using a local processing strategy to discern the similarities of neighboring node-created data sets.

In order to minimize the amount of information sent to the sink, the implementation of a lightweight lossless data compression algorithm is our data reduction technique at the sensor node. It is based on Huffman and differential encoding techniques.

This section presents some relevant background information related to the development of the measurement system, the energy calculation methodology, and the evaluation of the experimental results.

3. The Lightweight Data Compression

In this section, we describe the data compression issue for smart farming measurements in WSN-based IoT and introduce an easy compression solution, that considers the features of the captured measurements by the sensor nodes, which acts to decrease algorithmic complexity without compromising compression ratios.

A. *Problem Definition*

We take into account a sensor node in smart farming that monitors environmental measurements. Every sensor node captures at a time interval of $\mathfrak{t}$ one measurement that defined, after analog to digital conversion, as $\mathfrak{m}[t] \in \mathcal{M}$, where $\mathcal{M} \subset \mathbb{Z}$.

The source alphabet is said to be the set $\mathcal{M} = \{\mathfrak{m}_0, \mathfrak{m}_1, ..., \mathfrak{m}_{N-1}\}$. Every symbol $\mathfrak{m}_i \in \mathcal{M}$ is represented by a specific source encoder with a codeword of length l_i bits, such that $L = \sum_{i=0}^{N-1} p_i l_i$ expresses the mean of bits number required for representing every source symbol, where p_i is the likelihood of $\mathfrak{m}[t] = \mathfrak{m}_i$. If the source encoder is unavailable, then equal-length codewords will be used for representing all the source symbols, such that the length of symbol is $Lu = \lceil log_2 N \rceil$ bits/symbol. In fact, the theoretical limit for the minimal number of bits/symbols for a distinct source is the entropy of the source and is expressed by Equation 1:

$$\mathcal{H}(\mathcal{M}) = \sum_{i=0}^{N-1} p_i l_i = -\sum_{i=0}^{N-1} p_i log_2(p_i) \qquad (1)$$

where $l_i = -log_2(p_i)$ is the measure of information for a source symbol $\mathfrak{m}_i$. We can calculate the compression algorithm performance by making a comparison between the mean length of the symbol after compression to the entropy of the source. For example, let's take the case of the integer temperature data set, denoted as Set 1, for the period from 1/1/2009 to 7/8/2011 for the city of Hagerstown, MD, USA, which contains 26,843 samples and the rate of sampling is every 10 minutes [30]. Because the values of temperature measured ranging between −16 C and +37 C, therefore, the 54 distinct source symbols in this alphabet required $L_u = 6$ bits/symbol to represent these symbols without compression. In this case, the entropy of the source is $\mathcal{H} = 5.29$ bits/symbol, that could be achieved by simple Huffman code. Evidently, this specific source and after developing a Huffman code, is need after compression just $L = 5.31$ bits/symbol. Even so, since the values of temperature have somewhat a uniform probability distribution therefore the reduction in the mean length of symbol is just 0.69 bits or 11.5%.

We will perform even better when taking into account the successive measurements differences, such that the transmitted data will be $d_i = \mathfrak{m}_i - \mathfrak{m}_{i-1}$. For example, the entropy regarding to Set 1 for the difference in successive measurements is just $\mathcal{H}_d = 2.13$ bits/symbol, that resulted in reducing of 59.7% regarding to the temperature entropy of Set 1. The robust correlation between the successive temperature measurements has led to this reduction, which in turn causes the differences probability distribution to be significantly erratic. Therefore, the compression of the differences between the successive temperature measurements is much more appropriate than the compression of the temperatures themselves.

B. *Proposed Scheme*

The goal of this research is to formulate an easy lossless compression technique that approaching the efficiency of optimum entropy coding when depending on a fixed-dictionary. The new technique proposed introduces a minor improvement to the basic technique, which is to exclude the negative values that can occur as a consequence of finding differences between the consecutive measurements. The method for eliminating negative values suggests finding the smallest value in each group and then finding the difference between the group's values and the smallest value. This process aims to decrease the size of the dictionary utilized to encode the resulting differences without sacrificing data accuracy or compression ratio

Assuming N measurements set, $\mathcal{M} = \{\mathfrak{m}_1, \mathfrak{m}_2, ..., \mathfrak{m}_N\}$, where $\mathcal{W} - bits$ is used to define each measurement, $\mathcal{W}$ represents the resolution of analog to digital module. δ denotes the minimal measurement in set $\mathcal{M}$, i.e. $\delta = min\{\mathfrak{m}_i\}$, and denotes the set of differences as $\mathcal{D} = \{d_1, ..., d_N\}$, where $(d_i = \mathfrak{m}_i - \delta,\ i[1, ..., N])$.

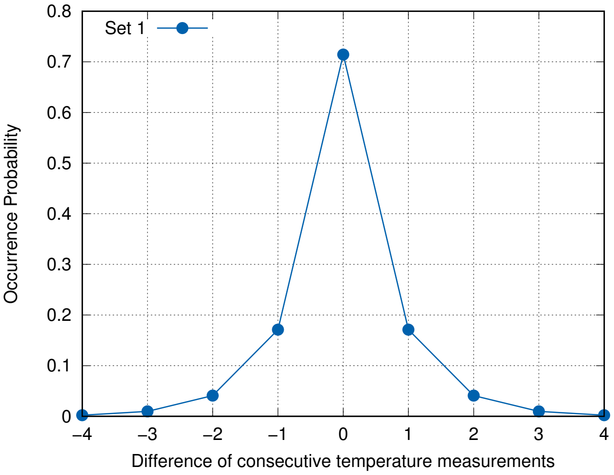

Figure 1. The successive temperature measurement differences.

It is clear that the use of set $C = \{\delta, \mathcal{D}\}$ will lead to reconstructing the original set of values $\mathcal{M}$ with very high accuracy where, $\mathfrak{m}_i = d_i + \delta$. In addition, if the correlation between measurements in set $\mathcal{M}$ is high then the set C will need a few bits to represent it. So, the set C can be interpreted as a compress version of set $\mathcal{M}$.

In fact, in our proposed technique β is used to denotes to the number of bits needed to represent the biggest difference *di*, also, δ is represented with $\mathcal{W}$ bits, as a result, the number of bits needed to represent the set C is $L_C = \mathcal{W} + \beta \times N$ bits. Now, let's compare the bits number required to represent the set $\mathcal{M}$ compared with C, i.e. $L_{\mathcal{M}} = N \times \mathcal{W}$. Presuming the correlation among the measurements of set $\mathcal{M}$ is high, then each difference d_i is represented by a few bits β such that $\beta < \mathcal{W}$, therefore if $\beta < \mathcal{W} \times (N - 1)/N$ would match $L_C < L_{\mathcal{M}}$.

So, by following this method of compression, we will have obtained a lossless data compression algorithm that we call the *Minimum Differential Encoding* (MDE) if we can represent the $\mathcal{M}$ set by the C compressed set without using a dictionary and source encoder.

When studying the distribution of probability for the measurements and for the differences of successive measurements for several sensors data set conducted at various locations we found a high similarity in the distributions of the differences across all sensors data set. As shown in Figure 1, illustrating the distribution of differences probability between successive measurements in Set 1.

Almost all of the distributions as shown in the figure are Laplacian with a mean of zero. Most importantly, if we list the differences between all sensors dataset, the result will be (0, ±1, ±2, ±3, ±4, ...), from most probability to the least.

In our case, as seen in Figure 2, there would be a large likelihood of residues having small values (i.e. $d_i <= 3$). Therefore, in practice, a fixed Huffman alphabet can be used to compact various sets of measurement if we take into account the temperatures differences, since all sets have very similar behaviour and the ideal Huffman alphabet appears to be identical for each set.

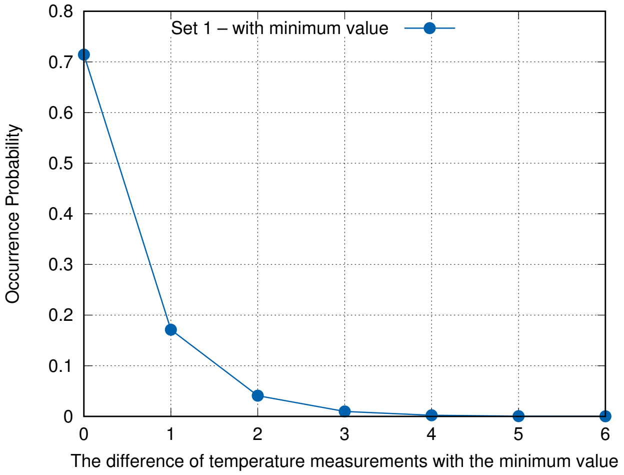

Figure 2. Successive measurement differences with minimum value.

Thus, we suggest in this research to create a fixed alphabet derived by applying the Huffman algorithm to a broad temperature measurement data set of sensors. We assign Sensor 1 to be the reference data set, for no special reason. The proposed technique utilizes for creating a dictionary a data set as a reference for a specific criterion under monitoring (e.g. temperature), unlike LEC, that often utilizes the same dictionary. In the reference data set, the repetition of each available symbol is counted for using them to create the Huffman tree that describe the alphabet of compression. The various temperature datasets are compressed based on this alphabet, and as illustrated in Table III. The suggested technique's complexity is very limited due to the fixed alphabet. For example, the AVR microcontrollers, commonly utilized in sensor nodes, require just 468 bytes of memory when implementing the encoding and decoding of Huffman [31]. Due to its ultimate simplicity we use Huffman coding in this research; furthermore, similar findings will possibly be provided by other entropy coding methods like arithmetic coding. Then we can describe the modified algorithm as MDEH if we use the MDE together with the Huffman dictionary as presented in Table III. Note that MDE and MDEH are methods of lossless compression.

TABLE III. MDEH Dictionary.

d_i	*codeword*	l_i
0	1	1
1	10	2
2	100	3
3	1000	4
4	10000	5
5	100000	6
ξ	000000	$6+\rho_i$

In the suggested technique, like in any differential compression method based on a dictionary, two specific situations should be assumed: (i) the minimum measurement, δ, should be sent uncompressed; (ii) also, the table of compression, e.g. Table III, encompasses a small number of difference values, depending on the data present in the reference dataset. Even so, the likelihood of a word that doesn't appear in the dictionary is incredibly low (view Figure 2). And therefore its value may be transmitted uncompressed and defined in the Huffman dictionary by the inclusion of a special symbol. In other words, the codeword not included in the original dictionary and whose existence reveals without ambiguity can be sent to indicate that the symbol that follows it is not compressed using a codeword of predetermined length. The special symbol in Table III is ξ with 6 bits. After the MDEH calculate the difference $d_i = \mathfrak{m}_i - \delta$, it will be feeding it to the entropy encoder. The encoder is lossless and based on the dictionary presented in Table III, a codeword string is used to express any difference value d_i that is non-zero. Finally, it must be noticed that the proposed MDEH needs a very limited dictionary that could be avoided using the statement of switch-case. As a consequence, we might assume that MDEH has lower memory requirements compared to other algorithms of compression.

4. Results

To test our proposed technique using the OMNET++ discrete simulator, we carried out multiple collections of simulation experiments. We use a set of real data in simulation experiments, as an example of weather measurements, obtained by sensors equipped with a weather-board from the research center of Intel Berkeley [32]. In Berkeley, there are 54 sensors placed at the lab for collecting environmental measurements like light values, voltage, temperature and humidity, one measurement every 31 seconds. In this research, only one class of the sensor node measures has been taken for the purpose of the study, which is the temperature (the rest of the measures can be treated in the same manner) for simplicity.

A. Comparing with Lossy Algorithms

In this section, the efficiency of MDE/MDEH techniques is compared with ATP and PFF. ATP and PFF are protocols for WSNs devoted to lossy aggregated data. As we have known, the compression ratio in lossy approaches is higher than lossless approaches, even so, we selected ATP and PFF to demonstrate the efficacy of our techniques. In this relation, we carried out a number of simulation tests with various criteria so as to include all the cases. Owing to the scale of experiments and limited papers, the findings of the tests are not displayed all. We have evaluated the efficiency of our proposed techniques and used the following metrics:

quantity (in KB) of transmitted measurements, compression ratio, energy consumption and data precision.

Figure 3 displays the quantity of sent measurements (in KB) by sensor nodes using different techniques (PFF, ATP, MDE and MDEH) to the sink.

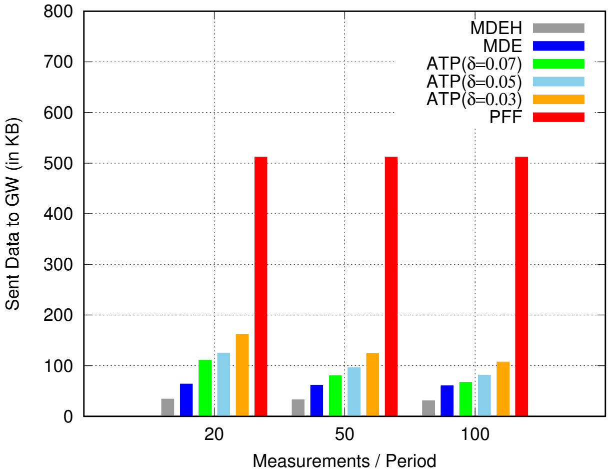

Figure 3. The quantity of sent Measurements (in KB) to the Sink.

As shown in Figure 3, our proposed MDE/MDEH techniques outperform the other methods and decreased the quantity of sent measures to a maximum of 12.5% and 39.4% using MDE, while MDEH decreased the quantity to a maximum of 6.7% and 21% compared to PFF and ATP respectively. The explanation for this is due to the procedure used in the encoding. Also, MDEH optimizes the performance more than MDE, where MDEH uses a dictionary of fixed size for encoding the differences between successive measurements using as low as possible bits.

The cumulative energy consumption of the entire network is total energy dissipation. The radio model of the first-order [33], [34], as in Equation 2, is MDE/MDEH's model of energy usage for transmitting a κ-bit packet with a distance of ν.

$$E_{TX}(\kappa,\nu) = E_{elec} \times \kappa + \beta_{amp} \times \kappa \times \nu^2 \tag{2}$$

Where, E_{elec} is the energy dissipated by the transmitter/receiver and β_{amp} represents the amplification energy for free space. The analytical findings, as shown in Figure 4, the energy absorbed by MDE is decreased up to 66% and 59% while MDEH is decreased up to 82% to 76% compared to PFF and ATP, respectively. Also, note that the MDEH dictionary greatly decreases energy usage in comparison with MDE by 46%.

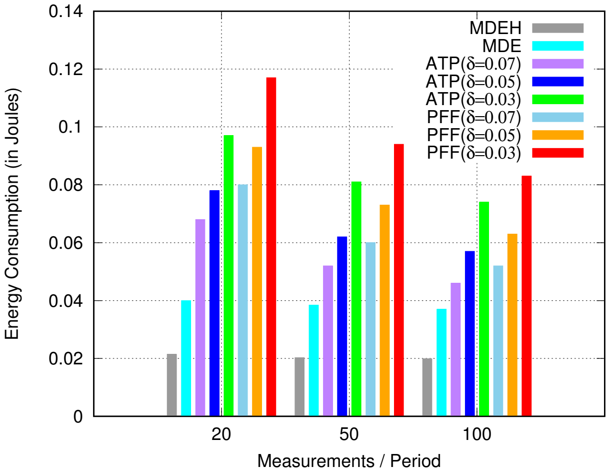

Figure 4. Total Energy Consumption.

The analytical findings, as shown in Figure 4, disclose that the energy consumed by MDE is reduced to a maximum of 66% and 59% while MDEH is reduced to a maximum of 82% to 76% compared to PFF and ATP, respectively. Also, note that the MDEH dictionary greatly decreases energy usage in comparison with MDE by 46%.

The C_R compression ratio is utilized in this research as another criterion for measuring the efficiency of the proposed techniques. The criterion calculates using Equation 3.

$$C_R = 100 \times \left(1 - \frac{C}{\mathcal{M}}\right)\% \tag{3}$$

Where C represents the compressed data set and $\mathcal{M}$ is the original set. MDE/MDEH offers improved efficiency in almost all instances as regards the mean of ratios of compression as seen in Figure 5. The analysis findings disclose that when applying ATP, MDE and MDEH respectively, the compression ratio achieved is 83%, 87.8% and 93.7%.

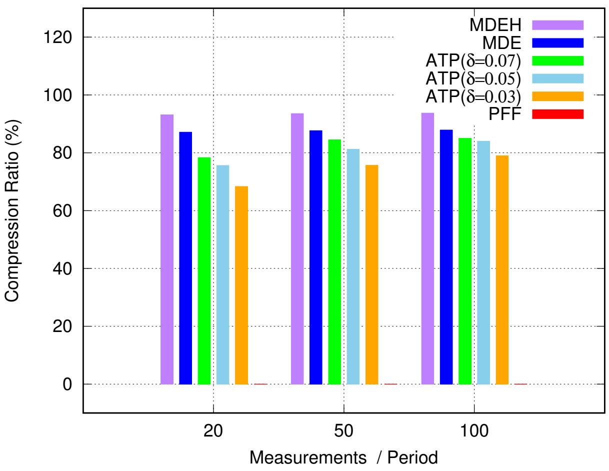

Figure 5. Total Compression Ratios.

We have used another metric in our studies to show the feasibility of the techniques suggested, which is the precision of the data reconstructed. According to this study, the accuracy of the data is specified as the percentage of data lost or not obtained in the sink as a result of operations of aggregation or compression within each sensor node. The results of experiments regarding the accuracy of the data are illustrated in Figure 6.

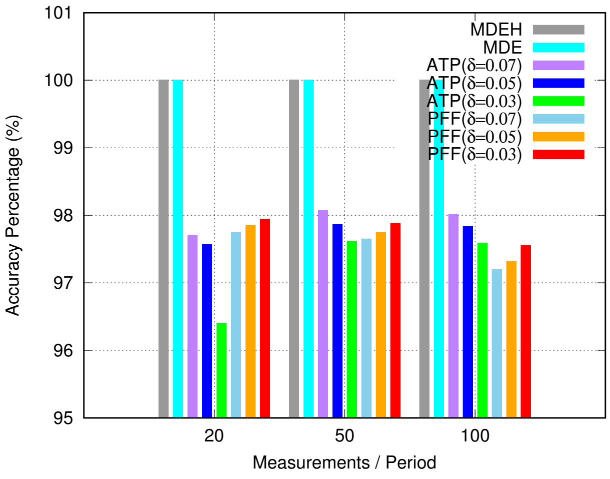

Figure 6. Decompressed Measurement Accuracy.

In all scenarios, MDE/MDEH offers good performance concerning the accuracy of data in comparison with ATP and PFF, as displayed in Fig. 6. The empirical findings disclose that when implementing our suggested lossless MDEH, MDE techniques, the accuracy of reconstructed data is 100%. Unlike PFF and ATP, in which some data is lost.

B. Comparing with Lossless Algorithms

We assess the efficiency of MDE / MDEH in this section with lossless compression algorithms explicitly designed for WSNs. The Two-Modal algorithm, the algorithm ND-Encoding and the LEC algorithm were specifically taken for comparison. In these experiments, the temperature measurements from Intel and for three different sensor nodes as in [26] are taken for comparison. Table IV presents simple descriptive statistics regarding the sensor's measurements used in the comparison.

TABLE IV. Descriptive statistics regarding the sensors measurements.

Node ID	Physical parameter	Min	Max	Mean	Std
3	Temperature (°C)	17.6	26.4	21.2	2.4
8	Temperature (°C)	17.2	26.5	20.9	2.3
19	Temperature (°C)	16.2	28.9	21.7	3.2

A set of 10,000 temperature measurements divided into 100 packets each include 100 words defined by $\mathcal{W} = 16$ bits is considered for each node. The ratios acquired for compression are displayed in Figure 7 and Figure 8. MDEH offers better results concerning mean and maximum ratios of compression in all scenarios as seen in Figure 7 and Figure 8.

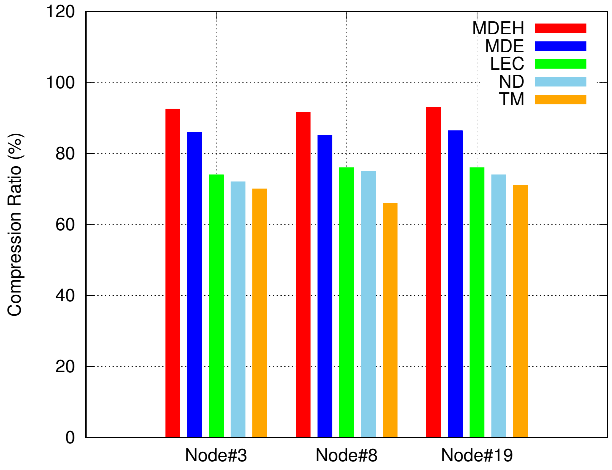

Figure 7. Mean Compression Ratios.

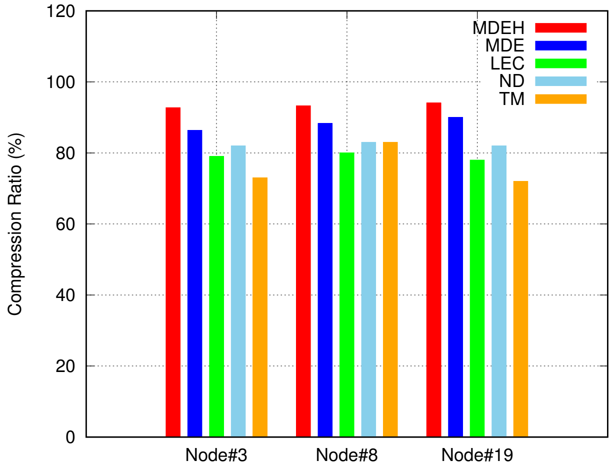

Figure 8. Maximum Compression Ratios.

Through experiments, it was found that there is a direct relationship between the compression ratio and energy-saving. Note also that the new MDEH dictionary greatly improves the mean and maximum compression ratio as opposed to MDE. In comparison, the great difficulty of TM methods (e.g., linear prediction and optimum encoding) does not substantially increase the compression ratio as seen in the figure.

To highlight the generality of the technique proposed, we used some other data sets randomly, rather than the sensor 1 data set, to assess the efficiency of the proposed technique if we used another set for creating the dictionary. The findings of the assessment are illustrated in Table V.

The length's average of the symbol (L), that produced if we used the data set at the top for generating the dictionary to compress data set on the left side, is illustrated in the rows of the table. The mean length of the symbol for all selected data sets is displayed in the last row. As appear

TABLE V. The length's average of the symbol (L) when compressing the measurements utilizing dictionaries created from another set.

	Sensor 1	Sensor 14	Sensor 28	Sensor 35	Sensor 40	Sensor 51	σ
Sensor 1	1.512	1.519	1.617	1.547	1.589	1.652	0.056
Sensor 14	1.512	1.505	1.617	1.546	1.526	1.603	0.047
Sensor 28	1.813	1.813	1.792	1.89	1.932	2.051	0.098
Sensor 35	1.104	1.104	1.414	1.104	1.104	1.112	0.125
Sensor 40	1.26	1.26	1.533	1.267	1.281	1.316	0.106
Sensor 51	1.12	1.12	1.428	1.12	1.13	1.128	0.124
Avg.	1.38	1.38	1.56	1.41	1.42	1.47	

that the length's average of the symbol for all data sets is fairly consistent. The symbol length standard deviation σ is displayed in the last column. Generally speaking, the variation is very minimal as shown in the table.

In terms of the proposed method's complexity, it's worth noting that only counting operations are needed for implementation, and the iterations number required for execution is in the order of $O(N)$. Furthermore, only the compressed set C must be stored in terms of storage requirements. As a result, storage requirements are restricted.

The GW should be knowing the value of the minimum measurement δ in each received set to recreate the original collection $\mathcal{M}$ in a successful manner; hence, this value must encode and transmit in addition to the compressed sequence, resulting in an $h - bit$ overhead. It is necessary to obtain the value of δ, if δ is restricted to be a power of two, i.e., $\delta = 2^{\beta-1}$. We have $h = \lceil log_2\ \beta \rceil = O(log_2(log_2)(N))$ bits in this instance, which is a very minimal overhead.

ATP algorithm has an $O(N^2)$ time complexity. Finally, PFF has a time complexity of $O(N \times log_2(N))$. Furthermore, the message complexity in our proposed method is primarily determined by the number of collected data (N) in the period, which is determined by the application. If a large value for N is required, several solutions, such as data packet division, can be used.

C. Comparing with Another Dataset

To demonstrate the efficacy of the approaches we've conducted other simulation experiments using another real data set as in [35]. This dataset represents meteorological readings captured periodically every hour for 6 years (for the period 1/1/2010 to 31/12/2015) in five different Chinese cities. Liang et al. measure data in a very different way from [32]. Since it is collected at quite a lower recurrence and in a setting in which the physical phenomena being tested have a wider range of variance, data collected in [35] has a higher variance in measured values. We take a time period of 18 and 42 slots in both datasets of five cities and Intel, we found the range of maximum and minimum differences in [35] are 18 and 1, while in [32] are 2.78 and 0.02 respectively. Table VI shows the statistical characteristics of the five cities dataset used in these experiments.

In Figure 9, we can see the amount of data remaining after performing the compression based on the MDE and MDEH methods. From the figure we conclude the following things:

- The results show that the greater the amount of variation in the data will lead to data compression by a lesser degree.
- The amount of captured data is affecting the percentage of remaining data, as the amount of data remaining increases with the increase in the captured data.
- The remaining data percentages range from 23.6-90.9% and from 18.6-81.1% for MDE and MDEH respectively.

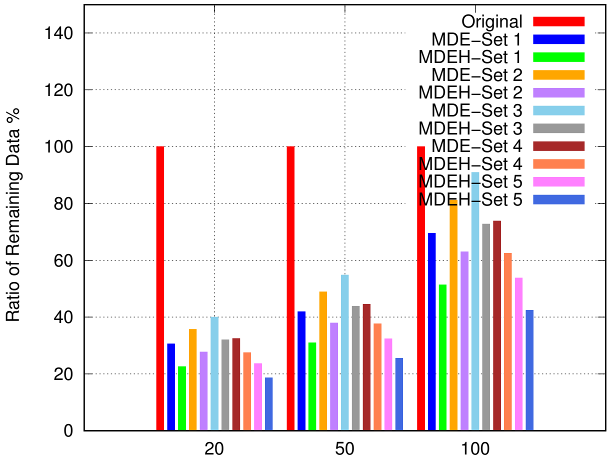

Figure 9. Ratio of remaining data using five cities datasets.

Figure 10 depicts the quantity of data (in KB) sent to the GW when the MDE and MDEH methods are employed by the sensor nodes.

The results shown in Figure 10 can be summarized as follows:

- The more data collected by the sensor nodes; the more data is sent to the GW.
- The greater the amount of variance between the data, the greater the amount of data sent to the GW.

TABLE VI. The Key Features of the Temperature Datasets.

	Location	Range	Mean	Samples	Sampling Interval
Set 1	Beijing	-19 to 42	12.58	52585	60 Min
Set 2	Chengdu	-3 to 38	17.69	52585	60 Min
Set 3	Guangzhou	2 to 38	21.99	52585	60 Min
Set 4	Shanghai	-5 to 41	17.47	52585	60 Min
Set 5	Shenyang	-28 to 35	8.64	52585	60 Min

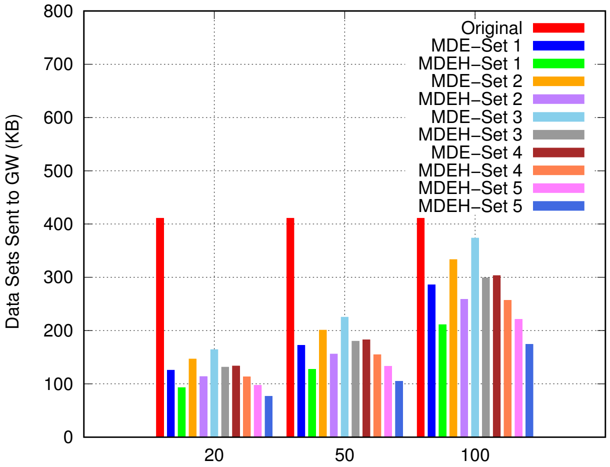

Figure 10. Ratio of sent data sets using five cities datasets.

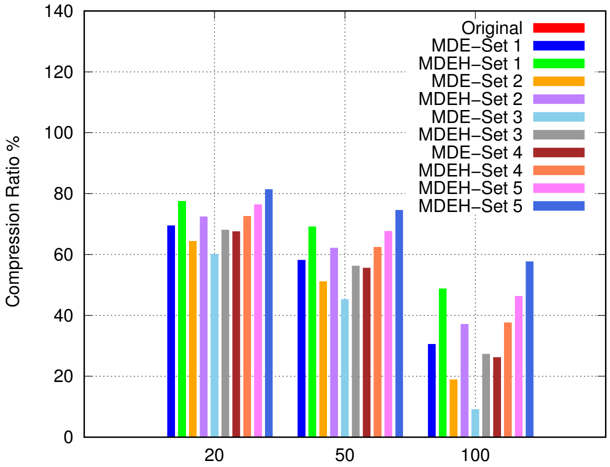

Figure 11. Compression Ratio of data using five cities datasets.

- Using MDEH reduces the amount of data sent by a higher percentage than the MDE method due to the use of the fixed dictionary.
- The saving percentage (that is, the amount of data captured but not sent to the GW) that we obtained is from 23.6-90.9% and from 18-72% using MDE and MDEH methods, respectively.

The main goal of data compression is to reduce the amount of data. In Figure 11, we can see the amount of data remaining after performing compression based on the MDE and MDEH methods.

From the figure we conclude the following:

- The results show that the greater the amount of variation in the data, the less compression will be given to the data.
- The amount of data captured is affecting the percentage of data compression, as the amount of data remaining increases with the increase in the captured data.
- The data compression ratios range between 9-76% and 27-81% for MDE and MDEH respectively.

Finally, with regard to the accuracy, the proposed methods maintain the accuracy of the data completely, and when decompressing it, the accuracy ratio is 100%, as the large variance between measurements does not affect its accuracy, but rather affects the amount of data that will be sent to the GW.

5. Discussion

This section focuses further on the suggested method by exploring the viability of applying it under the limitations and requirements that the application faces. To ensure the correctness of the assumptions of the proposed methods, we used measurements of integers that have lower data variance. If the measurements in the dataset are real, we round it before applying the methods under-study.

For real measurement processing, we need a certain mechanism to define the integer part and fractional part of the number. This requires the use of additional bits, or the use of codes of fixed length and not variable length as is the case in Huffman encoding. Or it is possible to reduce the resolution of the measurements to one or two decimal digits according to the requirements of different applications, and then suggest the generation of a fixed dictionary for them. It will also lead to a noticeable reduction in the amount of data that will be sent and the amount of energy spent, which leads to prolonging the life of the network.

From the simulation results obtained and presented in the fourth section, it becomes evident that the proposed methods work well with data of the integer type in which the variance is small. While it gives acceptable results to some extent with the data that have a large variance. Thus, it depends on the type of application and the nature of the data. The number of data collected in a period and the range of data variance are two significant factors in the efficiency of our techniques. For example, take into account a system which measures data with a greater variation, such as outdoor temperature readings over a broad variety of frequencies, than a system which measures the temperature of a living being. In comparison to the application that measures outdoor temperature, the application that measures body temperature would generate the same or similar readings many times in each period.

Finally, data compression is an energy-saving technique. The network's load is the data in several respects. Evidently, the network's whole life is focused on the monitoring of data of interest. The load, packet, or parcel that the network plans to collect and transmit to the GW is known as data. As a result, data compression at sensor nodes reduces network load, thus extending network lifespan.

6. Conclusions and Future Work

In this research, we proposed a lightweight lossless compression algorithm based on Differential Encoding (DE) and Huffman techniques which is particularly beneficial for IoT sensor nodes, that applied in WSN for agriculture to monitor any irregular meteorological data situation that could damage farming. Instead of trying to formulate innovative ad hoc algorithms, we demonstrate that, provided general awareness of the features to be monitored, classical Huffman coding can be used effectively to describe the same features that measure at various time periods and locations. Owing to the very low computing and memory requirements of the proposed system, it can be conveniently used in practical WSNs. Results utilizing temperature measurements indicate that it outperforms common methods developed especially for WSNs, even though the suggested system does not reach the theoretical maximum.

Acknowledgment

The author wishes to express his gratitude to the University of Babylon in Iraq for their support.

References

[1] A. K. M. Al-Qurabat and A. Kadhum Idrees, "Data gathering and aggregation with selective transmission technique to optimize the lifetime of internet of things networks," *International Journal of Communication Systems*, vol. 33, no. 11, p. e4408, 2020.

[2] I. D. I. Saeedi and A. K. M. Al-Qurabat, "A systematic review of data aggregation techniques in wireless sensor networks," in *Journal of Physics: Conference Series*, vol. 1818, no. 1. IOP Publishing, 2021, p. 012194.

[3] A. K. M. Al-Qurabat, A. K. Idrees, and C. Abou Jaoude, "Dictionary-based dpcm method for compressing iot big data," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 1290–1295.

[4] A. K. M. Al-Qurabat, C. Abou Jaoude, and A. K. Idrees, "Two tier data reduction technique for reducing data transmission in iot sensors," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2019, pp. 168–173.

[5] C. Salim and N. Mitton, "Machine learning based data reduction in wsn for smart agriculture," in *AINA 2020-34th International Conference on Advanced Information Networking and Applications*, 2020.

[6] T. N. Gia, L. Qingqing, J. P. Queralta, Z. Zou, H. Tenhunen, and T. Westerlund, "Edge ai in smart farming iot: Cnns at the edge and fog computing with lora," in *2019 IEEE AFRICON*. IEEE, 2019, pp. 1–6.

[7] K. P. Musaazi, T. Bulega, and S. M. Lubega, "Energy efficient data caching in wireless sensor networks: A case of precision agriculture," in *International Conference on e-Infrastructure and e-Services for Developing Countries*. Springer, 2014, pp. 154–163.

[8] K. Hossain, M. Rahman, and S. Roy, "Iot data compression and optimization techniques in cloud storage: current prospects and future directions," *International Journal of Cloud Applications and Computing (IJCAC)*, vol. 9, no. 2, pp. 43–59, 2019.

[9] S. A. Abdulzahra, A. K. M. Al-Qurabat, and A. K. Idrees, "Data reduction based on compression technique for big data in iot," in *2020 international conference on emerging smart computing and informatics (ESCI)*. IEEE, 2020, pp. 103–108.

[10] A. K. M. Al-Qurabat and A. K. Idrees, "Two level data aggregation protocol for prolonging lifetime of periodic sensor networks," *Wireless Networks*, vol. 25, no. 6, pp. 3623–3641, 2019.

[11] A. K. Idrees, A. K. M. Al-Qurabat, C. Abou Jaoude, and W. L. Al-Yaseen, "Integrated divide and conquer with enhanced k-means technique for energy-saving data aggregation in wireless sensor networks," in *2019 15th International wireless communications & mobile computing conference (IWCMC)*. IEEE, 2019, pp. 973–978.

[12] A. K. M. Al-Qurabat and A. K. Idrees, "Energy-efficient adaptive distributed data collection method for periodic sensor networks," *International Journal of Internet Technology and Secured Transactions*, vol. 8, no. 3, pp. 297–335, 2018.

[13] A. K. M. Al-Qurabat and A. Idrees, "Distributed data aggregation and selective forwarding protocol for improving lifetime of wireless sensor networks," *Journal of Engineering and Applied Sciences*, vol. 13, no. 5 S1, pp. 4644–4653, 2018.

[14] A. K. M. Al-Qurabat and A. K. Idrees, "Distributed data aggregation protocol for improving lifetime of wireless sensor networks," *Qalaai Zanist Scientific Journal*, vol. 2, no. 2, pp. 204–215, 2017.

[15] A. K. Idrees and A. K. M. Al-Qurabat, "Distributed adaptive data collection protocol for improving lifetime in periodic sensor networks." *IAENG International Journal of Computer Science*, vol. 44, no. 3, 2017.

[16] A. K. Al-Quraba and A. K. Idrees, "Adaptive data collection protocol for extending lifetime of periodic sensor networks," *Qalaai Zanist Scientific Journal*, vol. 2, no. 2, pp. 93–103, 2017.

[17] S. A. Abdulzahra, A. K. M. Al-Qurabat, and A. K. Idrees, "Compression-based data reduction technique for iot sensor net-

works," *Baghdad Science Journal*, vol. 18, no. 1, pp. 0184–0184, 2021.

[18] A. K. Idrees and A. K. M. Al-Qurabat, "Energy-efficient data transmission and aggregation protocol in periodic sensor networks based fog computing," *Journal of Network and Systems Management*, vol. 29, no. 1, pp. 1–24, 2021.

[19] A. K. M. Al-Qurabat and S. A. Abdulzahra, "An overview of periodic wireless sensor networks to the internet of things," in *IOP Conference Series: Materials Science and Engineering*, vol. 928, no. 3. IOP Publishing, 2020, p. 032055.

[20] G. A. M. Jawad, A. K. M. Al-Qurabat, and A. K. Idrees, "Compression-based block truncation coding technique to enhance the lifetime of the underwater wireless sensor networks," in *IOP Conference Series: Materials Science and Engineering*, vol. 928, no. 3. IOP Publishing, 2020, p. 032005.

[21] T. Schoellhammer, B. Greenstein, E. Osterweil, M. Wimbrow, and D. Estrin, "Lightweight temporal compression of microclimate datasets," in *29th Annual IEEE International Conference on Local Computer Networks*. IEEE, 2004.

[22] E. P. Capo-Chichi, H. Guyennet, and J.-M. Friedt, "K-rle: a new data compression algorithm for wireless sensor network," in *2009 Third International Conference on Sensor Technologies and Applications*. IEEE, 2009, pp. 502–507.

[23] K. Römer, "Discovery of frequent distributed event patterns in sensor networks," in *European conference on wireless sensor networks*. Springer, 2008, pp. 106–124.

[24] C. M. Sadler and M. Martonosi, "Data compression algorithms for energy-constrained devices in delay tolerant networks," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, 2006, pp. 265–278.

[25] F. Marcelloni and M. Vecchio, "A simple algorithm for data compression in wireless sensor networks," *IEEE communications letters*, vol. 12, no. 6, pp. 411–413, 2008.

[26] X. Ren and D. Fang, "A normal distribution encoding algorithm for slowly-varying data compression in wireless sensor networks," in *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*. IEEE, 2010, pp. 1–4.

[27] Y. Liang and W. Peng, "Minimizing energy consumptions in wireless sensor networks via two-modal transmission," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 12–18, 2010.

[28] H. Harb, A. Makhoul, R. Couturier, and M. Medlej, "Atp: An aggregation and transmission protocol for conserving energy in periodic sensor networks," in *2015 IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, 2015, pp. 134–139.

[29] H. Harb, A. Makhoul, R. Couturier, and M. Medlej, "Atp: An aggregation and transmission protocol for conserving energy in periodic sensor networks," in *2015 IEEE 24th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises*. IEEE, 2015, pp. 134–139.

[30] W. Underground, "Weather underground," 2012. [Online]. Available: http://www.wunderground.com

[31] D. Otto, "Avr-huffman," 2012. [Online]. Available: http://www.das-labor.org/wiki/AVR-Huffman/en

[32] P. Bodik, "Intel berkeley research lab," 2004. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

[33] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd annual Hawaii international conference on system sciences*. IEEE, 2000, pp. 10–pp.

[34] A. K. Idrees, C. Abou Jaoude, and A. K. M. Al-Qurabat, "Data reduction and cleaning approach for energy-saving in wireless sensors networks of iot," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)(50308)*. IEEE, 2020, pp. 1–6.

[35] X. Liang, S. Li, S. Zhang, H. Huang, and S. X. Chen, "Pm2. 5 data reliability, consistency, and air quality assessment in five chinese cities," *Journal of Geophysical Research: Atmospheres*, vol. 121, no. 17, pp. 10–220, 2016.

Ali Kadhum M. Al-Qurabat Ali Kadhum M. Al-Qurabat received his Ph.D. in Software (Wireless Sensor Networks) in 2018 from the University of Babylon, Iraq. He received the M.Sc. degree in Information Technology from Universiti Tenaga Nasional (UNITEN), Malaysia, in 2012. He received the B.Sc. degree in computer science from the University of Babylon, Iraq, in 2002. He joined the Department of Computer Science, University of Babylon, Iraq, in 2006, where he is currently an Assistant Professor. His research interests include WSN, Data Aggregation, IoT, Data Mining, Data Compression and E-Procurement. Dr. Al-Qurabat has been a TPC Member of several national and international networking conferences and a Reviewer for several international journals. He is the author of a great deal of research studies published in national and international journals, and at conference proceedings.