Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ANALYSIS AND MODIFICATION OF RICE GOLOMB CODING LOSSLESS COMPRESSION ALGORITHM FOR WIRELESS SENSOR NETWORKS

[1]S.KALAIVANI, [2]Dr.C.THARINI

[1]Assistant Professor (Sr.G), Department of Electronics and Communication Engineering,

BSA Crescent Institute of Science & Technology, INDIA

[2]Professor, Department of Electronics and Communication Engineering,

BSA Crescent Institute of Science & Technology, INDIA

E-mail: [1]skalaivani@bsauniv.ac.in, [2]hodece@bsauniv.ac.in

ABSTRACT

Wireless sensor networks (WSN) are network that are constructed using number of sensor nodes distributed and connected wirelessly to perform some specific applications. These networks are strictly restricted in the usage of energy as they use batteries with finite amount of power. This necessitates the need for data compression at each sensor node in a WSN so as to overcome the resource constraints of the network and increase its lifetime. Various functions carried out by each node are sensing, processing, communicating and storing the data, among which the communication consumes much energy than the other functions. Data compression is one technique that extends the network lifetime by reducing the energy consumed at each node during communication. The proposed work comprises of 5 different methods that suggests modification of lossless Rice Golomb Coding (RGC) compression algorithm with respect to the decision of tunable parameter based on preprocessing of the input data. Simulation results on various modifications of RGC using different datasets in MATLAB software is analyzed and compared with respect to root mean square error (RMSE) and saving percentage (SP). The modified lossless compression algorithm EMARGC_D with better saving percentage is applied and executed in real time using NI WSN nodes interfaced with LabVIEW.

Keywords: *Data Compression, LabVIEW, MATLAB, NI 3202 Sensing Node, NI 9792 Gateway Node, Rice Golomb Coding, Wireless Sensor Networks*

1. INTRODUCTION

Wireless Sensor network is comprised of number of sensing nodes called as source nodes and a sink node that are connected wirelessly to form the network. It is used for continuous monitoring and large scale data gathering in many fields. These networks have been drawing more attention in various application domains like environmental monitoring, structural monitoring, medicine, agriculture, industrial monitoring and surveillance. The sensor nodes gather the sensed data, process it and transmit it to sink node directly or via the neighboring nodes between them [1]. Among various challenges faced by the researchers in WSN, the power consumption problem is considered to be the foremost one as it directly affect the life time of the network. Also from the survey it is known that the power consumed during transmission of data from sensing node to sink node is more [2].

As the sensor nodes in a WSN are battery driven, they drain out off power due to various activities performed by them such as sensing, computation, communication and storage. Among these activities much of the power is consumed by the node when it transmits the accumulated data to the gateway or sink node. This results in the failure of the sensor nodes and in turn affects the life time of the WSN. Compression techniques, data aggregation, sleep scheduling algorithms and mobile data collection network are the different solutions proposed to reduce the power problem during communication in WSN [3-6]. Data aggregation aims at accumulating, summarizing and transmitting the data from sensors in an energy efficient manner [7], [8]. Compression is the best solution to attack the energy problem in a WSN with the sensor nodes that periodically report their dataset to the sink node [1].

One of the challenging issues in WSN is to design a simple lossless data compression algorithm. Data compression is categorized into three categories types based on the recoverability: lossless, lossy and unrecoverable data compression [9]. In a lossless compression the decompression results with the original data without any loss, in lossy compression the decompression results with some losses in the original data and unrecoverable compression is where no decompression is performed. Selecting the type of compression depends on the application and always an efficient compression algorithm is to be used to reduce the amount of the data being transmitted. Rice Golomb coding is a simple lossless compression algorithm considered for the proposed work. The proposed work aims to analyze and conclude with one modified RGC to be used for any applications in WSN.

2. RELATED WORK

Data compression is categorized into distributed data compression and local data compression [10]. Distributed data compression is used for data with high spatial correlation in dense networks. Algorithms proposed under this category of compression are: distributed source coding (DSC) that involves gathering the correlated data by the cluster head from the subset of nodes and sharing the summary to the sink node [11, 12]; distributed transform coding (DTC) supports lossy compression that transform raw data into a set of coefficient of appropriate basis functions used to reconstruct the original data at the receiver [13, 14];distributed source modeling (DSM) suitable for distributed WSN supports application related to big data analytics [15, 16]; and compressed sensing (CS) [17,18] is based on sampling theory to reconstruct compressed signals from small number of measurements without relying on any prior knowledge about the signal. But these methods results with conservation of energy due to loss of information, also involves complex processing that require lot of memory.

The sensor data fluctuation with respect to time is taken into account in local data compression. Both lossy and lossless compression algorithms come under this scheme. Examples of lossy compression algorithms are: lightweight temporal compression (LTC) [19] algorithm developed for habitat monitoring applications makes use of data windows and generates a series of lines segments which represent the data perfectly; K-RLE [20] a variation of the run length encoding (RLE) method for data compression in WSN; and ADPCM a lossy compression technique [21] used to improve the usage of energy in cluster based WSN.

Examples of lossless compression algorithms are: Improved Lempel Ziv Welch (LZW) [22] a dictionary based algorithm that generates different dictionaries for different compressed contents; Lossless Entropy Compression (LEC) [23] computes the differences of consecutive sensor measurements and divides them into groups that are entropy coded using a fixed compression table; Modified Adaptive Huffman compression scheme [24] has a tree structure with leaves that represent sets of symbols with the same frequency where the data is coded traversing the tree from the root to the leaf with acquired data; Median-Predictor-based Data compression (MPDC) [25] uses the Static Huffman algorithm on the preprocessed data; and Two-modal transmission (TMT) [26] is based on predictive coding approach. Most of these algorithms [27] involve complex computation and are dictionary based that require more memory thereby reduces compression efficiency.

In the proposed work Rice Golomb coding (RGC), a lossless data compression method is considered. It is a well-known compression algorithm for sensor data [28]. WSN an energy constrained network requires simple efficient algorithm in terms of energy consumption, computational complexity and storage. From the literature survey it is observed that the existing compression algorithms have high computational complexity and requires huge storage space for maintaining the adaptive dictionary. The proposed modifications based on RGC leads to a simple, adaptive and lossless compression with no memory storage required for adaptive dictionary structure. The existing work on RGC focuses on the calculation of ‘k’ value that decides the length of the encoded data [29]. The proposed work concentrate on modifying RGC as an adaptive, lossless and efficient compression algorithm that suits a WSN application which reports the data collected to the sink node periodically. Different modifications have been adopted with the RGC algorithm and analysis is performed on the same to conclude with the lossless compression algorithm with better saving percentage. The proposed

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

algorithm is also compared with the existing RGC methods.

Two data sets namely relative humidity and temperature each of 744 data collected for 31 days in hourly basis taken from the database [34] are considered on which different modified RGC are applied using MATLAB software for the analysis. As RGC is a lossless compression, the different modified RGC algorithms are also lossless for which the root mean square error (RMSE) is to be zero. The analysis and comparison of the algorithms is based on saving percentage (SP) and root mean square error (RMSE). The proposed work results with a simple and lossless compression algorithm that suits a WSN application which reports the data collected to the sink node periodically.

3. CONVENTIONAL RICE GOLOMB CODING

Rice Golomb code is much simpler and faster than Huffman code because it can be computed with a few logical operations [30]. In Golomb coding when a non negative input value 'A' is divided by a tunable parameter 'B' it results into two parts: quotient 'Q', and the remainder 'R' i.e (A/B) =Q coded as a unary value and (A mod B) = R coded as a binary value.

3.1 Encoding Algorithm

B is considered as an integer power of 2 i.e. B = 2^K. Encoding the input value 'A' is as follows:

1. R = A & (B - 1) in binary.
2. Q = A >> K in unary.

3.2 Decoding Algorithm

The tunable parameter K is obtained from the encoded bits and then B is computed using the equation B = 2^K. The following steps are carried out to retrieve the input value (A):

1. Count the number of 1s before the first 0 to determine Q.

2. The next K bits are read as a binary value to determine R.
3. Now A= Q × B + R.

As an example when a number 18 represented in 8 bit as 00010010 is compressed using RGC then it is represented as 110010 in 6 bits. But when RGC is to be applied for a sequence of data to be transmitted from a sensor node to a sink node, along with the compressed bits of data sequence the 'k' value of each data has to be sent to the sink node so as to retrieve the original data sequence. When the conventional RGC is simulated using Matlab software applied for relative humidity dataset, the length of the encoded data is 7869 bits out of which the compressed data bits is 5637 bits and extra 2232 bits are used to represent the 'k' value of each data during compression (i.e. 3 bits for each 'k' value). This result is shown in figure 1.

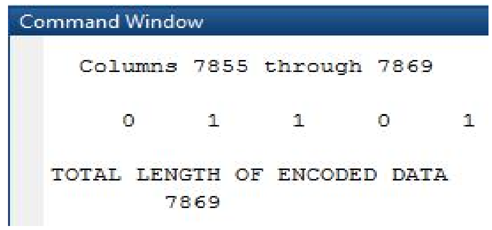


Figure 1: Encoded Output Of RGC

If the data is sent without applying RGC, it requires only 5952 bits (i.e. 744* 8bits). The drawback of this method is that the number of bits for encoding increases due to the tunable parameter.

4. PROPOSED MODIFICATIONS IN RGC ALGORITHM

In the proposed work four algorithms have been developed by modifying the RGC which are then analyzed and compared to conclude with one algorithm that would provide a better SP. The saving percentage (SP) and root mean square error (RMSE) is calculated using equations (1) and (2),

$$SP = 1 - \frac{\text{Compressed data length}}{\text{Uncompressed data length}} \times 100 \quad (1)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (\text{Inputdata}(i) - \text{Outputdata}(i))^2}{n}} \quad (2)$$

The concluded algorithm is further considered for implementing in the WSN node and the respective results are displayed in labVIEW.

4.1 Modification I

The algorithm developed by considering fixed length compression over conventional RGC is described in the following steps:

i. Given the sequence A = {a1, a2,….a_n}.
ii. Find the maximum in the sequence A_{max}.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12
© 2005 – ongoing JATIT & LLS

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

iii. A_{max} is divided by the adaptive tunable parameter 'B' that results with minimum encoding length which in turn depends on the size of 'Q'(i.e, Q is 0 and R with 1 as MSB bit).

iv. A_{max} is coded as per conventional RGC.

v. Now bitwise-XOR of the encoded bits is performed for further reduction in the size of encoded bits.

vi. Find minimum length of encoded A_{max} 'L_{min}'.

vii. Now encode the sequence A with adaptive K value using RGC, also prefix zeros to get fixed length code that matches L_{min} for all data in the sequence.

viii. The first three bits in the encoded format is the fixed length followed by the encoded bits of the sequence.

ix. While decoding, reverse of bitwise-XOR operation is performed by removing the appended zeros in the encoded data.

x. Since 'K' is made adaptive to result with 'Q' as 0 for all data in the sequence, now 'R' is determined from the remaining K bits of binary value from which 'B' is obtained.

xi. The input sequence is now determined from The expression A = Q × B+ R.

The pidgin code of the algorithm is shown in the below figure 2.

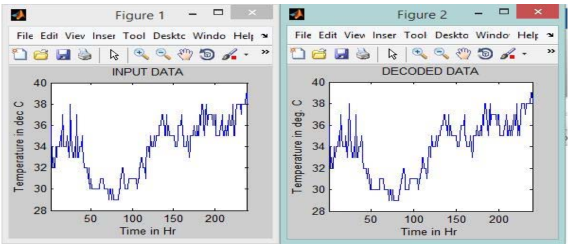
\\ ENCODING

1. A[n] = Input sequence;
2. A_{max} = maximum (A);
3. K= 0 to n;
4. B=2^K; Q= Unary(A_{max}/B);
5. R= Binary(A_{max} mod B);
6. E(K) = bitwise-XOR(concatenate{R,Q});
7. L_{min} = min Length (E(K));
8. Len = binary3 (Lmin);
9. Q = Unary(A/B);R = Binary(A mod B);
10. E = bitwise-XOR(concatenate$\{R,Q\}_{len}$);
11. En( )=concatenate{0,E}until len(En) =L_{min};
12. Enc = concatenate{Len, En}; \\ Encoded data

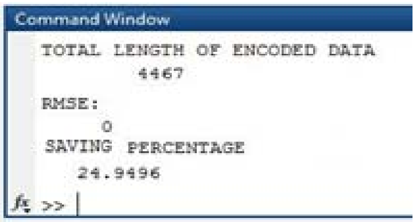
\\DECODING

13. L_{min} = decimal(Enc(1:3));
14. D( )=decatenate {Enc (4:L)}until 1 in MSB of each encoded data of length L_{min}
15. Dec = bitwise XNOR (D)
16. Q = no. of 1's until first 0 in Dec;
17. K=length (Dec – (Q+1)); B= 2^K;
17. R= binary to decimal (Dec);
18. AD= B*Q+R; \\ Decoded data
19.end

Figure 2: Pidgin Code Of The Modification I Algorithm



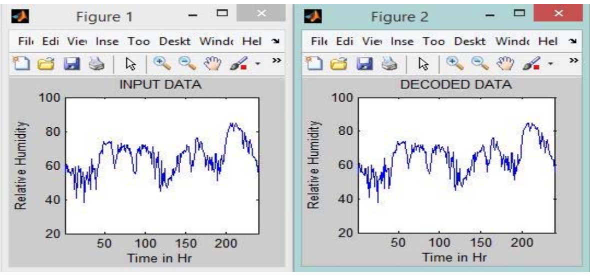
(a) (b)



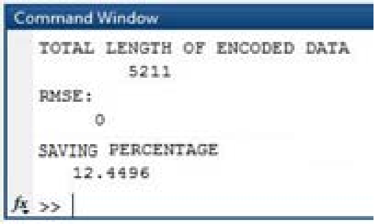
(c)

Figure 3: Result On Applying Modification I Of The RGC Algorithm On Temperature Dataset

From the above figure 3 it is found that while applying the modification I of the RGC algorithm on the temperature dataset it results with the SP of 24.9496% and RMSE = 0.



(a) (b)



(c)

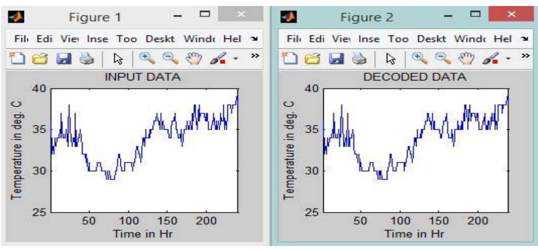
Figure 4: Result On Applying Modification I Of The RGC Algorithm On Relative Humidity Dataset

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

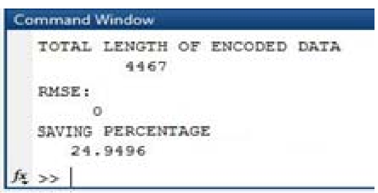
Figure 4 shown above results with the SP of 12.4496% and RMSE = 0 when the modification I of the RGC algorithm is applied on the relative humidity dataset.

4.2 Modification II

Modification II adopted in RGC is similar to modification I except, considering the maximum value of the sequence to determine the fixed length of the encoded sequence, the average value is considered. All the other procedure remains the same.



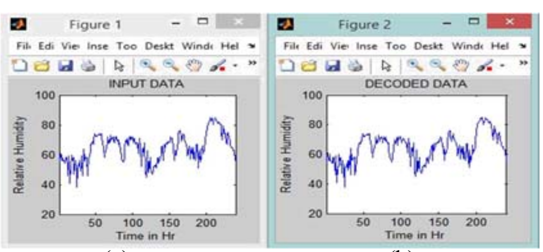
(a) (b)



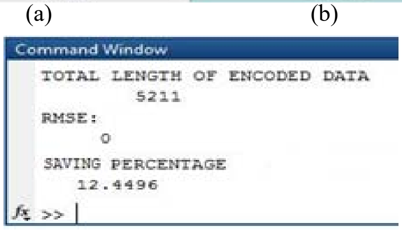
(c)

Figure 5: Result On Applying Modification II Of The RGC Algorithm On Temperature Dataset

The above figure 5 shows the result of applying the modification II of the RGC algorithm on temperature and relative humidity dataset. It results with the SP of 24.9496% and RMSE = 0.



(a) (b)



(c)

Figure 6: Result On Applying Modification II Of The RGC Algorithm On Relative Humidity Dataset

From figure 6 it is found that while applying the Modification II of the RGC algorithm on relative humidity dataset the SP obtained 12.4496% and RMSE = 0.

Both the modifications discussed in this section results with same SP for the two different dataset considered, but may have slight variations in SP with respect to the dataset considered. These two algorithms will result with same SP for the data collected continuously/periodically over the same environment by a WSN.

4.3 Modification III – MARGC Algorithm

The next modification made with RGC is referred as MARGC (Modified Adaptive Rice Golomb Coding) algorithm proposed in [31] is used as a lossy or lossless algorithm that depends on the application considered. For the analysis of different modifications adopted with RGC, lossless MARGC algorithm is considered.

Encoding procedure of MARGC algorithm [31]:

(i) Find the average value of the input sequence.

(ii) The difference sequence is obtained by differencing the average value with the input sequence.

(iii) Find the maximum negative number and add it to the sequence to generate a positive sequence.

(iv) The fixed length for the encoded data is obtained coding the maximum value of the positive sequence with a suitable adaptive parameter as per RGC.

(v) The input data of the sequence 'A' is divided by the adaptive tunable parameter 'B' that results with small encoding length which in turn depends on the size of 'Q' for each data value.

(vi) Each data is then coded as per RGC.

(vii) Bitwise-XOR operation is performed on each encoded data.

(viii) Now the fixed length code equivalent to fixed length is obtained by appending zeros.

(ix) Format of encoded data: first 8 bits - average value, next 3 bits - fixed length and then the encoded data.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Decoding procedure of MARGC algorithm:

(i) The average value and the fixed length are determined from first 8 bits and the next 3 bit of the encoded sequence.

(ii) The first encoded data of the encoded sequence represents the maximum negative value used for to obtain positive sequence from the difference sequence while encoding.

(iii) Reverse bitwise-XNOR operation is performed after removing the appended zeros from the encoded sequence leaving a '1' in the MSB of each data.

(iv) Since 'K' is made adaptive to result with 'Q' as 0 for all the data in the sequence, now 'R' is determined from the remaining K bits of binary value from which 'B' is obtained (i.e. B = 2^K).

(v) The difference sequence is obtained by differencing the maximum negative value with the sequence obtained by the formula Q*B+R.

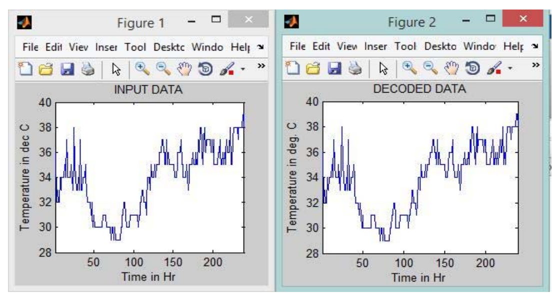
(vi) The original input sequence 'A' is then obtained by adding the average value with the difference sequence.

The pidgin code of MARGC algorithm is shown in figure. 7 below [31]:

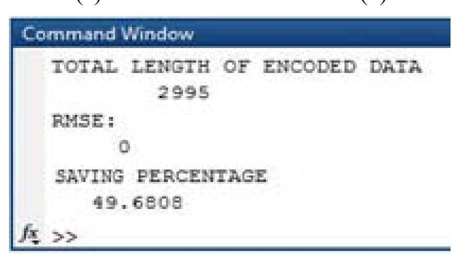
```
\\ENCODING

1.  A(N) = Input Sequence
2.  Av = average(A(N));
3.  A(N) = A(N) – Av;
4.  MN = Maximum (negative value(A));
5.  D = A+ MN; DS(1:N+1) = D (0:N);
6.  DS(0) = MN; N= N+1;DSM=Max(DS);
7.    K = 0 to 8;B=2^K;
8.    Q= Unary(DSM/B);R= Binary(DSM mod B);
9.    if (Q==0 && R(K) == 1)
10.   DSE(K) = bitwise-XOR(concatenate{R,Q});
11.   Lmin = minLength (DSE(K));
12.   Len= binary3 (Lmin);
13.   K = 0 to 8; B=2^K;
14.   Q= Unary(DS/B);
15.   R= Binary(DS mod B);
16.   E=bitwise- OR(concatenate{R,Q}min len);
17.   En()=concatenate{0,E}until len (En==Lmin);
18.   Enc = concatenate{Len,En}; \\ Encoded data

\\DECODING

19.  Av = decimal (Enc(0:7))
20.  Lmin = decimal(Enc(8:10));
21.  DES( ) = decatenate {Enc (4:L),0} until 1 in
     MSB of each encoded data of length Lmin
22.  DEC = bitwise XNOR (DES( ));
23.  Q = no. of 1's until first 0 in Dec;
24.  K==length (DEC – (Q+1));
25.   B= 2^K;
26.  R = binary to decimal (DEC);
27.  PD= B*Q+R; D = PD (0);
28.  A= DS + Av; \\Decoded data
29. DS( ) = PD(1 : length(PD)-1) – Mn;
28. end.
```

Figure 7: Pidgin Code Of MARGC Algorithm

Figure 8 shown below represents the input and the corresponding decoded temperature data on applying MARGC algorithm that results with the SP of 49.6808% and RMSE=0.



(a) (b)

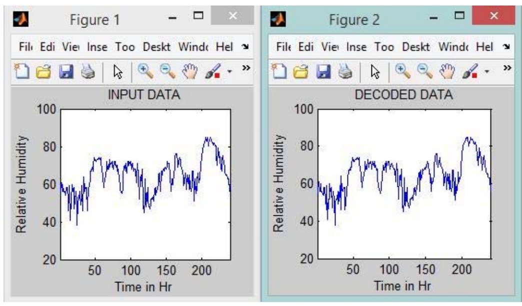


(c)

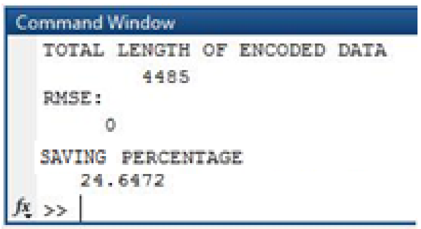
Figure 8: Results Of Applying MARGC Algorithm On Temperature Dataset

The result of applying MARGC algorithm on relative humidity dataset with the SP= 24.6472 % and RMSE = 0 is shown in figure 9.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



(a) (b)



(c)

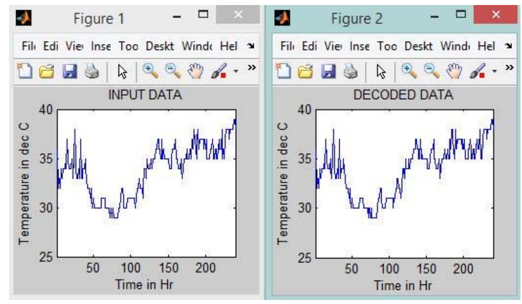
Figure 9: Results Of Applying MARGC Algorithm On Relative Humidity Dataset

In both the figures 8 & 9 (a) shows the input data, (b) represents the decoded data and (c) displays the total length of decoded data, SP and RMSE. From this result it is found that MARGC algorithm provides better SP than the previous discussed algorithm.

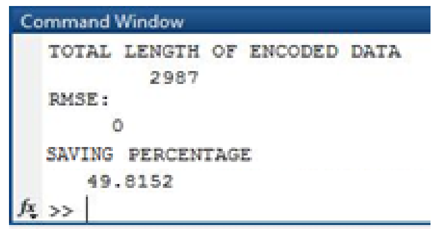
4.4 Modification IV – MARGC_D Algorithm

Modification IV is based on MARGC algorithm where instead of taking the average to get the differenced sequence in the first step of the algorithm; auto-differenced sequence is considered as the input for the algorithm. The remaining procedure is the same where in the encoded format, the first 8 bits will represent the first data in the input sequence instead the average value as seen in MARGC algorithm. This modified algorithm is referred as MARGC_D (Modified Adaptive Rice Golomb Coding with Auto Differencing) algorithm.

In both the figures 10 & 11, (a) shows the input data, (b) represents the decoded data and (c) displays the total length of encoded data, SP and RMSE. From the below figure 10 it is found that while applying the MARGC_D algorithm on the temperature dataset it results with the SP of 49.8152% and the RMSE = 0.



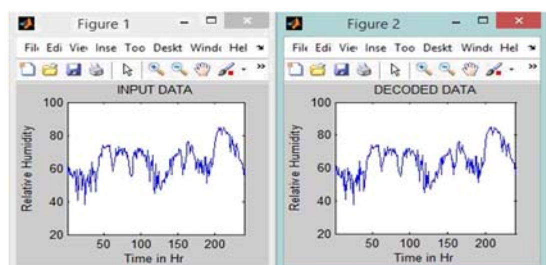
(a) (b)



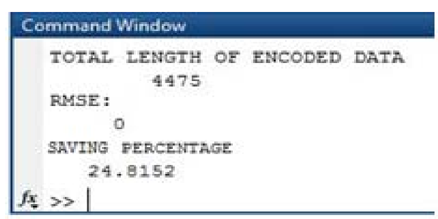
(c)

Figure 10: Results Of Applying MARGC_D Algorithm On Temperature Dataset

The figure 11 shown below displays the results of applying MARGC_D algorithm on relative humidity dataset with SP = 24.8152% and RMSE = 0.



(a) (b)



(c)

Figure 11: Results Of Applying MARGC_D Algorithm On Relative Humidity Dataset

By applying MARGC_D algorithm on temperature and relative humidity dataset it is found that it provides better SP than MARGC algorithm.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

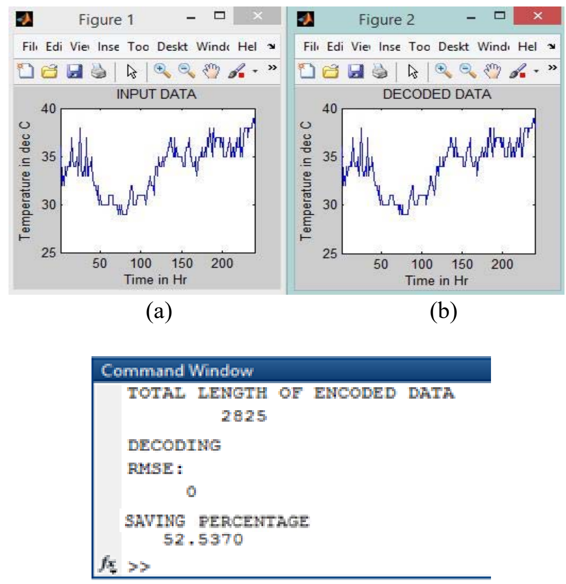
4.5 Modification V – EMARGC_D Algorithm

In modification V the encoding of the data is performed by selecting the tunable parameter based on the probability of occurrence of data. The tunable parameter for mostly occurring data in the sequence is selected and is fixed for encoding all other data in the sequence that leads to variable length coding instead of fixed length coding as in previous algorithm. The pidgin code of the algorithm is given below in figure 12. The algorithm is referred as EMARGC_D (Efficient Modified Adaptive Rice Golomb Coding with Auto Differencing).

```
\\ ENCODING
1.  A(N )= {a1,a2,…an}  \\Input Sequence
2.  F=a1; Ad = auto-difference( A);
3.  MN = Maximum (negative value(A));
4.  DS (1:N+1) = Ad + MN;
5.  DS(0) = MN; DSM= mode(DS);
6.  K = 0 to 8; B=2^K;
7.  Q= Unary(DSM/B);
8.  R= Binary(DSM mod B);
9.  if (Q==0 && R(K) == 1)
10. DSE = concatenate{R,Q});
11. LD= Length (DSE);
12. Lf = Min(LD); B=2^Kf;
13. Q= Unary(A/B); R= Binary(A mod B);
14. En=concatenate{Q,R};
15. EnSq=concatenate{binary8(F),bin3(Kf),
    En}; \\Encoded data
16. end;
\\ DECODING
27. Len=length(EnSq);
28. F= decimal (EnSq(0:7));
29. Kf = decimal(EnSq(8:10));
30. B=2^Kf;Q=0; j=0;
31. do{
32. for (i=11; i<=Len; i++)
33. { if (Ensq(i) !=0 )   Q=Q+1;}++i;
34. R=Decimal(EnSq(i:i+K));
35. A(j)= B*Q+R; // Decoded Data
36. ++j; While(i != Len);
37. End.
```

Figure 12: Pidgin code of EMARGC_D algorithm

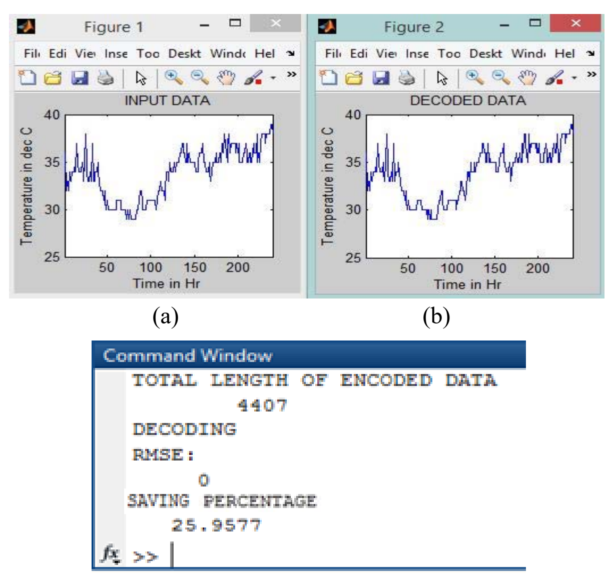
The figure 13 represents the input and the corresponding decoded temperature data on applying EMARGC_D algorithm that results with the SP of 52.537 % and RMSE=0.



(c)

Figure 13. Results Of Applying EMARGC_D Algorithm On Temperature Dataset

The result of applying EMARGC_D algorithm on relative humidity dataset with the SP = 25.96 % and RMSE = 0 is shown in figure 14.



(c)

Figure 14: Results Of Applying EMARGC_D Algorithm On Relative Humidity Dataset

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

In both the figures 13 & 14, (a) shows the input data, (b) represents the decoded data and (c) displays the total length of encoded data, CR and RMSE. From the result obtained it is found that EMARGC_D algorithm provides better SP than the previous discussed algorithms.

5. RESULT ANALYSIS

Two different datasets namely temperature (Temp) and relative humidity (RH) from climate weather database [34] is considered for applying the various modified RGC algorithm and analyze their performance using MATLAB software. The proposed algorithm is also compared with the existing RGC methods discussed below.

As discussed in section III an optimum 'k' value is required in order to transfer the sequence of data from the sensing node to sink node with better saving percentage. Number of works is still being carried out in the selection of Golomb parameter that is based on the following two methods [32, 33].

5.1 Optimum 'k' using maximum value of the sequence – Existing Method I

In this method the 'k' value is obtained for the maximum value of the given sequence and that value is kept constant throughout the sequence of data. Instead of having individual 'k' values for each data in the sequence a single 'k' of the maximum value for the given sequence is considered for all the data which is represented as first 3 bits in the encoded data to be sent from the sensing node to sink node.

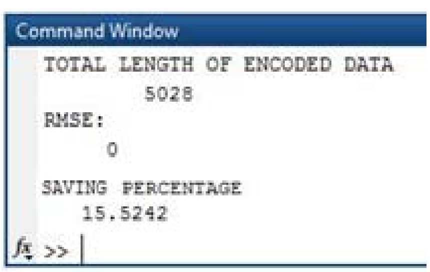


Figure 15: Result Of Applying RGC With Optimum 'K' Value Using Maximum Value Of The Temperature Dataset.

The above Figure 15 shows the result of applying RGC with optimum 'k' value using maximum value of the temperature dataset with the total length of the encoded data as 5028 bits, RMSE = 0 & SP = 15.5242%. The Figure 16 shows the result of applying RGC with optimum 'k' value using maximum value of the relative humidity dataset with the total length of the encoded data as 5640 bits, RMSE = 0 & SP = 5.2419%.

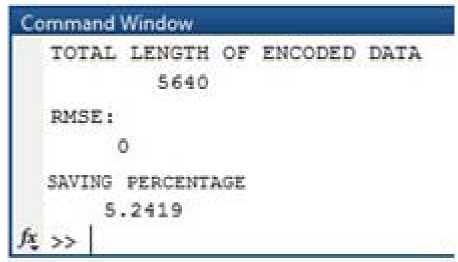


Figure 16: Result Of Applying RGC With Optimum 'K' Value Using Maximum Value Of The Relative Humidity Dataset.

From the above result it is seen that the algorithm with optimum 'k' value using maximum value of the input data provides better SP compared to the conventional RGC.

5.2 Optimum 'k' using average value of the sequence – Existing Method II

In this method the 'k' value is obtained for the average value of the given sequence and that value is kept constant throughout the sequence of data. Instead of having individual 'k' values for each data in the sequence a single 'k' of the average value for the given sequence is considered for all the data which is represented as first 3 bits in the encoded data to be sent from the sensing node to sink node.

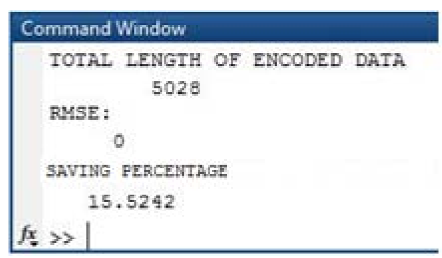


Figure 17: Result Of Applying RGC With Optimum 'K' Value Using Average Value Of The Temperature Dataset.

The figure 17 shown above is the result of applying RGC with optimum 'k' value using average value of the input temperature data with the total length of the encoded data as 5028 bits, RMSE = 0 and SP = 15.5242%. The figure 18 below shows the result of applying RGC with optimum 'k' value using average value of the relative humidity data with the total length of the encoded data as 5640 bits , RMSE = 0 & SP = 5.2419%. This has given a better SP compared to the above seen conventional RGC.

Journal of Theoretical and Applied Information Technology
30th June 2018. Vol.96. No 12
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

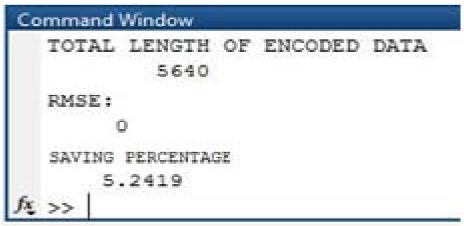


Figure 18. Result Of Applying RGC With Optimum 'K' Value Using Average Value Of The Relative Humidity Dataset.

Both the methods discussed in this section results with same CR for the two different dataset considered, but SP is better than the conventional RGC. These two methods may have slight variations in SP with respect to the dataset considered but will result with same SP for the data collected continuously/periodically over the same environment by a WSN.

Parameters considered to analyze the efficiency of different modified RGC algorithms are:

1. Length of encoded data (EL)
2. Saving Percentage (SP) and
3. RMSE (Root Mean Square Error).

The results are tabulated in Table I shown below. The actual encoded length before applying compression is 5952 bits i.e. (8bits*744 data).

TABLE 1: Length of Encoded Data (EL), Saving Percentage (SP) and RMSE by Applying Modified RGC Algorithms on Temperature and Relative Humidity Dataset.

	Modified RGC Algorithms									
	Existing method I & II		Modification I & II		Modification III MARGC		Modification IV MARGC_D		Modification V EMARGC_D	
	Temp	RH	Temp	RH	Temp	RH	Temp	RH	Temp	RH
EL (bits)	5028	5604	4467	5211	2995	4485	2987	4475	2825	4407
SP (%)	15.52	5.24	24.95	12.45	49.68	24.65	49.82	24.82	52.53	25.96
RMSE	0	0	0	0	0	0	0	0	0	0

From Table 1, it is observed that EMARGC_D algorithm results with minimum number of encoding bits i.e. 2825 bits for temperature dataset and 4407 bits for relative humidity dataset. Also it gives better SP of about 52.53% for temperature dataset and 25.96% for relative humidity dataset. As the proposed algorithm is a lossless data compression algorithm it results with RMSE = 0 for the considered datasets. Therefore from the results obtained it is concluded that the proposed EMARGC_D algorithm performs better than the other modified RGC algorithms. This simple computational, lossless compression algorithm is implemented in real time using NI WSN Hardware.

6. EXPERIMENTAL SETUP

The real time implementation of the EMARGC_D algorithm is carried out using NI WSN hardware. The experimental setup of the implementation is shown in figure 19. The hardware set up consists of two nodes namely the NI 3202 measurement node to sense the temperature data and a gateway NI 9792 considered as the sink node.

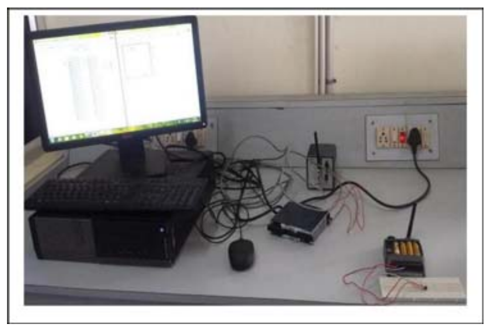


Figure19: NI WSN Hardware Module

The sensing node is placed at NI Laboratory and the gateway node is placed at the Network laboratory as shown in figure 20.

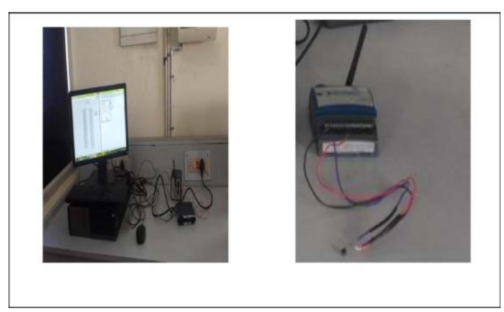


Figure 20: Experimental Setup With NI WSN Modules
(a) Gateway Node In Networks Lab
(b) Measurement Node In NI Lab

© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LabVIEW software [35, 36] is a graphical programming language used to develop the algorithm and implement it in the NI sensing node and the gateway node.

EMARGC_D algorithm is developed using LabVIEW software to program the algorithm in the respective nodes. The measurement node is interfaced with the temperature sensor LM35 to acquire the temperature data. The acquired data is then subjected to compression algorithm developed using LabVIEW. The encoded data is then transmitted at 2.4 GHz to the gateway node where the decoding algorithm is implemented. The decoded data, saving percentage and the RMSE value is displayed in the front panel of the LabVIEW as shown in figure 21.

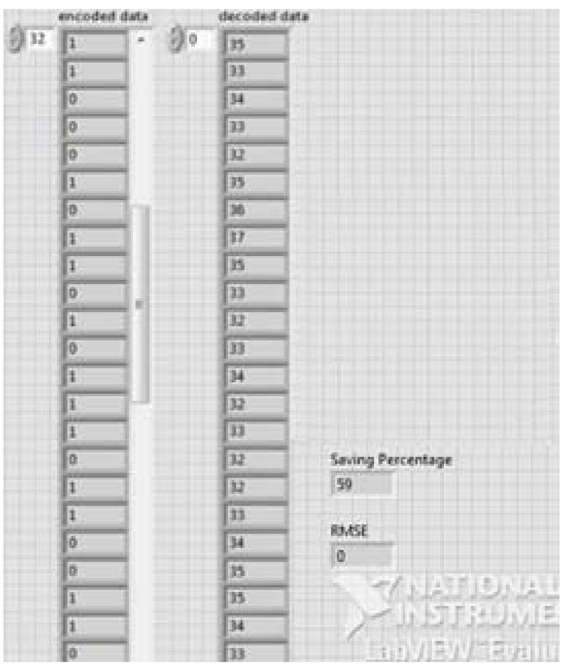


Figure. 21: Labview result in the front panel

The real time implementation of EMARGC_D algorithm results with the saving percentage of 59% that is better than the simulation results as the real time data are highly auto-correlated and the RMSE is 0 since the algorithm implemented is a lossless algorithm.

7. CONCLUSION

In this paper, the simple RGC lossless compression algorithm is considered. Different modifications were applied in the conventional algorithm to analyze and conclude with one efficient algorithm to be used in wireless sensor network. The proposed simple and lossless algorithm EMARGC_D that was based on auto-differencing of input data subjected to the encoding & decoding procedure resulted with better saving percentage. Thereby the proposed algorithm will help in increasing the network lifetime by reducing the amount of energy consumed by the sensor nodes during transmission. The proposed algorithm is best suited for resource-constrained wireless sensor nodes in a wireless sensor network that periodically transmit a set of data to the sink node. Additionally, the proposed algorithm may also be used for different dataset acquired in various applications of WSN that would provide a satisfactory saving percentage. The algorithm is implemented using NI WSN hardware. The limitation of the proposed algorithm is that it works on with integer data values. To work with decimal values, data preprocessing is required. Also the performance of the algorithm increases with highly auto-correlated data and degrades for uncorrelated data. As an extension of the work, the algorithm will be applied on biomedical signals like electrocardiogram (ECG) used in telemedicine applications to test its performance on sensitive signals and also in IoT applications where huge amount of data is transmitted and stored in the cloud.

REFRENCES:

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless sensor networks - A survey,” *Computer Networks, Elsevier*, Vol. 38, Issue 4, pp. 393-422, March 2002.

[2] Prabhudutta Mohanty, Manas Ranajan Kabat and Manoj Kumar Patel, “Energy efficient transmission control protocol in Wireless sensor networks,” *Wireless Networks and Computational Intelligence*, pp. 56-65, Springer-Verlag Berlin Heidelberg 2012.

[3] Rick W.Ha et al., “Sleep Scheduling for WSN via network flow model”, *Computer communication*, Vol. 29, Issue 13-14, pages. 2469-2481, August 2006.

[4] Guanxiong Shi et al., “Comprehensive Review of Sleep/Awake Scheduling in WSN”, *High Performance Networking, computing and communication systems, CCIS*, Vol.163, pp. 492-499.

[5] Arun K. Kumar and Krishna M. Sivalingam, “Energy efficient mobile data collection in WSN with delay reduction using wireless communication”, *International conference on*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

communication systems and networks, IEEE explore, March 2010.

[6] Kai Li and Kien A. Hua, "Mobile data collection networks for wireless sensors", *Communications in Computer and Information Science, Springer*, vol.287, pp.200-211, 2012.

[7] Geetika Dhand and S.S.Tyagi, "Data Aggregation Techniques in WSN: Survey," *Procedia in Computer Science, Elsevier,* vol. 92, no. 1, pp. 378-384, 2016.

[8] M. Hussaini et al., "Enhanced Clustering Routing Protocol for Power Efficient Gathering in Wireless Sensor Network", *International Journal of Communication Networks and Information Security* (IJCNIS), Vol.4, No. 1, 2012.

[9] Yimei and Yaoliang, "Temporal Lossless and Lossy Compression in Wireless Sensor Network," *ACM Transactions on Sensor Networks (TOSN)*, Volume 12 Issue 4, November 2016.

[10] Jonathan Gana Kolo, S.Anandan Shanmugam, David Wee Gin Lim, Li-Minn Ang and Kah Poohi Seng, "An Adaptive Lossless Data Compression Scheme for Wireless Sensor Networks", *Journal of Sensors, Hindawi Publishing*, Volume 2012, Article ID 539638, 2012.

[11] Guogang Hua and Chang Wen Chen, "Distributed Source Coding in wireless sensor networks," *Quality of Service in Heterogeneous Wired/Wireless Networks, IEEE Explore*, 2005.

[12] Yeun, K., Liang, B. and Li, B. ,"A Distributed framework for correlated data gathering in sensor networks", *IEEE Trans. Vehicular Technology*, 57, 1, pp. 578-593,2008

[13] Wagner,R.S., Baraniuk,R.G., Johnson,D.B. and Cohen A, "An architecture for distributed wavelet analysis and processing in sensor networks", *In proceedings of the 5th International Conference on Information Processing in Sensor networks*, ACM, 243-250,2006.

[14] Jian Chen and Xue Han, "The distributed source coding method research based on clustering wireless sensor networks", *International Journal of Sensor Networks*, Volume 17 Issue 4, Pages 224-228, May 2015.

[15] You-Chiun Wang, Yao-Yu Hsieh, and Yu-Chee Tseng,"Compression and Storage Schemes in a Sensor Network with Spatial and Temporal Coding Techniques", *Vehicular Technology Conference*, 2008. VTC Spring 2008. IEEE, 11-14 May 2008.

[16] Mohamed Moustafa Fouad et al, "Big Data Pre-processing Techniques Within the WSN", *Proceedings of the Second International Afro-European Conference for Industrial Advancement*, Springer International Publishing, 2015.

[17] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sensing," *IEEE Transactions on Signal Processing*, vol. 56, no. 6, pp. 2346–2356, 2008.

[18] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp.1289–1306, 2006.

[19] Renu Sharma, "A data compression application for wireless sensor networks based on LTC (Lightweight Temporal Compression)", *Electro/Information Technology (EIT), IEEE Explore*, 08 October 2015.

[20] E. P. Capo-Chichi, H. Guyennet, and J. M. Friedt, "K-RLE: a new data compression algorithm for wireless sensor network," in *Proceedings of the 3rd International Conference on Sensor Technologies and Applications*, (SENSORCOMM"09), pp.502–507, June 2009.

[21] Kasirajan P., Larsen C. and Jagannathan S, "A new data aggregation scheme via adaptive compression for wireless sensor networks", *ACM Trans. Sensor Network* 2012.

[22] Zhou Yan-li et al,"Improved LZW Algorithm of Lossless Data Compression for WSN", *3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), 2010, IEEE* explore, page 523 – 527, DOI. 10.1109/ICCSIT.2010.5563620.

[23] F. Marcelloni and M. Vecchio, "An efficient lossless compression algorithm for tiny nodes of monitoring wireless sensor networks," *Computer Journal*, vol.52, no.8, pp.969 – 987, 2009.

[24] C.Tharini and P.VanajaRanjan, "Design of modified adaptive Huffman data compression algorithm for wireless sensor network," *Journal of Computer Science*, vol. 5, no. 6, pp. 466– 470, 2009.

[25] A. K. Maurya, D. Singh, and A. K. Sarje, "Median predictor based data compression algorithm for Wireless Sensor Network," *International Journal of Smart Sensors and Ad Hoc Networks*, vol. 1, no. 1, pp. 62–65, 2011.

[26] Y. Liang and W. Peng, "Minimizing energy consumptions in Wireless Sensor Networks via two-modal transmission," *Computer*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Communication Review, vol. 40, no. 1, pp. 13–18, 2010.

[27] G.Campobello et al., "Comparison of local lossless compression algorithms for wireless sensor networks", *Journal of Network and Computer Applications*, Volume 47, Pages 23-31, January 2015.

[28] Tetsuya Takezawa, Koichi Asakura, Toyohide Watanabe, "Lossless compression of time-series data based on increasing average of neighboring signals", *Electronics and Communications in Japan*, Vol. 93, No. 8, 2010.

[29] Weinberger, M J., Seroussi G. and Sapiro G., " LOCO-I: A Low Complexity, Context-Based, Lossless Image Compression Algorithm." *Proceedings of the IEEE Data Compression Conference,* Snowbird, p.no.140-149, March, 1996.

[30] Zoran H. Peric, Milan R. Dincic and Marko D. Petkovic, "Design of a hybrid quantizer with variable length code", *Fundamental Informatics*, vol. 98, no. 2-3, pp. 233-256, 2010.

[31] S.Kalaivani, C.Tharini, "Efficient Data Compression Technique Using Modified Adaptive Rice Golomb Coding For Wireless Sensor Network", *ARPN Journal of Engineering and Applied Sciences*, Vol. 10, No. 12, July 2015.

[32] A. Kiely, "Selecting the Golomb Parameter in Rice Coding", *IPN Progress Report* 42-159, November, 2004.

[33] Moab Mariz Meira et al,"An FPGA implementation of a lossless ECG compressor based on prediction and Golomb Rice coding for telemedicine applications", *XXI Symposium Brasileiro DeTelecomunication-SBT*'04, 06-09 September 2004.

[34] http://www.climate.weather.gc.ca/

[35] http://sine.ni.com/nips/cds/view/p/lang/en/nid/206915

[36] http://www.ni.com/white-paper/11529/en/