

Study on data compression algorithm and its implementation in portable electronic device for Internet of Things applications

Nor Asilah Khairi^{1,*}, and Asral Bahari Jambek¹

¹School of Microelectronic Engineering, Universiti Malaysia Perlis, Pauh Putra Campus, 02600 Arau, Perlis, Malaysia

Abstract. An Internet of Things (IoT) device is usually powered by a small battery, which does not last long. As a result, saving energy in IoT devices has become an important issue when it comes to this subject. Since power consumption is the primary cause of radio communication, some researchers have proposed several compression algorithms with the purpose of overcoming this particular problem. Several data compression algorithms from previous reference papers are discussed in this paper. The description of the compression algorithm in the reference papers was collected and summarized in a table form. From the analysis, MAS compression algorithm was selected as a project prototype due to its high potential for meeting the project requirements. Besides that, it also produced better performance regarding energy-saving, better memory usage, and data transmission efficiency. This method is also suitable to be implemented in WSN. MAS compression algorithm will be prototyped and applied in portable electronic devices for Internet of Things applications.

1 Introduction

Internet of Things (IoT) is known as a network of physical objects containing embedded technology that communicates and senses or cooperates with internal states or an external environment via wireless and wired connection [1]. The authors in [2] indicated that the IoT could be defined as the next generation of internet evolution, where it can connect everyday objects such as refrigerators, air conditioners, and automotive devices, through embedded wireless devices that allow interaction with each other and provide new services. The existence of IoT creates a seamless intelligence that improves human life quality.

Cisco predicted that 50 billion new connections would be part of the IoT by 2020 [3]. These billions of connected devices will produce a significant amount of data to be stored. The current methods to store data have proven to be insufficient for the demands of the IoT. Therefore, a few papers discussed the possibilities of embedding a compression algorithm into IoT devices or sensor nodes. According to the authors [4], compression is the perfect solution that allows the reduction in both storage requirements and input/output operations. Therefore, it is very useful during the transmission of data via a network.

Wireless Sensor Network (WSN) is popular regarding monitoring the different types of physical environments. However, sensor nodes were designed with limited battery power. Hence, it limits the energy to function and necessitates a low power to save energy and in turn extend the lifetime of the sensor node [4]. Furthermore, radio transmission is a major contribution

to energy consumption in sensor nodes. An efficient data compression approach is highly recommended to overcome this problem. Therefore, several past studies were selected to investigate the concepts of data compression algorithm.

This paper is organized as follows: Section 2 reviews previous compression algorithms; the comparison between different types of data compression is discussed in Section 3; Section 4 focuses on the project proposed of MAS compression algorithm; and finally, conclusions are drawn in Section 5.

2 Latest trend of data compression algorithm

Data compression is a popular issue among researchers due to its ability to be used in nearly every functional area. According to various researches, data compression is divided into two categories: lossless and lossy compressions. The Autoencoders (AE) compression algorithm in [5] is known as a lossy compression. The improvement of the IoT device's lifetime such as energy constrained, optimization of the internal memory space, and efficient data transmission via wireless connection was brought into focus in this experiment, while biomedical signals such as ECG, photoplethysmography, and respiratory traces were used as the dataset. The performance of the AE compression algorithm was then compared with Discrete Cosine Transforms (DCT), Wavelet transforms, Principal Component Analysis (PCA), and linear approximations. Based on the results produced in the paper [5], the

* Corresponding author: asilah.khairi91@gmail.com

energy consumption required for the online compression of the signal is quite small when compared with a scheme using linear approximations. According to researchers, this project also posed a weakness. It requires an offline training phase, which means it has to be executed only once for each class of signals.

Vecchio, Giaffreda, and Marcelloni [6] proposed to endow the lossless compression algorithm (LEC) for wireless sensor networks using two simple adaption schemes that relied on the new concept of the appropriately rotated prefix-free tables. This project focused on improving the power consumption of IoT devices during transmission and the reception of data. According to the researchers, the primary idea of the LEC algorithm relied on dividing the alphabets or numbers into a group in which the sizes will increase rapidly. Hence, the LEC codeword was a hybrid of unary and binary codes, where the unary code (a variable-length code) specified the group and the binary code (a fixed-length code) represented the index within the group that is almost similar to Golomb and Elias's coding. In this project, the researchers introduced two more different adaption blocks known as GA-LEC and FA-LEC. Air temperature, surface temperature, relative humidity, and solar radiation were used as the dataset. Both GA-LEC and FA-LEC compressions produced a high percentage of efficiency.

Domingo-Prieto et al. [7] focused on the relationship between the amounts of messages each one sends that affects the lifetime of the battery in mote. Huffman coding is applied as data compression in this project. The dataset was extracted from the IoT device, and the performance of the Huffman compression algorithm was later compared against address clustering. Based on the results, the motes that send more messages have improved their lifetime compared to the motes that send fewer messages. According to researchers, the implementation of Huffman coding does not seem to be a trivial task in a real deployment, while the Address Clustering is straightforward.

In paper [8], Hsu et al. proposed the development of the WSN system that collects vibration acceleration from bridges in real time and sends the data via the internet. This project not only focused on remote monitoring, but also in achieving the lowest production cost, shortest installing time, and convenient relocation. The Huffman coding was used as the data compression in this project, whereas the vibration frequency of bridge was used as a dataset. The performance of the Huffman compression algorithm was then compared against three other compression algorithms, which included LZ77, Run Length Encoding (RLE), and Rice coding. Based on the results, the number of WSN end device node grew 300% more compared to the similar sampling rate that did not have an algorithm implemented on the WSN network. According to researchers, the wireless transmission payload was reduced to about 60% and the node number of the implemented increased about three times.

According to Assi et al. [9], the MAS compression algorithm is one of the first algorithms to compress floating-point data specifically. In this project, the researchers focused on exploiting the disproportionality

in energy consumption between data transmission and processing. MAS compression algorithm stands for Minimalist, Adaptive, and Streaming. Carbon dioxide, water levels, radioactivity, temperature, humidity, and low sea pressure were used as the dataset in this experiment. The researchers conducted six different types of experiments: compression ratio, computation time and energy, transmission energy, total consumed energy, energy-savings, and memory requirements. The results showed that MAS produced higher compression ratio than the S-LZW and K-RLE compression algorithms. Hence, using MAS algorithm can achieve better results compared to other algorithms. Besides, MAS consumed the least amount of transmitted energy among the other algorithms. Despite K-RLE consumption of the least computation energy, it still consumed the most amount of total energy, as it sent significant amounts of data over the network. Meanwhile, MAS consumed the least energy, thus proving that MAS is the strongest candidate for compression in WSN. It also managed to save more energy than S-LZW and K-RLE. MAS consumed the largest amount of flash memory because it had a longer programming code, with two representations for integers and real numbers. However, it did not cause any problem, since the flash memory was reserved for program code and not for random access. Moreover, MAS consumes only 44 bytes regarding RAM usage. According to researchers, the results of the experiment showed that MAS produced energy-saving on an average of 54%, defeating K-RLE and S-LZW while maintaining the highest compression ratios.

Marcelloni and Vecchio [10] proposed a straightforward and efficient data compression algorithm that is precisely suited to be used on nodes of WSN, where energy, memory, and computational resources were limited. They focused on developing a simple compression algorithm that reduced memory and computational resources of the WSN node. A simple lossless compression algorithm was introduced as data compression in this project, and it used a combination of the Huffman and JPEG algorithms. The researchers used temperature and humidity as the dataset. Based on the results, the proposed compression algorithm achieved higher compression ratio than the S-LZW. Last but not least, it proved that this algorithm could outperform gzip and bzip2.

3 Comparison between different types of compression algorithms

Every single data of compression algorithms is built differently. There are different types of compression algorithms in this world, some of which can support image files, while some can only support text files. Some data compressions cannot preserve the originality of the data after decoding, for example, the Autoencoder algorithm [5]. In Section 3, the data compression algorithms that were previously discussed in Section 2 are analyzed and displayed in a table for comparison.

Table 1 shows the summarization of the comparison between the different types of compression algorithms.

Table 1. Comparison of data compression algorithms.

Description	[5]	[6]	[7]	[8]	[9]	[10]
Year	2015	2014	2014	2014	2013	2008
Focus	Improves lifetime of IoT devices (energy constraint, optimization on internal memory space, and efficient transmission of data).	Improves power consumption of IoT devices	Focuses on the relationship between battery lifetime in mote and number of messages each one sends	Remote monitoring, low cost, short installation time, and convenient relocation	Exploit the disproportionality in energy consumption between data transmission and processing	Simple compression algorithm suited to reduce memory and computational resources of WSN node
Types of Data Compression	Autoencoder (AE)-Lossy Compression	GAS-LEC, FAS-LEC	Huffman coding	Huffman coding	Minimal, Adaptive, and Streaming compression algorithm (MAS)	Simple lossless compression algorithm - Combination of Huffman, and JPEG coding
Comparison of Other Data Compression	Discrete Cosine Transform (DCT), DCT-LW, Discrete Wavelet Transform (DWT), DWT-LW, LTC (based on linear approximation), Principal Component Analysis (PCA)	Lossless Compression Algorithm (LEC)	Address Clustering (non-data compression)	LZ77, Run Length Encoding (RLE), Rice	S-LZW, K-RLE	S-LZW, gzip, bzip2
Dataset	Biomedical signal (ECG, photoplethysmographic, respiratory traces)	Air, surface, temperature, solar radiation, relative humidity	Extracted from the IoT device deployment	Vibration frequency of bridge	Carbon dioxide, water levels, radioactivity, temperature, humidity, sea level pressure	Temperature, humidity
Advantage	Small reconstruction, high compression efficiency, low	Improves the compression efficiency of LEC	Has better overall improvement because it used	Wireless transmission payload can be reduced by 60%	Better regarding compression ratio, compression speed, memory	Higher compression ratio, com

	energy consumption		variable-size packets and also compressed data	node number of the implemented network can be increased by three times	requirements, and energy-saving	
Disadvantage	Requires an offline training phase (has to be executed only once for each class of signals)	Not stated	Does not seem a trivial task in real deployment	Not stated	Not stated	Not stated

Based on the summarization in Table 1, project [9] has the most potential and a more suitable project to be used as a prototype of this experiment. The similarities of the project [9] with past studies can be seen in the advantage, method, and focus. Project [9] produced better compression ratio, which is also similar to the project [10]. Regarding its method, MAS project concentrated on the implementation of the data compression algorithm in WSN. This approach is similar to projects [5], [6], [7], [8], and [10]. The last similarity of the project [9] was that it focused on improving the performance of the data compression. Project [9] was focused on reducing the usage of power consumption during data transmission, which was similar to projects [5], [6], and [7].

Paper [9] was selected as the reference paper not only for its similarities but also for its uniqueness. One of it is the data compression algorithm. This algorithm is one of the first algorithms to compress, floating-point data specifically. Besides that, it supported not only the floating-point data but also integer number data. Another factor that made this paper unique is its use of measurement other than the compression ratio, which included compression speed, memory requirement, energy-saving, and energy cost. The measurements of the project [9] are not similar to the other projects. Therefore, it produces various results that will ease the observation process.

Based on the analysis, paper [9] was chosen as the reference project due to its high similarity concept to the one proposed. However, this project required certain adjustments to meet the IoT concept. The recommendation of the work was the implementation of the compression algorithm in the end devices and the end user devices. The structure of the MAS algorithm required minor adjustments to allow the MAS compression to be implemented inside the end devices and MAS decompression inside the end user devices. Different types of collected data sets were suggested in this project with the intention of producing various kinds of results. Furthermore, it was also done to test the compatibility of MAS algorithm with the devices.

4 Proposed MAS compression algorithm in IoT Device

Once the description of the data compressions was compared with each other in Section 3, the suitable compression algorithm was selected to be used as a reference and prototype for this project. On that regard, the MAS compression algorithm was chosen as a prototype project due to its potential to meet the project requirement.

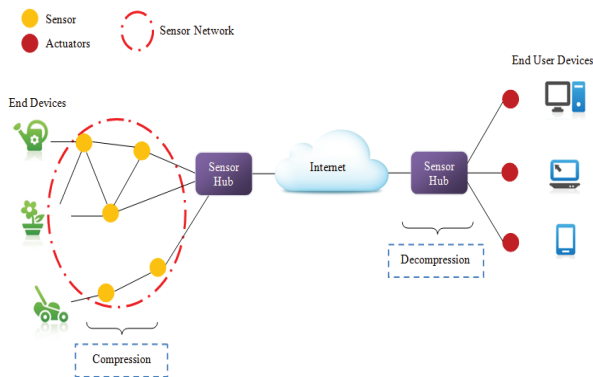


Fig. 1. MAS data compression framework for IoT device.

The proposed data compression framework for IoT device is illustrated in Figure 1, which consisted of the compression and decompression processes in the sensor network and the sensor hub. The figure shows that the watering can, flowering plant, and lawnmower are considered as the end devices. However, other relevant items can also be used as the end devices.

According to the authors in [1], the sensor network is the first phase where all the data are generated. The sensors collect the data from the end devices and compress the data with the intention of reducing power consumption and memory space while increasing the speed of the data transmission. After the compression process completed, the data are sent to the sensor hub via a wireless sensor network. Later, this modem redirects the data via the internet using basic protocols.

The second sensor hub receives data from the first sensor hub through the internet. The modem then decompresses the data and triggers the actuators on the other side. The intention of the data decompression is to convert the data into its original form. Any smart gadgets such as smartphones, tablets, laptops, and desktop computers are classified as the end user devices. After the decompression process is completed, the data is displayed on the end user devices.

5 Conclusion

IoT devices are always powered by a small battery, and this battery does not usually last long. As a result, saving energy in IoT devices has become the main focus of this paper. Radio communication is known for its high power consumption. Therefore, several researchers have proposed data compression algorithms with the sole purpose of overcoming this particular problem. Several

compression algorithms in previous reference papers were analyzed and discussed in this paper. The descriptions of previous compression algorithms were collected and tabulated. Based on the analysis, the MAS compression algorithm was selected as the project prototype due to its high potential in meeting the requirements such as high energy-saving, low memory usage, and efficient data transmission. Besides, this method is suitable to be implemented in WSN. This proposed algorithm will be prototyped and applied in portable electronic devices for Internet of Things applications.

References

1. P. Srivastava and N. Garg, *2015 International Conference on Computing, Communication & Automation (ICCCA)*, pp. 720-723 (2015)
2. Z. Wang, R. Bie, and M. Zhou, *2012 IEEE 11th International Conference on Privacy in Computing and Communications (TrustCom)*, pp. 1799-1804 (2012)
3. S.E. Collier, *2015 IEEE Rural Electric Power Conference (REPC)*, pp. 65-68 (2015)
4. S. Sheikh and H. Dakhore, *International Journal of Computer Science and Information Technologies (IJCSIT)*, **6**, 1, pp. 818-821 (2015)
5. D.D. Testa and M. Rossi, *IEEE Signal Processing Letters*, **22**, 12, pp. 2304-2308 (2015)
6. M. Vecchio, R. Gaffreda and F. Marcelloni, *IEEE Transactions on Wireless Communications*, **13**, 2, pp. 1088-1100 (2014)
7. M. Domingo-Prieto, B. Martinez, M. Monton, I. Vilajosana-Guillen, X. Vilajosana-Guillen and J. Arnedo-Moreno, *2014 Eighth International Conference on Complex, Intelligent and Software Intensive Systems (CISIS)*, pp. 170-176 (2014)
8. C-H. Hsu, C-T. Lin, H-P. Tserng and J-Y. Han, *2014 IEEE World Forum on Internet of Things (WF-IoT)*, pp. 548-552 (2014)
9. M.E. Assi, A. Ghaddar, S. Tawbi and G. Fadi, *International Journal of Ad hoc, Sensor & Ubiquitous Computing (IJASUC)*, **4**, 5, pp. 13-28 (2013)
10. F. Marcelloni and M. Vecchio, *IEEE Communications Letters*, **12**, 6, pp. 411-413 (2008)