Performance Analysis of K-RLE Based Data Compression Algorithm for Internet of Things (IoT) Application

N. A. Khairi[1], A. B. Jambek[2]

[1,2]Faculty of Electronic Engineering Technology, Universiti Malaysia Perlis, Kampus Alam UniMAP, Pauh Putra, 02600 Arau, Perlis, Malaysia

*Corresponding author's email: asral@unimap.edu.my

ABSTRACT: The increased use of Internet of Things (IoT) devices has created considerable data in the network. As a result, insufficient data storage and heavy traffic in the network may arise. Selecting suitable data compression for IoT devices is essential to improve the compression ratio and reduction in computational complexity. Therefore, a lightweight compression algorithm implementation is the main focus of this study. Three proposed algorithms, namely Modified K-RLE, Delta + K-RLE, and K-RLE + Huffman, are studied in this work. Based on the experimental results, K-RLE + Huffman achieved the best compression ratio of 0.51 among the three candidates, while Delta + K-RLE achieved the lowest computational time of 0.027s when processing 100000 datasets. For overall balance performance, Modified K-RLE is generally a good candidate where it offers a lossless compression ratio of 0.825 and an execution time of 0.054s when processing 100000 datasets.

Keywords: *Data compression, Modified K-RLE, compression ratio.*

1. INTRODUCTION

Internet of Things (IoT) is a future computing concept that leads to the increase of internet users every year. The Cisco company predicted 50 billion new IoT connections by 2020 [1][2]. These billions of IoT devices have created a large number of data in the network [3].

Implementing the data compression algorithm on sensor nodes helps to achieve better performance in Wireless Sensor Network (WSN) [4]. Data compression can reduce storage capacity and energy usage during data transmission [5].

Many factors need to be considered before selecting a suitable data compression algorithm since different sets of data have different characteristics. Thus, choosing a suitable compression algorithm is crucial to achieving better power consumption [6]. An algorithm with high computational complexity can cause long code execution time and high energy usage.

This paper focuses on analyzing the compression performance and execution time of the K-RLE based data compression algorithms. This study used three performance measurements: compression ratio, time complexity, and code execution time.

This article is organized as followed: Section II describes the proposed algorithms while Section III focuses on the experimental results of proposed algorithms. Section IV summarizes our findings.

2. METHODOLOGY

RLE algorithm is a loss-less data compression where original data can be recovered fully after decompression. The data is compressed by removing redundancy in data and replacing it with a shorter symbol [7]. To increase the compression ratio, K-RLE is used. The algorithm replaces the n data occurrences with a single pair of symbols nd, where d is the data value between d + K and d - K [8]. K-RLE can achieve better compression at the expense of some data loss (i.e., lossy data compression).

To minimize the data losses after decompression using the K-RLE algorithm, this work proposes a Modified K-RLE (MKRLE) algorithm. Fig. 1 shows the pseudocode of the MKRLE algorithm. The squared area in Fig.1 is the proposed improvement that introduces marker notation. Three markers are used in this proposed algorithm which are 'a', 'b.' and 'c'. Table 1 summarize the MKRLE operation.

Delta + K-RLE (DKRLE) is the second proposed algorithm that implements the double data compression technique. Delta encoding algorithm is executed first to compress the differences between the sequential data. Then, the K-RLE algorithm is used to compress the output from the Delta algorithm. The third proposed algorithm is K-RLE + Huffman (KRLEH). For this algorithm, K-RLE is performed first. Then, the output from the K-RLE is compressed using the Huffman algorithm.

DKRLE and KRLEH are lossy methods among the proposed algorithms since some data cannot be recovered after decompression. On the other hand, MKRLE is considered a loss-less method where the original data can be fully recovered after decompression.

3. RESULT AND DISCUSSION

Three climate datasets are used in this work, which consists of measurement data for temperature, humidity, and sea-level pressure [15]. Compression ratio refers to the ratio between compressed data and uncompressed data [16]. A smaller compression ratio means better compression performance.

Fig. 2 shows the compression ratio for the proposed algorithms. The best compression performance goes to KRLEH (0.513), followed by MKRLE (0.825) and

DKRLE (0.877). The analysis of the compression algorithms on time complexity is shown in Table 2. Based on the analysis, DKRLE has the lowest time complexity compared to the other two methods, where the time complexity for DKRLE is O(n) and O(n2) for compression and decompression, respectively. The MKRLE and KRLEH have a higher time complexity of O(n3). Hence, these algorithms spend more time compressing and decompressing the data.

The execution time for each algorithm is plotted as shown in Figure 3. This experiment is run on 1000, 5000, 10000, 50000, and 100000 data. These five data sets are executed three times, and the average execution times are taken in this study. Based on the graph, DKRLE achieved the lowest computational time, followed by MKRLE and KRLEH, with the time taken at 100000 datasets are 0.0269s, 0.0542s, and 0.104s, respectively.

4. CONCLUSION

The increased number of Internet of Things (IoT) devices has created a considerable amount of data in the network that can cause insufficient data storage and heavy traffic. Therefore, selecting a suitable data compression algorithm is highly recommended to achieve a better compression ratio and low computational complexity. Three algorithms are proposed in this study: MKRLE, DKRLE, and K-RLE + Huffman. Based on the experimental results, KRLEH achieves a better compression ratio (0.513) than the other two algorithms, while DKRLE achieved the lowest execution time of 0.0269s for 100000 datasets. MKRLE can offer optimal compression performance with reasonable execution time with zero data loss for overall balance performance.

Modified K-RLE algorithm

```
Read data inside text file
while not end of file
    if index == 0 then
        Store current reading value inside temporary int. variable and array
    else if current value – temporary int. variable ≥ -1 AND current value – temporary int.
    variable ≤ 1 then
        Count repetitive value
        if current value – temporary int. variable == 1 then
            Store character 'a' inside array
        else if current value – temporary int. value == -1 then
            Store character 'b' inside array
        else
            Store character 'c' inside array
        end if
    else
        Store repetitive value inside array
    end if
    Store current reading value inside temporary int. variable
    index = index + 1
end while
```

Figure 1. Pseudocode of modified K-RLE

Table 1: Example of Modified K-RLE algorithm

Original Stream	: 26 25 25 25 27 28 28 29 29 29 27 27 26 29 28 30 31 29
RLE	: 26 25 -2 27 28 28 29 -2 27 27 26 29 28 30 31 29
K-RLE	: 26 -4 28 -6 26 29 -2 31 29
Modified K-RLE	: 26 b3 a 28 c a3 c2 26 29 b a 31 29

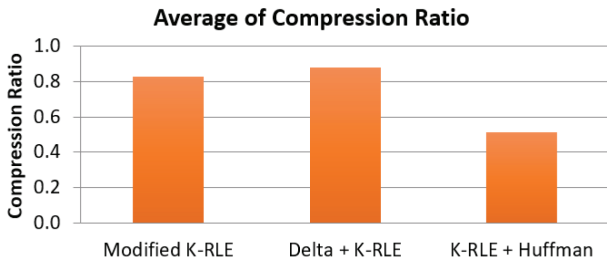

Figure 2: Average compression ratio results for eleven methods

Table 2: Time complexity analysis of the proposed algorithms

Algorithm	*Compression*	*Decompression*	*Overall*
MKRLE	$O(n^2)$	$O(n^3)$	$O(n^3)$
DKRLE	O(n)	$O(n^2)$	$O(n^2)$
KRLEH	$O(n^2)$	$O(n^3)$	$O(n^3)$

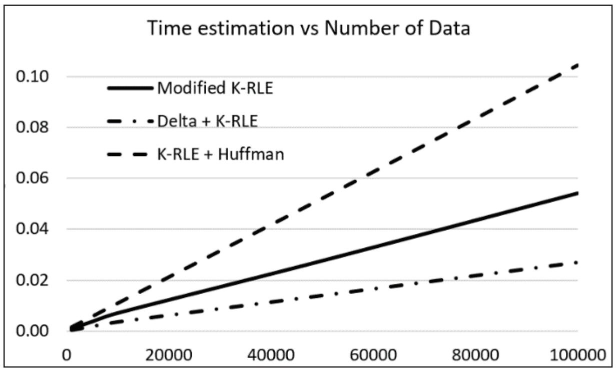

Figure 3: Time complexity graph

REFERENCES

[1] D. Bremner, "Analysing the IoT Ecosystem: The Barriers to Commercial Traction," in embedded world Exhibition and Conference 2016.

[2] S. E. Collier, "The Emerging Enernet: Convergence of the Smart Grid with the Internet of Things," in 2015 IEEE Rural Electric Power Conference, 2015, vol. 23, no. 2, pp. 65–68.

[3] I. Lee, "Big data: Dimensions, Evolution, Impacts, and Challenges," Business Horizons, vol. 60, no. 3, pp. 293–303, May 2017.

[4] M. Z. Gashti, Y. S. Gasimov, and G. Farjamnia, "New Study the Required Conditions for using in Compression WSNs During the Data Collection," International Journal of Mechatronics, Electrical and Computer Technology (IJMEC), vol. 8, no. 27, pp. 3725–3735, 2018.

[5] P. Kaur, S. Kaur, and A. Kaur, "Big Data Compression in Mobile and Pervasive Computing," International Journal of Wireless and Microwave Technologies, vol. 6, no. 3, pp. 1–8, May 2016.

[6] T. Srisooksai, K. Keamarungsi, P. Lamsrichan, and K. Araki, "Practical Data Compression in Wireless Sensor Networks: A Survey," Journal of Network and Computer Applications, vol. 35, no. 1, pp. 37–59, Jan. 2012.

[7] Tiwari, B., & Kumar, A. (2012). Aggregated Deflate-RLE compression technique for body sensor network. In 2012 CSI 6th International Conference on Software Engineering, CONSEG 2012.

[8] Capo-Chichi, E. P., Guyennet, H., & Friedt, J.-M. (2009). K-RLE: A New Data Compression Algorithm for Wireless Sensor Network. In *2009 Third International Conference on Sensor Technologies and Applications* (pp. 502–507).