International Journal of Engineering Research and Technology.
ISSN 0974-3154 Volume 11, Number 1 (2018), pp. 117-143
© International Research Publication House
http://www.irphouse.com

# An Efficient Tag Generation based Data Compression Algorithm for Wireless Sensor Network

**Usha Tiwari[1], Dr. Shabana Mehfuz[2]**

*[1] Research Scholar, Electrical Engineering Department, Jamia Millia Islamia, New Delhi, India.*

*[2]Professor, Electrical Engineering Department, Jamia Millia Islamia, New Delhi, India.*

*[1]ushapant@rediffmail.com, [2]smehfuz@jmi.ac.in*

## Abstract

In recent years, Wireless Sensor Networks (WSN's) deployments have rapidly increased for real time applications in various areas. In WSN, power consumption is of utmost importance as they are battery operated with limited capacity,and therefore power consumption is one of the key constraint in the design of wireless sensor networks. Communication unit which consist of a Radio transmitter is the main source of power consumption in the node. A variety of techniques have been proposed to address this issue, data compression is one of them, which reduces the data volume to be transmitted, and hence results in power saving. In the past several years few data compression approaches for WSN were proposed, but few options are available which uses the lossless data compression techniques utilizing the Local data compression on a particular node in WSN. Lossless data compression technique regenerates the information as it is transmitted and helpful in those phenomena where loss of information is critical.

In this paper, an efficient tag generation based adaptive data compression scheme is introduced which generates the unique identifier or tag for the sequence of wireless input from real world to be encoded. The proposed technique sends a single coded message for group of symbols as compared to other entropy based techniques which generate the code for each input symbol in a particular stream of input from real deployments. By sending the single coded message for a stream of signal it reduces the data to be transmitted and hence result in a reduction in power consumption. The results obtained are

69.82, 63.54, 68.5 (%) for temperature data set and 59.3, 52.6, 54.2 (%) for humidity data set. Comparison with existing available techniques has been carried out to validate the proposed algorithm. Memory requirement and power consumption have also been compared with the available techniques.

**Keywords:** Wireless Sensor Networks (WSN's), power dissipation, sensor nodes, sink.

## 1. INTRODUCTION:

Wireless Sensor network (WSN's) are large scale deployments of variety of sensors equipped with a microcontroller board called sensor nodes which work together to monitor or track a particular region and collects the various types of data related with the environment. The sensor nodes are scattered randomly in an adhoc manner in a particular geographical terrain. WSN can be used to monitor the environment, track object and monitor the activity of target, control industrial operation and reporting events or information via the communication unit to a base station or a sink node for further processing required or for transmitting to other nodes. There are many application areas including home automation, sales tracking, industrial process control and enemy target tracking in military operations. In general a sensor node mainly consists of three parts: *a sensing unit*, *a processing un*it and *a communication unit*. A sensing unit is used to acquire the physical variable from the target or data of interest. A variety of sensors like Temperature sensor, pressure sensor, and humidity sensor can be present on a node. A GPS (Global positioning System) can also be incorporated on the sensor board to get the position of the sensor location. A processing unit is generally a microprocessor or microcontroller with limited memory and computational power. A communication unit is generally a radio transmitter which collects the information or data from various sensor nodes in a network and communicates it to a collection center called as a sink or base station. Sensor nodes are generally powered by small sized batteries which cannot be charged or changed frequently as sensors are deployed in some remote locations. So a Data compression is one method that can be used to manage the high fidelity data transmission to the base station (sink node) and make use of limited sources of sensor nodes without any loss of information. A byte of information saved via compression process is equivalent to saving the 4,000 computation cycles for Chipcon CC2420 transceiver and 2million computation cycles for the Max Stream XTend transceiver. Reduction in data transmitted will result in considerable saving in power. So compressing the data before transmission is one of the key methods for increasing the lifetime of a sensor network.

Data compression algorithms in WSN are generally classified into two main categories: a distributed data compression approach and a local data compression approach [1]. These approaches can be Lossless or Lossy depending upon the output generated. A lossless approach ensures the integrity of data during the compression/decompression process, while a lossy algorithm may generate a loss in information but guarantees a higher compression ratio. Depending upon the area of application and requirement of the system, a desired data compression algorithm is used. For the applications like environmental monitoring, the accuracy of observation is critical for understanding the principle behind it. In addition to this the application like BAN (Body Area Networks) in which sensors continuously monitor the log vital signs. Each and every small variation in these signals should be recorded because it provides crucial information about the diagnosis of the patient as in Marcelloni [5]. So, lossless data compression techniques are essential and desirable in WSN. Although the Lossy compression algorithm may result in some loss of information, but it enables the encoder to increase its compression rate, which is high as compared to the Lossless algorithm. These algorithms are used for applications which do not require the precise information, for e.g. for storing or transmitting the voice signals the exact value of each sample is not necessary to record. Depending on the required quality of the reconstructed voice, varying amounts of loss of information about the value of each sample can be tolerated. If the quality of the reconstructed speech is to be similar to that heard on the telephone, a significant loss of information can be tolerated. So, according to the area of application a suitable data compression technique is required as per the WSN constraints. As the sensors are having few kilobytes of memory and 4-8MHz microprocessor, so the data compression algorithm for WSN should be lightweight (require less memory) and computational requirement of the algorithm should be low for effective operation of WSN due its various constraints in terms of hardware, energy, processing speed and memory. Several approaches have been proposed in past years to meet these specifications, but only few numbers of lossless data compression techniques are available.

The main challenge in the data compression algorithm is to exploit the correlation among the data in time, space or frequency. Usually it is very difficult to pre-estimate the correlation between the consecutive readings received from the sensor nodes. Correlation among the consecutive observation depends upon the type of physical phenomena.

Generally there are two types of correlation in sensor networks:

1. Temporal correlation.
2. Spatial Correlation.

Spatial correlation is due to high density in the network topology. Spatially proximal sensor observations are highly correlated where the degree of correlation increases with decreasing internodal separation.

Some events like tracking application may require sensor nodes to periodically operate and transmit sensed features of the event. The nature of energy radiating physical phenomena constitutes temporal correlation between the consecutive readings. For e.g. due to the spatial correlation, data from the spatially separated sensor is more useful to the sink than highly correlated data from nodes in proximity.

For tracking applications, the measurement frequency at which node transmits the data can be adjusted such that the temporal correlated phenomena signal is captured at the sink with acceptable distortion level and minimum energy expenditure.

Most of the compression algorithm proposed so far doesn’t provide an effective data compression technique that uses the correlation in WSN. To address the above issue we have proposed a lossless data compression algorithm which utilizes the temporal correlation among the consecutive readings coming from sensor nodes and generates the unique tag or an identifier for group of symbols.The identifier is coded and sent through the channel to the sink node. The salient feature of this technique is that it clearly separates the input model from channel encoding.

The silent features of the proposed methodology are:

1. This technique utilizes the temporal correlation among the consecutive readings. It compresses the difference between the two consecutive readings. In case difference is zero, no compression is required. Hence, by passing the first sample and difference among the samples all the values can be retrieved.
2. It uses a method of tag generation, it generates a unique tag or identifier for a group of symbols and then converts it into binary for transmission to sink node. The Salient feature of the algorithm is that there is no need of passing the tree information or saving the tree and it generates the code for group of symbols not the individual entity. Hence reduces the data to be transmitted.
3. Tested the algorithm on four temperatures and relative humidity data sets collected by real WSN deployments and obtained the compression ratio as 69.82, 63.54, 68.5 (%) for Temp. Data sets and 59.3, 52.6, 54.2 (%) for humidity data set.
4. Results are validated by comparing it with other existing techniques for data compression in WSN like LEC, S-LZW and ALDC for block size 32 and 64. And some recent algorithm like FELACS and mLEC has been also been considered for comparison. The result shows that proposed algorithm give comparable results in terms of compression and takes lesser time for compression and decompression. The ratios are almost half as compared to the other already proposed techniques. As it is light weight and there is no requirement of preserving the tree information, this feature makes it a suitable choice for WSN constraints. It is computationally efficient as compared to other entropy based techniques which makes it a suitable choice.

The rest of the article is organized as follows; section-II gives the related researches in the field of data compression in the past years. This section describes the methodology used in data compression and its limitations and advantages. Section-III describes the block diagram of the algorithm used and discuses its underlying principle. In Section-IV we review & analyze and compare the performance of our compression algorithm with existing algorithms. Section-V gives the conclusion and future work in the desired field.

## 2. RELATED WORK

Due to limited available resources in WSN and low memory constraint, two types of approaches have been followed to apply the data compression technique in WSN:

- The First approach is used for dense and cooperative WSN's where the cost is distributed on the overall network. In these types of networks all the nodes communicate with each other to perform a particular series of task that they can't perform alone.
- The Second approach utilizes the statistical features of the data to be monitored, and this results in a reduction in energy consumption.

Only few algorithms present in the literature make use of the second approach. This approach helps in power saving only if the designed data compression algorithm does not require a lot of energy as compared to the energy saved in the compression process for data transmission.

In[2] gives the indepth analysis & implementation of all DWT and DCT image compression technique on TelosB hardware platform of TinyOS sensor node. Compression Ratio(C.R), throughput, PSNR (peak signal to noise ratio), battery life time and end to end delay parameters are considerd to do the analysis part. Experiement is done for single hop and multihope networks, results shows that DWT techniques are more efficient than the DCT techniques if the parameters like throughput, PSNR, battery lifetime & end to end delay are considered. On other hand if compression ratio is main concern DWT based techniques provides better results. If multimedia sensors are considered which generally transmits image frames to the sink node or base station. In WMSN( wireless multimedia sensor networks) lossy techniques are generally preferred over lossless. Many researcher applied the modification of standard SPIHT algorithm in order to minimise the limitation of SPIHT. In[3] author divides a input image into strip and each of this strip is encoded separately. SPIHT still suffers from the problem of high internal memory usage which is not appropriate for power limited application. In new[4] author introduced a listless pipelined strip based SPIHT of media senor network, which helps in reducing the complexity of system, processing time and memory requirement. By merging the refinement pass and sorting pass into single pass, will result in reduction in complexity of the system. Results shows that superiority of proposed algorithm over the others in terms of PSNR which is about 1dB for all bit rates. Memory requirement is also reduced to 71% with overall energy saving of 27%.

After analysing the conventional compression algorithm for use in compression in WSN, Barr and Asanvoic [6] concluded that compression before transmission in WSN may result in an increase in power consumption if no energy awareness is introduced, as compression may result in saving the memory space and not the power consumption of the sensor network.

In dense WSN, the data compression approach which collects and coordinates the sensor data from multiple sensors is known as Distributed Data compression. These strategies exploit the spatial correlation existing among the sensor network. While the network having less number of sensors employs a data compression technique which uses the correlation among the consecutive values also known as temporal correlation are termed as Local data compression approaches. These two can be combined as required in the application to exploit both temporal and spatial correlations among the sensor readings. While there are some issues with the distributed approaches which are not present in the Local approaches. On other hand distributed techniques which correlate the data from multiple sensors can provide the high compression ratio even when the delay is small.

Murad A. Rasam [7] proposed an adaptive dimension reduction model for WSN which is based on the CCIPCA (candid covariance free incremental PCA). They have evaluated the performance of the proposed algorithm on real word sensor data sets collected in standard laboratory and obtained the result as 33.33% and 50% reduction of multivariate data in static and dynamic environment. Author compared the algorithm with MLR ( multivariable linear regression model) and simple linear regression model(SLR) and results show that proposed algorithm has high efficiency, accuracy and adaptability with a dynamically changing environment.

Another effective approach in image compression in WCSN (Wireless Camera Sensor Network) is given by Cristian Duran [8]. This algorithm is specifically designed for resource constrained wireless camera sensor called TiBS. It operates on blocks of 2x2 pixels. Then it is combined with a chaotic pixel mixing scheme to increase the robustness of the image communication against the packet losses. To validate the results the TiBS and JPEG like algorithm have been implemented on a real world camera sensor network composed of Mica2 mote and a Cyclops imager. An experimental work shows that not only it is providing a high compression ratio, but also enables energy efficient image communication for the source camera node. As this algorithm is a local compression algorithm, there is scope of finding the options for using the TiBS algorithm for reducing the energy consumption of camera present in the source node for in-network image processing.

For effective transmission of images in wireless sensor network another adaptive compression scheme is proposed by M.Nasri[9] which is based on the wavelet image transform and distributed image compression and shares the processing task to increase the lifetime of the sensor network. Results show that proposed algorithm optimizes the network lifetime and there is a significant reduction in the memory required and computational energy by reducing the number of mathematical operations. The approach can also be applied to the real world platform to satisfy the real time constraints. It includes multipath routing to improve the performance of distributed image compression in WSN.

Amar etal [10] in 2010 proposed a distributed transform based technique; the transform based technique splits the source output from the sensor as per an existing transform concepts into the coefficients which are then coded as per their individual characteristics. One of them is Karhumen-Loeve approach which performs the task of determining the encoding matrix of each sensor node while it is assumed that other matrices are fixed. The algorithm selects the single sensor node that provides the largest contribution in minimizing the mean square error. Simulation results show that the mean square performance of the above algorithm is equivalent to the iterative approach. Such types of approaches are ideally used in image, audio and video compression algorithms, but power consumption at the node has not been evaluated yet. This algorithm has the potential of saving the power as it works mostly on the sink node or fusion center. This algorithm is more relevant to the applications which require real time operations. Also has less complexity as compared to its first counterpart based on same transform, but both the algorithms do not address the power consumption issue in WSN.

In high density wireless sensor networks, the traditional MAC protocol is insufficient in terms of energy efficiency. To address this issue novel algorithm has been proposed by Y.Peng [11] which aims to reduce the data redundancy at the source node. CCS-MAC is different from all other previous MAC protocols. The key idea involved is that it reduces the data redundancy by exploiting the overheard data which is avoided or neglected by the previous work in MAC protocols. To validate the results the comparison is done with B-MAC and S-MAC, and results show that CCS-MAC is helpful in significant improvement in lifetime of the sensor network. As the sensors are having limited power, Alexander Ciancio [12] proposed an algorithm which first selects the routing strategy, and then for every route an optimal combination of assignment at each node is selected. This provides a tool for comparing different routing techniques and identifies those which are most efficient for a particular node location. Validation of an algorithm is done on second order autoregressive model and empirical data from real sensor deployments. Simulation result of the various routing techniques were compared and the most efficient among them specifically for sensor networks were identified. They verified that proposed algorithm gives the optimum results for finding the best coding scheme that should be used by each of the nodes in the sensor network, and shortest path routing does not always result in better performance in terms of energy consumption in the network.

Another work by Ciancio[13] addresses the problem of compression in WSN. They have introduced distributed compression algorithm based on the lifting factorization of wavelet transform and it exploits the nature of data flow in the network. Simulation results show that the method can reduce the transmission cost depending on the type of network configuration being used. The main concept of this work is to perform partial computation of transform coefficients at each and every node of the sensor network. This reduces unnecessary transmission over the network which results in significant reduction of energy consumption of WSN. But this technique lacks in quantization of partial coefficients which introduces a distortion in the output.

A more recent development in the field of distributed sensor network is RIDA (Robust Data Compression for Irregular Wireless Sensor Network) proposed by Thanh Dang [14].The main idea is to determine the correlation among the data received by group of sensor nodes based on its data value rather than solely depending on the spatial correlation existing in the sensor network. In RIDA a mapping approach assigns the virtual index to the nodes based on its data value, which helps in implementation of data transformation technique on the input source. They evaluated the RIDA for both DCT (Discrete Cosine Transform) and DWT (Discrete Wavelet Transform) on a constrained node without any help as no other information was present. RIDA works on the assumption that nodes have been organized into clusters using any of the clustering protocols like LEACH [15]. It reorganizes the data from the sensor before performing compression. The three main components of the architecture are: *a logical mapping unit, a compression algorithm and a resiliency mechanism.* The two main parts of the cluster are mapping and compression. In mapping part, the cluster head designates the virtual indices to sensor and sends the map to sink for further mapping. After the mapping process the sensor node exchanges the data with their desired cluster. Data from the sensor is transformed using DWT or DCT to calculate the desired coefficient. The coefficients are quantized to its nearest integer value. A resiliency mechanism is simply a classification method to increase the robustness of the sensor network toward the missing data from the sensors. Experimental results show that 30% of energy and 80% to 95% of the bandwidth can be saved by using RIDA for any sensor data network.

Next we discuss the local approaches. These approaches address the issues left unresolved by Distributed approach. They perform the compression locally on each sensor node without any collaboration among the sensor nodes. As a result, they exploit the temporal correlation and do not depend on the specific topology of the WSN. These are suitable for generally the dictionary approaches which are used to compress the variety of data available. There are so many techniques which consist of a static dictionary and adaptive dictionary techniques such as static code have fixed length and adaptive have a dynamic code length which changes with conditions. The well known examples of dynamic dictionary technique are LZ77, LZ78, and LZW. For WSN first Barr and Asanoic (2006)[6] and Sadler [16] used the original version of these algorithms on WSN and found that these algorithms are not addressing the WSN constraints. These algorithms were implemented as per the desired requirements of WSN. Sadler [16] in 2006 proposed the S-LZW and mini LZO which is adapted version of LZW and LZ77. S-LZW splits the uncompressed input bit stream into fixed size blocks and then compresses separately each block. In this a dictionary is used which is updated every time a new entry is added to the dictionary. For each new block dictionary used is re initialized using the 256 codes that represent the same standard character set. As the sensor node is having limited memory/storage so the size of the dictionary should be as small as possible. As the sensor data that are generated are repetitive in nature, a mini cache is added to the main S-LZW to improve the performance. It utilizes the temporal correlation existing among the consecutive sensor readings.

S-LZW requires a large space in memory and delay in transmitting the packet of sensor data is very high. Schollhammer [17] developed a LTC algorithm (Lightweight Temporal Compression), which is a simple lossy compression method for monitoring applications. This technique introduces a small amount of error into each reading by using a control knob. Larger the error higher will be the saving obtained from compression. It is moreover similar to the run length encoding (RLE) because it represents the longest sequence of same datatype by a single symbol. While the difference is that LTC searches for linear trends and RLE searches for the strings of the same kind of symbols.

Marcelloni [5] proposed entropy based Lossless compression technique called LEC algorithm which exploits the temporal correlation existing among the consecutive data collected by the sensor networks. This is the first algorithm to be developed which was applicable for multiple data types. This approach is one of the first type of lossless approach, motivated by the fact that the data received from environment monitoring collected from real world deployments are temporarily correlated and thus results in predictive data compression where the incoming reading can be found out from the current reading and so on. It uses a short fixed sized code whose size depends on the output bit size of Analog to Digital converter (ADC). So it does not suffer from the growing dictionary problem as it acquires less space in the memory. But it has the limitation that the algorithm is static. If the statistics of the input sample changes it will not work effectively. It requires prior information about the statistics of the input samples like entropy, standard deviation, deviation among the samples etc.

Marcelloni [18] proposed a lossy algorithm working on a single node which is based on differential pulse code modulation (DPCM), and the difference of the consecutive samples is quantized. Algorithm exploits the multi objective evolutionary technique to get a particular tradeoff between the compression performance and information loss in WSN. Comparison results show that the algorithm, outperforms LTC in terms of compression rate and complexity of the algorithm. As the algorithm performance is best as compared to the other, but the algorithm is not adaptable to the changes in the data model .To resolve this issue the data sets should be collected by the same type of sensors and the frequency of measurements should be same as the real world WSN.

LEC has lots of advantages in terms of compression ratio and memory requirement , but the disadvantage is that the algorithm is static, it cannot adapt to the changes in the statistics of the data received. The algorithm requires prior knowledge of the data source statistics like entropy, standard deviations etc. S. Jonathan [19] proposed a simple algorithm called as ALDC(Adaptive lossless data compression) implemented in a few lines of codes. It uses the LEC compression table for performing the compression. It has same compression complexity as of LEC but has a lower compression efficiency. It is basically a lossless compression algorithm which performs compression adaptively by using the two ALEC (Adaptive LEC) coding options. One is 2-Huffmann Table ALEC and other is a 3-Huffmann Table. These two tables are an adaptive coding scheme which adaptively uses the 2-Huffman table and the 3-Huffman table as per the applied input condition. Theses tables are designed after working on many real words sensor node data sets with varying correlation among the consecutive readings. These two ALEC code options compresses the block of sample data at a particular instant. Validation of this algorithm is carried out by testing it on real world data sets obtained from the environment like relative humidity, temperature and seismic events. Results show that the compression performance of the proposed algorithm outperforms all the recently proposed lossless data schemes. It is simple and efficient and is as per the resource constraints present in sensor network. Recently another lossless data compression algorithm is proposed by S.Jonathan [20] called as FELACS ( Fast and efficient lossless adaptive data compression scheme) which uses the basic RICE codes to compress the block of data and Adaptivity is added by giving the two block size option and variable compression ratio. It is a very lightweight algorithm and has low complexity and achieves a compression rate of about 4.11 bits per sample.

As there are only a few options available for Local Lossless data compression schemes so we have proposed an efficient Tag generation scheme based on the baseline of arithmetic coding. For simulation and testing purpose the algorithm is tested on real world sensor data sets available and comparison is done to with other standard algorithm available for sensor networks.

## 3. TGS ALGORITHM:

As the well known lossless compression algorithm LEC requires the prior knowledge of the statistics of the incoming data stream from the WSN and other entropy based techniques like Huffman and adaptive Huffman based techniques send the code for each and every symbol in a stream of input and we to send the tree information along with codes or tree is required to decode the information. But here we have introduced a novel Tag generation based technique which is based on baseline of standard arithmetic codes. It generates a unique tag in the interval (0 -1) for a group of symbols, than a binary equivalent of this tag is sent to the sink node. But its efficiency depends on the number of input samples which we are considering to group together. For e.g., if we have very less number of symbols and coding with Huffman based and our techniques in that it may result in expansion in place of compression. So a perfect balance is required to choose the length of the stream so that we get the data rate closer to the entropy of the input bit stream.

Major advantage of our proposed algorithm which is adaptive to the changing statistics of input, By keeping the count of input symbols we can calculate the probabilities, there is no requirement of preserving the tree as in case of Huffman and adaptive Huffman algorithms based algorithms. This property permits us to separate the modeling and coding procedure which is not feasible with Huffman methods. This property makes the compression system more flexible and is an added advantage for our system.

## 3.1 Block Diagram:

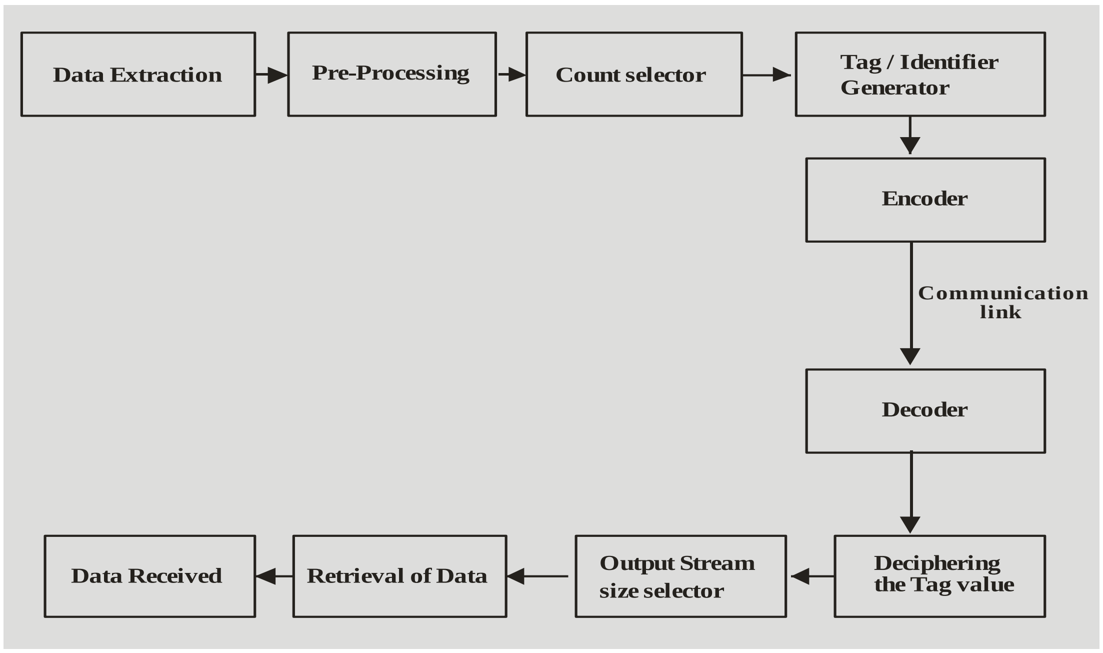


**Fig 1:** Block Diagram of Proposed Algorithm

## 3.2 Data Extraction:

Data that is used in algorithm is a real time data available at Sensorscope [22], it contains lots of information about the data like time of epoch, temperature, pressure, humidity, date and time and also the node ID (from which data is taken). As we are concerned with two parameters like temperature and relative humidity,in data base temperature and humidity exist in the fourth and fifth row, so these two quantities are extracted from huge data base for which pseudo code is given below:

```
Data Extraction ( )
 {
      Enter number of Columns;
      Enter column position of data base required;
      Read data base
      If (n==4)
        {
          get temp data();
          }
      If (n==5)
        {
          get humidity data();
          }
```

**Fig 2:** Pseudocode for Data Extraction

- Input obtained from data extraction block is analog in nature, but processing unit requires the digital data so it is converted into digital quantity as per the formula given in sensor data sheet [23]. These digital data $r_i$ and its delayed data $r_{i-1}$ is applied to Data processing block.

## 3.3 Pre- Processing:

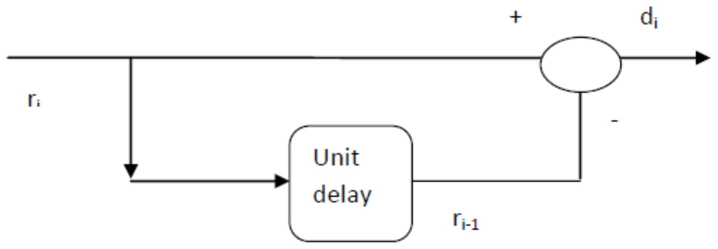


**Fig 3:** Pre-Processing block

As per the standards of data compression CCDS recommendations as in [21] any lossless algorithm should include a preprocessing step before a coder, this step helps in removing the correlation among the consecutive readings and converts the input obtained into a series of non negative integer. This series of sequence has a property that lower values have more probability as compared to large values and hence lesser amount of memory is required. Fig3 represents the preprocessor step. It generates the predicted value of any incoming signal. For e.g for any incoming data $r_i$, generally next reading is taken as the predicted value. After applying a delay, we can get $r_{i-1}$. The difference among the consecutive readings is calculated as:

$$d_i = r_i - r_{i-1} \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(1)$$

## 3.4 Count Selector:

The output obtained from preprocessor block is grouped together to form the block of length K. The factor K can be varied to get a exact balance between the entropy rate and compression ratio. Because length of block chosen play a critical role in our algorithm. If the length of sequence is too small, it may result in expansion in place of compression.

## 3.5 Tag Generation:

To distinguish a sequence of sensor input with other input we need to tag it with a Identifier, this process is called Tag Generation. This process is initiated by dividing the (0 1) interval into unique subinterval according to probability of occurrence of symbols in a group as given by

[ $F_x(i-1), F_x(i), i=1\ldots\ldots m$], Where $F_x(i)$ is a cumulative density function

$$Fx(i) = \sum_{k=1}^{i} P(X = K)..............(2)$$

As the minimum value of complementary density function is 0 and maximum value is 1, as another symbol arrives the subinterval again partitioned in the same way as mentioned earlier.

For e.g. for the first symbol $a_k$, the interval containing tag value is

[*Fx(k-1),Fx(k)]*

The jth interval to particular input $a_j$ is given by:

$$\left[\frac{Fx(k-1)+Fx(j-1)}{Fx(k)-Fx(k-1)}, \frac{Fx(k-1)+Fx(j)}{Fx(k)-Fx(k-1)}\right]$$

Every other incoming signal causes signal causes the tag to be restricted to the concerned subinterval that further partitioned in the same way. Pseudo code is for the same is given in figure4,

```
 l=0.0;
h=1.0;
while((c=getc(input))!=EOF)
   {
      range=h-l;
      h=l+range*h_range(c);
      l=l+range*l_range(c);
     }
Output(l);
```

**Fig 4:** Pseudocode for Tag generation

### 3.6 Encoder:

As already described earlier that the tag represents a unique representation for a given sequence, encoder finds an equivalent binary representation of the tag and form a unique binary code for input sequences. Some of the converted binary representations are very long. To make it more efficient there is a need of truncation of converted input. To make the truncation process efficient, a validated procedure is used. If Tx(x) is the tag value in the interval (0 1) a binary code for the tag is obtained by converting it into binary and then truncating it into length *l(x)* given by eq. No. (2)

$$l(x) = \left\lceil \frac{1}{P(x)} \right\rceil + 1 \quad \text{bits..............(3)}$$

The code resulting from above process are uniquely decodable. As the tag is a unique representation in tag interval, so its binary representation is also unique whether truncated or not. Because the truncation process results in a code the value of which is less than or equal to tag value. The pseudo code for that is given below in Fig4.:

The cumulative count $F_x(k)$ is given by

$$Fx(k) = \frac{\sum_{i=1}^{k} ni}{total\ count} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(4)$$

Where $n_j$ as the number of times the symbol j occurs in a sequence of the length Total Count.

C_Count(k)= $\sum_{i=1}^{k} ni$ ……………………………………………..This encoded message is sent via the communication link to the Decoder as explained in decoder part.

The cumulative count $F_x(k)$ is given by

$$Fx(k) = \frac{\sum_{i=1}^{k} ni}{total\ count} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots..(4)$$

Where $n_j$ as the number of times the symbol j occurs in a sequence of the length Total Count.

C_Count(k)= $\sum_{i=1}^{k} ni$ ……………………………………………..This encoded message is sent via the communication link to the Decoder as explained in decoder part.

```
Initialize the upper and lower limit l and u
Get values from
  $l \leftarrow l + \left\lfloor \frac{(u-l+1)X\ C\_Count(x-1)}{Total\ count} \right\rfloor$
                    $u \leftarrow \left\lfloor \frac{(u-l+1)XC\_Count(x)}{Total\ Count} \right\rfloor$
While (MSB of upper and lower limit u and l are equal to b or E3 condition hold)
If (MSB of u and l are equal to b)
{
  Send the value b
  Shift the low l by 1 bit and shift 0 into LSB
   While(scale3>0)
   {
     Send complement of b
      Decrement scale3
    }
}
If ( E3 condition holds true)
{
Shift l to left by 1 bit and shift 0 into LSB
Shift u to the left by 1 bit and shift 1 into LSB
Complement (new) MSB of l and u
Increment scale3
}
```

**Fig4:** Pseudo code for Encode function

### 3.7 Decoder:

Decoder receives the encoded message and convert it back it into the tag value. In this also pattern is more ever similar to first model is set up than these values are passed to the decoder. The pseudo code is given below in fig.5:

```
Initialize lower and upper limit l and u
Read the starting m bits of the received input into the tag t
K=0
while$\left(\left(\frac{(t-l+1)X\ Total\ Count-1}{u-l+1}\right) \geq C\_Count(k)\right)$
$k \leftarrow k+1$
Decode the given symbol x
$l \leftarrow l + \left\lfloor\frac{(u-l+1)X\ C\_Count(x-1)}{Total\ count}\right\rfloor$
$u \leftarrow l + \left\lfloor\frac{(u-l+1)XC\_Count(x)}{Total\ Count}\right\rfloor - 1$
While (MSB of upper and lower u and l are both equal to b or condition E3 condition hold)
If(MSB of u and l are both equal to b
{
  Shift the l to left by 1 bit and shift 0 into LSB
  Shift the u to left by 1 bit and shift 1 into LSB
 Shift the t to the left by 1 bit and read next bit from received bit stream into LSB
}
If (E3 condition holds)
{
Shift l  left by 1 bit and shift 0 into LSB
Shift u left by 1 bit and shift 1 into LSB
Shift t left by 1 bit and read the next bit from received bit stream into LSB
complement MSB of l, u and t
```

**Fig 5:** Pseudocode for Decoder

### 3.8 Deciphering the Tag:

Deciphering is the process to retrieve the sequence of symbol from a given tag value. The Tag generation process is useless, unless we can decipher it with lesser amount of computation. In this we first check the range in which this symbol falling, which gives the low and high range of probability, from which type of symbol is found out. The process continues till the whole sequence is generated. Pseudo code is given in fig6:

```
N=input the code
for( ; ; )
{
symbol =find the symbol in this range (number)
putc (symbol)
range = high range(symbol)-low range (symbol)
n=n-low range (symbol)
n=n/range
}
```

**Fig 6:** Pseudocode for Deciphering the Tag

### 3.9 Output Count Selector:

Output count selector converts back the series of input into individual symbols. This is passed to the next block for further processing.

### 3.10 Retrieval of Data:

Data can be retrieved from the differences by passing the first sample value to the receiver. By using the differences and first sample value other sample values can be find out by using simple computations.

**3.11 Data received:** Data retrieved by retrieval block is collected in the data received block for further processing or future requirements. The data retrieved is same as and therefore no loss of information occurred along with effective power saving during the transmission via the compression process.

## 3.12 The Flow Chart of Proposed Algorithm:

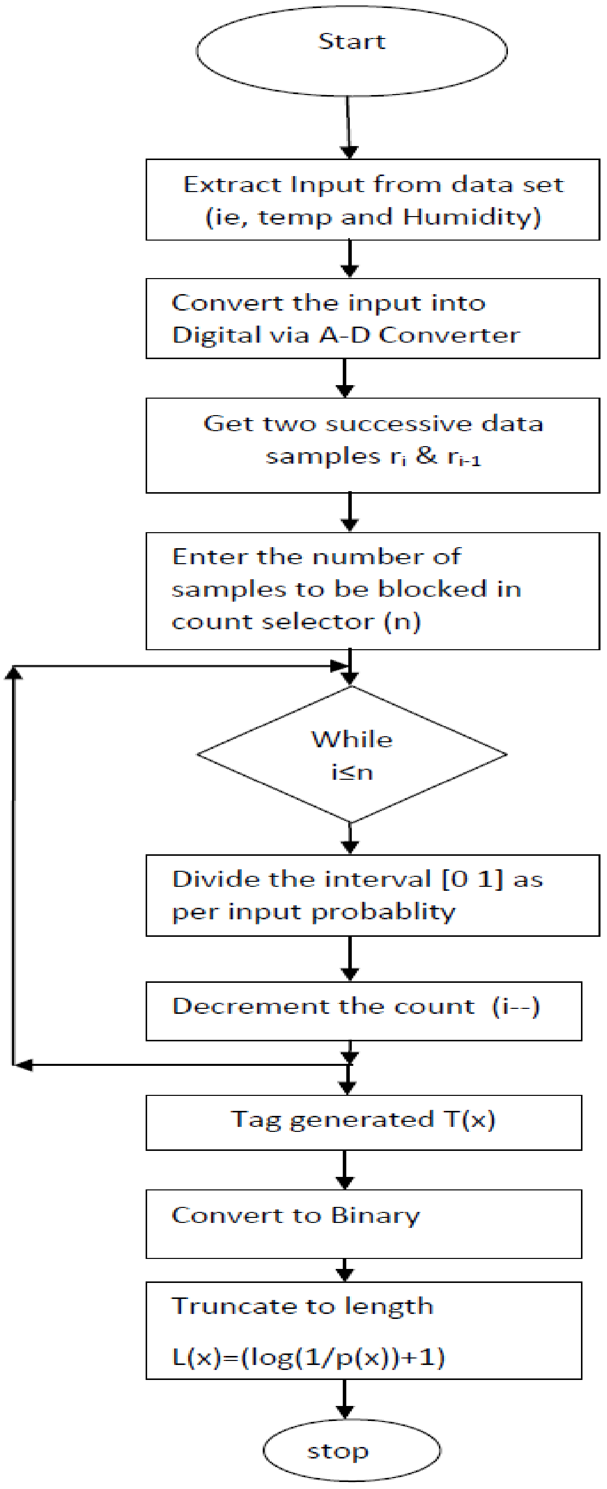

Start
Extract Input from data set (ie, temp and Humidity)
Convert the input into Digital via A-D Converter
Get two successive data samples $r_i$ & $r_{i-1}$
Enter the number of samples to be blocked in count selector (n)
While i≤n
Divide the interval [0 1] as per input probablity
Decrement the count (i--)
Tag generated T(x)
Convert to Binary
Truncate to length L(x)=(log(1/p(x))+1)
stop

**Fig 4.** Flow chart for proposed algorithm

## 4. RESULTS AND COMPARISON:

In order to check the effectiveness and validity of our proposed Lossless compression algorithm, we have tested it against a variety of real world sensor data sets. The data set used is obtained from real WSN deployements. We considered the signals like Temperature, relative humidity which are smooth signals, particularly suited to our algorithm. We have considered the four data sets from deployments of sensor networks, downloaded from sensorscope.com, named as Genepi Deployment, HES SO Fishnet deployment and LUCE deployment. These are publicly available for use. These deployments use a Tiny node [23] which consist of a TI SP430 microcontroller a XemicsXE1205 radio and a sensor named as Sensirion SHT75 sensor module. The output here is connected to a 14bit ADC (Analog to digital converter). Codes for the proposed encoder and decoder are developed using C language. The data set contains the analog data and algorithm works on the digital data so before compression it is converted to digital data by using the inversion formula given in sensor sheet[23].The salient features of the data set as obtained from [22] are expressed as in Table 1

**Table 1:** Salient characteristic of data sets received from real sensor world network deployment

| Deployment name | Node ID | Name | No of Samples | Time Interval |
|---|---|---|---|---|
| LUCE | 84 | LU ID-84 | 64913 | 23-Nov to 17 Dec 2006 |
| FISH NET | 101 | FN ID-101 | 12652 | 09-Aug to 31-Aug-07 |
| Le genepi | 20 | LG ID-20 | 21523 | 04-Sep to 03-Oct 2007 |
| Gr. Bernard | 10 | Gr B ID-10 | 23813 | 15 Sep to 18 Oct 2007 |

The statistical information of the input data set like the entropy H of the original data sets and entropy of preprocessed data obtain as $H_{pd}$ using the standard formulas for entropy as:

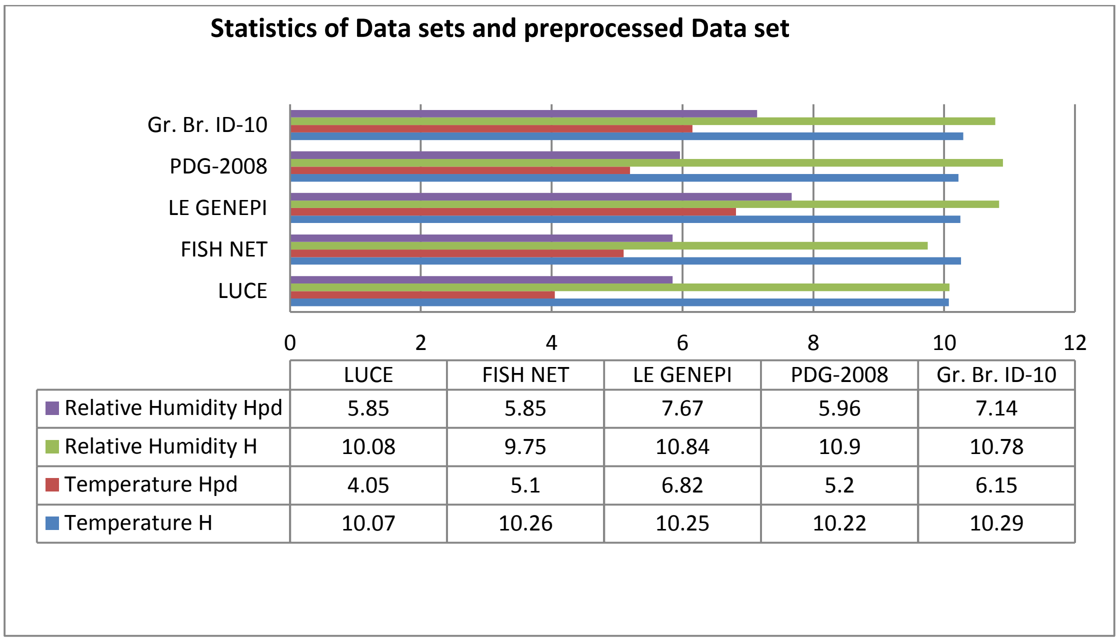
$$H=\sum p \quad (x_i) \quad \log_2 p(x_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (6)$$

Where i=1 to L

$$H_{pd}=\sum p \quad (pd_i) \quad \log_2 p(pd_i) \ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (7)$$

Where i=1 to L

Where L=Number of possible values of input.All the entropy values are compared in Fig5 for different deployements. Preprocessing is done as per the CCDS [21] recommendations to improve the compressibility of any lossless data compression algorithm.



| | LUCE | FISH NET | LE GENEPI | PDG-2008 | Gr. Br. ID-10 |
|---|---|---|---|---|---|
| 🟪 Relative Humidity Hpd | 5.85 | 5.85 | 7.67 | 5.96 | 7.14 |
| 🟩 Relative Humidity H | 10.08 | 9.75 | 10.84 | 10.9 | 10.78 |
| 🟥 Temperature Hpd | 4.05 | 5.1 | 6.82 | 5.2 | 6.15 |
| 🟦 Temperature H | 10.07 | 10.26 | 10.25 | 10.22 | 10.29 |

**Fig5:** Statistics of Data set and preprocessed Data Set

## 4.1 Compression Ratio (C.R):

To demonstrate the effectiveness of designing algorithm the factor called compression ratio is calculated for different set of real word environmental database available [22].Compression ratio is defined as the ratio of compressed size to uncompressed size given by

$$C.R=100. (1-COMP\ SIZE / DECOMP\ SIZE)\ \%$$

Where,

COMP SIZE= compressed size

DECOMP SIZE= De Compressed size

C.R= Compression Ratio in %

The Compressed and uncompressed size is calculated by considering that the uncompressed samples are byte aligned and temperature and humidity both represented by 16 bit unsigned integers. The effectiveness of a compression algorithm is usually computed by a factor known as compression ratio. Our approach generates the Tag for Block of input chosen by count selector and this tag is converted into an equivalent digital form and truncated to a particular number of symbol suitable for sending to the sink node. Results shows that for block of data our algorithm gives the comparable compression ratio and requires less time in data transmission as there is no need to transmit the tables and update tree information as in other entropy based techniques like LEC and ALDC. Major advantage of our proposed technique is that it separates the modeling and coding process as compared to the other entropy based techniques in which coding depends upon the input modeling. Table 2 shows the compression ratio obtained for different set of real world data available.

**Table 2:** Compression ratio obtained from the proposed algorithm.

| S.No. | Name of Deployment | Compression Ratio (CR)% for Temperature | Compression Ratio (CR)% for Relative Humidity |
|---|---|---|---|
| 1. | LU ID-84 | ----------- | ----------- |
| 2. | FN ID-101 | 70.54 | 60.2 |
| 3. | LG ID-20 | 66.54 | 56.8 |
| 4. | PDG-2008 | 68.32 | 57.4 |
| 5. | GrB-ID10 | 69.32 | 58.4 |
| | | | |

Other techniques like LEC and ALDC and other Huffman based techniques results in exponential growth in size of codebook as the length of input sequence increases. As our technique assigns a unique code word for a length of the sequences this feature make our algorithm a energy aware compression algorithm which helps in increasing the lifetime of a sensor node.

## 4.2 Performance Comparison with other standard Algorithm:

The proposed Tag generation scheme is validated by comparing it with some standard lossless algorithm available for sensor networks like LEC (Loss less entropy coding), SLZW (sensor LZW) and ALDC (Adaptive Lossless data compression algorithm) for two block option 32 and 48 block size. Some recent proposed algorithm like mLEC and FELACS which have been already explained in literature survey part have also been considered for comparison. Results show that the proposed algorithm outperforms in terms of compression ratio and it is computationally efficient as there is no need to send the code table or update tree information to the decoder as in LEC or other Huffman based techniques. The encoding time is approximately half compared to the other mentioned techniques. Table 3 gives the comparison of the proposed algorithm with the other standard algorithm proposed for sensor networks. Fig6. and Fig7. gives the graphical comparison of all the available algorithms with proposed method.Another major advantage that the proposed scheme easily adapts to the changing input statistics. For that it is required to find the probability of input symbols and this can be done by tracking the count of input symbols as they are coded. There is no requirement of preserving the tree information as in the LEC, ALDC and other Huffman and adaptive Huffman based techniques, and also there is no need of generating the code priori as in Huffman. This property separates the coding and modelling process which is not feasible in LEC or other Huffman based technique (like ALDC). This feature incorporates greater flexibility in the compression system which is an added advantage of our system, and which makes it a robust compression scheme. It also supports multiple data types ie, single algorithm works for both humidity & Temperature data base and in future the same algorithm can be applied to non-smooth data sets like seismic and ECG signals.

**Table3:** Comparison of Proposed Algorithm with other standard Algorithm

| S.No. | Name of Deployment | TGS | | LEC | | SLZW | | ALDC block size 32 | | ALDC block size 48 | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Temp | Hum | Temp | Hum | Temp | Hum | Temp | Hum | Temp | Hum |
| 1. | LU ID-84 | ----- | ----- | 70.81 | 62.8 | 70.81 | 62.8 | 73.87 | 65.50 | 73.94 | 65.94 |
| 2. | FN ID-101 | 70.54 | 60.2 | 65.39 | 62.95 | 65.39 | 62.95 | 67.44 | 66.28 | 67.48 | 66.33 |
| 3. | LG ID-20 | 66.54 | 56.8 | 53.83 | 48.7 | 53.83 | 48.7 | 56.86 | 52.80 | 56.90 | 52.87 |
| 4. | PDG-2008 | 68.32 | 57.4 | ---- | ----- | ---- | ----- | -------- | -------- | -------- | ------- |
| 5. | GrB-ID10 | 69.32 | 58.4 | 59.03 | 52.71 | 59.03 | 52.71 | --------- | --------- | --------- | ------- |

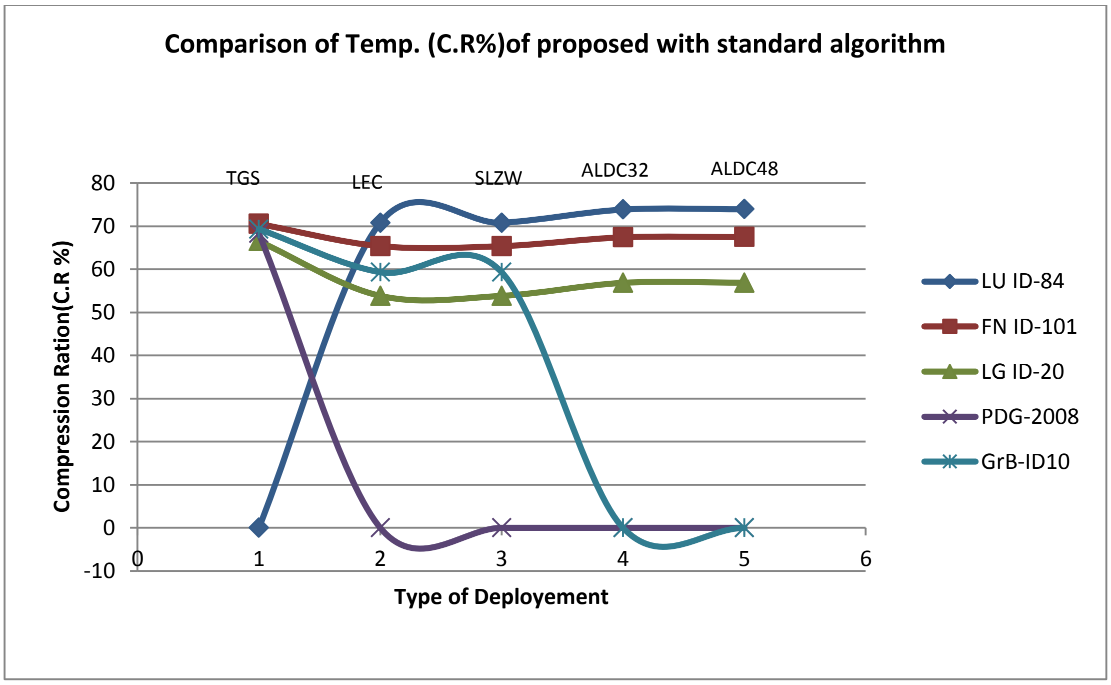


**Fig 6.**

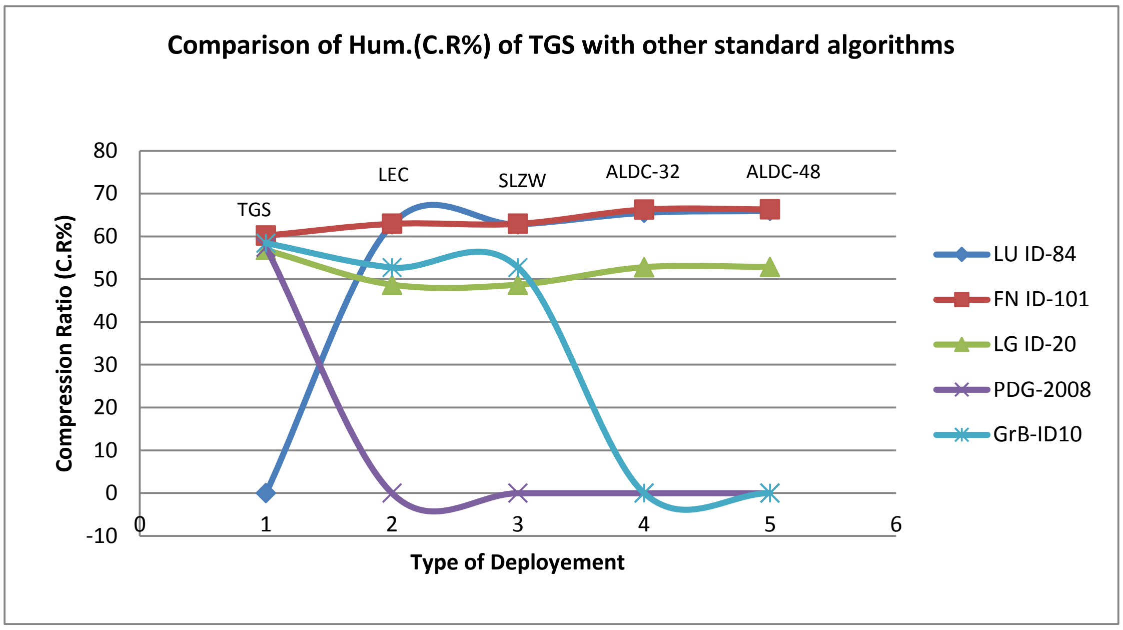


**Fig7.**

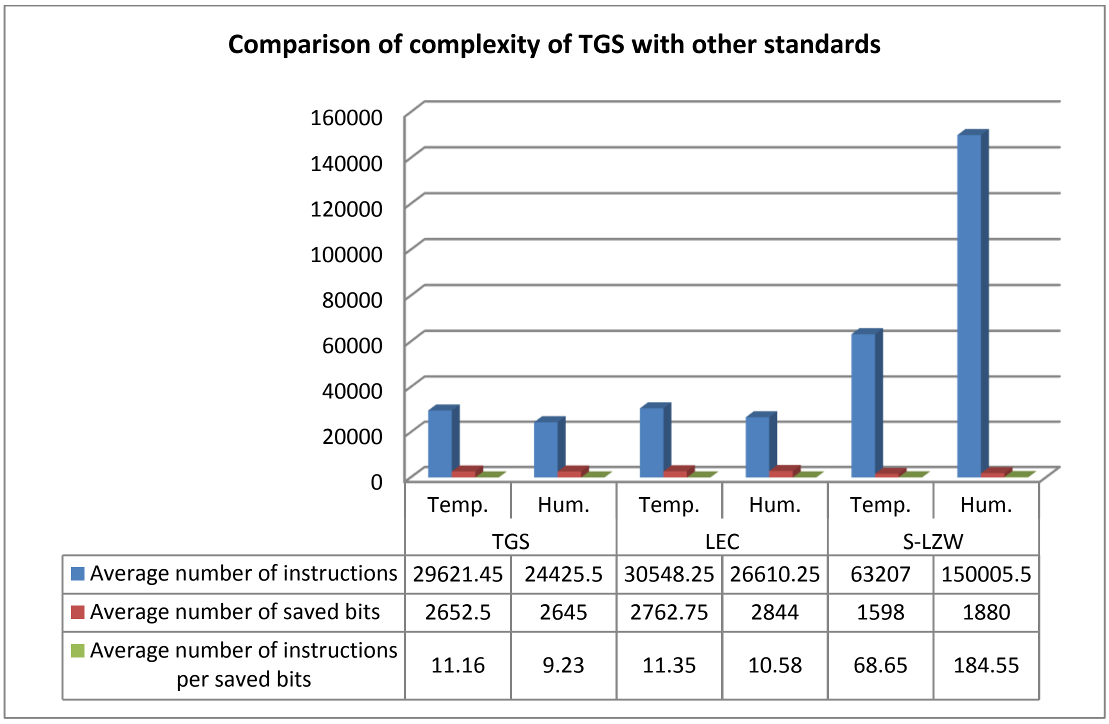
## 4.3 Memory Requirement and Power Consumption:

The implementation is done with 8 data sets, four for temperature and four for relative humidity data sets. As taken from[5] Sensor LZW(S-LZW) requires a MINI CACHE=32, MAX DICT ENTERIES=512,BLOCK SIZE= 528 BYTES and DICTIONARY STRAGEES=FROZEN[5,16]. As the block size is 528 bytes with each of block is equivalent to 528 measurements so the delay is very high when sink receives the data. For e.g. If a sensor takes measurements in every 30sec the sink receives the data in 4 hrs which results in high memory requirement and this is very unsuitable as per WSN constraints.

LEC for temp database requires 30548.25 average instructions, the number of saved bits is 2762.75 and average instruction per saved bits is 11.35. For Relative humidity data base average 26610.25 instructions are required and average saved bits are 10.58. So, LEC is less complex and the length of its code is also very suitable as per the WSN constraints. But it is not adaptive to the changing statistics of the input data sets.

As far as ALDC is concerned, it requires high memory as compared to LEC as it uses the three Huffman tables to add adaptivity in the system, but memory requirement is less as compared to the SLZW. In terms of compression ratio, it saves a good amount of power and it is adaptive to the changing correlation and statistics of input as the block size can be varied to get the desired result.

TGS requires 29621.45 average number of instructions and number of bits saved is 11.16. Similarly for the Relative humidity data base 2425.50 average instructions are required and average saved bits is 2645 and the number of instructions for each saved bits is 9.23. Fig8. gives the detailed comparison of complexity of the explained algorithms.



| | TGS | | LEC | | S-LZW | |
|---|---|---|---|---|---|---|
| | Temp. | Hum. | Temp. | Hum. | Temp. | Hum. |
| 🟦 Average number of instructions | 29621.45 | 24425.5 | 30548.25 | 26610.25 | 63207 | 150005.5 |
| 🟥 Average number of saved bits | 2652.5 | 2645 | 2762.75 | 2844 | 1598 | 1880 |
| 🟩 Average number of instructions per saved bits | 11.16 | 9.23 | 11.35 | 10.58 | 68.65 | 184.55 |

**Fig8.**

The proposed algorithm is little bit complex in terms of Tag generation process due overflow and underflow errors, but results obtained depends on how we are using it. For the long length of the sequence, Huffman based techniques result in the exponential rise in code book and hence rise in memory. If there is some perturbations in the input statistics and probability changes, it has a major effect on efficiency of code. So, the proposed technique is more efficient as it generates the code word for a group of sequences rather than separate code for each input symbol in the sequence and requires approximately 250 µs of time for coding and 290µs of time for decoding the sequence which is approximately half of the other mentioned techniques. It gives good results when we have two or more parallel systems using the same code.

## 4.4 Energy Saved by compression:

As we are considering the Tiny OS for Tmote sky (Telos B) we transmit or receive a packet of size 38 bytes. Energy consumed by packet is calculated as:

$$E_p=38.E_b \text{ ……………………………….(1)}$$

Where $E_p$=energy consume by a single packet in joules

$E_b$= energy consumed by a single byte in joules

As, $E_b=V.c_{tx}.t_{tx}$ ……………………………………….(2)

Where $c_{tx}$ and $t_{tx}$ are the current consumption and time spent for transmitting one byte.

For CC240 radio,

We have $c_{tx}$=0.0174mA and $t_{tx}$=0.032ms, V=3V

Thus, $E_p$= 38x1.6704x$10^{-6}$ =63.475x$10^{-6}$ Joules…………………………..(3)

Consider the case for FN_ID101deployement, we have to send the 873 packets each of size 38byte each( from Table1), so node consumes (original packet size is 873)

$E_{Twc}$= 873x63.475x10-6 J = 0.055413(energy without compression)………….(4)

If data has been compressed to a size of 259 packets (from Table3), so a energy consumed by compression is given by:

$E_{Tc}$= 259x63.475x10-6=0.016436 Joules (Energy with compression)……………(5)

As explained in the above section complexity, proposed algorithm requires 29621.45 instructions for a block of 528 bytes. This is equivalent to 56.10 instructions for a byte of the input signal. So for all the input samples it requires 709,777.2 instructions. By assuming that one instruction approximately requires 15nJ of energy.

Thus to compress the input mote consumes

$E_c$=709777.2x15x10-9 =0.010646 Joules………………………..(6)

Energy saved, $E_s$ = ($E_{Twc}$-$E_{Tc}$-$E_c$)=0.028337 Joules. ( From eq. 4,5 and 6)

So concluding by the saved energy=$E_s$/$E_{Twc}$ approximately equal to **51%** of the energy consumed to transmit the data.

## 4.5 Comparison with classical Text algorithm

This section presents comparison of proposed methodology with the well known classical compression algorithm from [5]. Gzip,bzip2, Rar and Huffman are considered for comparison with our approach for the same database available.Table4 gives the comparison results.

**Table 4:** Comparison of Tag Generation scheme with classical Algorithm for given database

| **S.No** | **Algorithms** | **Compression Ratios FN_ID101** | | **Compression Ratios LGID_20** | | **Compression Ratios Gr.Ber ID_01** | |
|---|---|---|---|---|---|---|---|
| | | **Temp.** | **Humidity** | **Temp.** | **Humidity** | **Temp.** | **Humidity** |
| 1. | Gzip | 34.76 | 41.29 | 31.38 | 27.61 | 34.35 | 31.02 |
| 2. | Bzip2 | 55.20 | 56.22 | 46.84 | 42.56 | 52.12 | 45.73 |
| 3. | Rar | 63.59 | 59.12 | 51.56 | 42.56 | 56.72 | 49.70 |
| 4. | Huffman | 21.59 | 23.19 | 22.34 | 18.85 | 22.32 | 18.97 |
| 5. | **TGS** | **69.82** | **59.3** | **63.54** | **52.6** | **68.5** | **54.2** |

## 5. CONCLUSION AND FUTURE WORK:

In this paper, we have presented a Lightweight novel tag generation scheme specific for wireless sensor network constraints. In this a unique tag is generated, which is converted into a binary code of  of sequences. Unlike the other entropy based techniques a unique code is generated for a length of sequence without any need of generating code for all the sequence of a  particular length.

The algorithm is adaptive to changing statistics and is fast and takes less amount of time to compress the data. A better compression ratio is obtained which result in decreased power consumption which makes it as a suitable choice for battery operated wireless sensor networks. We have implemented the given algorithm for four data set of Temperature and relative humidity as obtained a high compression ratio as compared to other standard algorithm already proposed in WSN.

In future further changes can be made in tree update process to reduce the complexity of the update process. The same algorithm can be applied to the other type of non smooth signals like ECG, Solar radiation, Seismic data, etc. In future it can be implemented it on real WSN hardware.

## REFERENCES

[1] Tossaporn Srisooksai, Kamol Keamarungsi, Poonalap Lamsrichan and Kiyomichi Araki “Practical data compression in Wireless sensor networks: A Survey” in journal of network and computer application,(35),2012, pg. 37-59.

[2] Tarek Sheltami, Muhammad Musaddiq, Elhadi Shakshuki “Data Compression technique in Wireless Sensor Networks, in journal of Future Generation Computer Systems”, 2016, pg 151-162.

[3] L. Chew, W. Chia, L. Ang, K.Seng, Very low memory wavelet compression architecture using strip based processing for implementation in wireless sensor networks, EURASIP J. Embedded systems, 2009(1),pg 1-16.

[4] Hanaa ZainEldin, Mostafa A. Elhosseini, Hesam A. Ali, " A modified listless strip based SPIHT for wireless multimedia sensor networks" in Computer and Electrical engineering journal, Elsevier, 2016, pg 519-532.

[5] Franceso Marcelloni and Massimo Vechhhio "An efficient lossless compression algorithm for tiny Nodes of monitoring Wireless sensor networks" in the computer journal, 2009, 52(8), pg969-87.

[6] Barr KC Asanvoic K "Energy Aware Lossless data compression, ACM Transaction on computer systems", 2006,( 24),pg 250-91.

[7] Murad A Rasam, Anazida Zainal, Mohd Aizani Maroof "An adaptive and efficient dimension reduction model for multivariate wireless sensor network applications",in journal of Applied Soft Computing, 2013, (13),pg 78-96.

[8] Cristian duran-Faundez, Vincent Lecuire and Francis Lapage "Tiny block size coding for energy efficient image compression and communication in wireless camera sensor network" in journal of Signal Processing: Image communication, 2009,(26),Pg. 466-471.

[9] Mohsen Nasri, Abdelhamid Helali, Halim Sghaier and Hasen Maaref "Adaptive image compression technique for wireless sensor networks, 2011, (37), pg 798-810.

[10] Alon Amar, Amir Leshem and Michael Gastpar "Recursive Implementation of the Distributed Karhunen-Loeve Transform" in IEEE transaction on Signal Processing,(58),2010,pg 5320-5330.

[11] Y., Hu,R.Li, S.Wang Zhou and Y.Ping Lin " CCS-MAC: Exploiting the overhead data for compression in Wireless sensor networks" in journal of Computer communication, 2011,(34),pg 1696-1707.

[12] Alexandre Ciancio, Sundeep Pattern, Antonio Ortega and Bhasker Krishnamachari " Energy efficient data representation and routing for wireless sensor network based on a Distributed Wavelet Compression Algorithm" in International conference on information processing in sensor networks.(ISPN). ACM press, 2006, pg 309-316.

[13] Alexander ciancio and Antonio Ortega " A Distributed Wavelet Compression algorithm for wireless multihop sensor network using lifting" in in International conference on acoustics, speech and signal processing(ICASSP) VOL 4, 2005, pg IV 633-36.

[14] Thanh Dang, Nirupama Bulusu and Wu-chi Feng " Robust Data compression for Irregular Wireless sensor networks using Logical mapping in Hindawi journal of Sensor networks, 2013, 18 pages.

[15] W.R Heinzelman, A chandrakasan and H. Balakrishan,"Energy efficient communication protocol for wireless microsensor networks", in proceedings of the 33rd annual Hawaii International conference on System Siences(HICCS),2000,(2),pg.223.

[16] Chirstopher M. Sadler and Margaret Martonosi "Data Compression algorithm for energy constrined devices in delay tolerant networks" in proceedings of the ACM conference on embedded networked sensor systems(SenSys), 2006, pg 265-78.

[17] Tom Schoellhammer ,Ben Greenstein, Eric Osterweil, Mike Wimbrow and Deborah Estrin "Lightweight Temporal Compression of Microclimate Data sets" in proceedings of IEEE International conference on Local computer Networks (LCN'04) ,2004,pg 224-516.

[18] Francesco Marcelloni and Massimo Vecchio "Enabling energy effient and lossy aware data compression in wireless sensor network by multiobjective evolutionary optimization" in Journal of Information Sciences,2010,(180), 1924-1921.

[19] Jonathan Gana Kolo,S. Anandan Shanmugam, David Wee Gin Lim, Li-Minn Ang, and Kah Phooi Seng " An adaptive Lossless Data compression scheme for wireless sensor networks" in Hindawi Journal of Sensors, 2012, 20 pages.

[20] Jonathan Gana Kolo, Anandan Shanmugam, David Wee Gin Lim, Li-Minn Ang , "Fast and Efficient lossless adaptive compression scheme for wireless sensor networks" in Jounal of Computers and Electrical Engineering,2014,13 pages.

[21] Khalid Sayood "Introduction to Data compression" third edition The Morgan Kaufmann Series in Multimedia Information and Systems, 2006.

[22] Sensor Scope Deployments homepage(2014) as available on http://sensorscope.epfl.ch/index.php/Main_Page. (Accessed Nov. 2014).

[23] TinyNode homepage (2009) as available on http://www.tinynode.com (accessed Nov, 2014).