In WSN data transmission is the most energy-expenditure operation. In this regard, many energy efficient research solutions have already been proposed in order to minimize the data traffic. These solutions can be classified into two main categories: (i) scheduling and (ii) in-network data processing.

The scheduling based methods aimed to provide an energy efficient sleep/wakeup plan for sensor nodes. In-network data processing, which is the focus of this section, is classified in three main sub-categories as follows:

(a) Query-driven data reporting: In query-driven approaches, sensor node senses the phenomena, collect and stores data, and finally transmit data in response to a query.

(b) Data aggregation: Data aggregation is the simplest type of in-network processing, which combines data from different sources or nodes into a single entity. Data aggregation methods exploit aggregation functions like as SUM, AVERAGE, MIN/MAX to aggregate data and transfer only the aggregated results.

(c) Data compression: There are applications like as weather prediction, which require raw data transmit to be transmitted to the base station. For such applications, data compression techniques have been proposed.

Data compression is a process that reduces the amount of data in order to reduce data transmitted and/or decreases transfer time. Due to the limited processing and storage resources of the sensor nodes, data compression algorithms for wireless sensor networks aim to find an efficient way to compress data for reducing node energy costs and improving the synthesized ability of the whole system. Meanwhile, the accuracy must be guaranteed when data is being decoded.

Although previous researches has studied and compared some of existing data compression algorithms[][] but they do not provide a comprehensive and updated classification and review on compression techniques. In this section, we are going to classify existing data compression techniques and study their performances to find out why we need to use CS and in which context it is a preferred compression technique.

Existing data compression techniques can be broadly classified into two categories: (i) lossless compression and (ii) lossy compression. Each of these classes can be subdivided to distributed and local approaches. In following we will describe each of these categories and at the end we will compare them from different perspectives.

**\subsection{Lossless data compression techniques}**

The Lossless algorithms, as the name indicates compresses data without any loss of data. Such applications which requires higher precision typically adopt these schemes. It is obvious that higher compression ratios cannot be achieved in case of requiring higher precision. When the application, with high certainty, needs to achieve the original data after decompression, lossless compression methods are the only choice. The rough answer to the question when to use lossless data compression methods is: we use them for digital data, or when we cannot apply lossy methods for some reasons. \\

Lossless data compression is the size reduction of data, such that a decompression function can restore the original data exactly with no loss of data. Lossless data compression is used ubiquitously in computing, from saving space on your personal computer to sending data over the web, communicating over a secure shell, or viewing a PNG or GIF image as well as wireless sensor networks. The basic principle that lossless compression algorithms work on is that most of data gathered from the environment will contain duplicated or redundant information that can be condensed using statistical modeling techniques which determine the probability of a character or phrase appearing. These statistical models can then be used to generate codes for specific characters or phrases based on their probability of occurring, and assigning the shortest codes to the most common data. Such techniques include entropy encoding, run-length encoding, and compression using a dictionary (Figure \ref{fig:LosslessMethods}). \\

\begin{figure}[h]

\centering

**\includegraphics**[width=0.75\textwidth]{LosslessMethods}

\caption{Lossless Data Compression Techniques}

**\label{fig:LosslessMethods}**

\end{figure}

**\subsubsection{Local Methods}**

This category takes advantage of strong temporal correlation that occurs in many WSN scenarios. Exploiting the redundancy among samples from a single node can lead to further reduction in the volume of transmitted data in a WSN. Lossless compression ensures the accuracy of information through the compression and decompression process. There has been extensive research on adapting lossless compression techniques to the reduced storage and computational resources of sensor nodes. Lossless local compression techniques are categorized in three different classes: (i) Dictionary based, (ii) entropy based and (iii) run length based compressions.

**\paragraph**{Dictionary Based}

Dictionary-based code compression techniques provide compression efficiency as well as fast decompression mechanism. The basic idea is to take the advantage of commonly occurring instruction sequences by using a dictionary. The repeating occurrences are replaced by a code word that points to the index of the dictionary that contains the pattern. The compressed program consists of both code words and uncompressed instructions. LZ77 and LZ88 are two example of this category. LZ77 introduces a sliding window as dictionary which keeps last N bytes of data.

The most famous dictionary-based lossless compression algorithm is the Lempel–Ziv–Welch (LZW) algorithm. Actually, LZW is the result of some modifications made by Welch [32] to LZ77 [33 ] and LZ78.

Sensor LZW[16] (SLZW) extends the Lempel-Ziv-Welch (LZW) algorithm, which encodes new data based on previously encountered data. The authors introduce a lossless compression algorithm, which is an adapted version of LZW [27] designed specifically for resource constrained sensor nodes. SLZW processes small blocks of data to accommodate the memory constraints of a sensor. SLZW improves the compression of sensor data by using the Burrow-Wheeler transform[], which reorganizes sensor data in a way that results in better compression. SLZW is a lossless compression algorithm and exploits only temporal correlation between readings produced by an individual sensor. It uses adaptive dictionary techniques with dynamic code length. The dictionary structure allows the algorithm to adapt to changes in the input and to take advantage of repetition in the sensed data. However, the algorithm suffers from the growing dictionary problem and its compression efficiency still needs to be improved. Although it is possible to apply SLZW across sensors to exploit spatial correlation, the approach will be inefficient. The algorithm is easy to implement and its performance in latency, memory, compression effect are all suitable for WSN, but its coding efficiency still needs to be improved. \\

According to the need for fully recovering the compressed numerical data, based on the LZW (Lempel-Ziv-Welch) algorithm, Yan-Li et al [] propose an improved lossless data compression algorithm for WSN nodes. In this algorithm, calculating increment between two adjacent data of sample sequence reduces the span of data to be compressed. They address following improvements in the encoding process:

(i) All single characters will no longer be put into the dictionary at first, which can reduce the dictionary size and also be correctly decoded.

(ii) Select appropriate dictionary capacity. As nodes are limited in memory space, it is suitable to store the dictionary with address space of two bytes, which is sufficient for compression in nodes.

(iii) Limit the length of substring in the dictionary, because appropriate length can both save memory space and maintain high efficiency. The length is not determined, but can only be got through testing. For numerical data, they adopt different method to reduce the data range, which is beneficial to improve data duplication possibility and reduce the size of dictionary.

(iv) Carry out preprocessing for different situations to convert other non-document file into text file. \\

Table \ref{table:Dictionary based Methods} compare the performance of these algorithms.

\begin{table}[h]

\centering

\caption{Performance Comparison of Performance Comparison of Distributed Source Coding Methods}

**\label{table:Dictionary based Methods}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net of Saving \\ \hline

\multicolumn{1}{|c|}{6} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{N} \\ \hline

\multicolumn{1}{|c|}{7} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

**\paragraph**{Entropy based}

An entropy based lossless encoding assigns codes to symbols so as to match code lengths with the probabilities of the symbols. Huffman coding and arithmetic coding are the two popular entopy encoding. \\

In [8] Lossless Entropy Compression (LEC) introduced Huffman coding into wireless sensor nodes. Their simple lossless entropy compression (LEC) algorithm which was based on static Huffman coding exploits the temporal correlation that exist in sensor data to compute a compressed version using a small dictionary. The algorithm was particularly suitable for computational and memory resource constrained sensor nodes. The algorithm is static. Hence, it cannot adapt to changes in the source data statistics. In the paper [9], the proposed algorithm was a modiﬁed version of the classical adaptive Huffman coding. The algorithm does not require prior knowledge of the statistics of the source data and compression is performed adaptively based on the temporal correlation that exists in the source data. The drawback of this algorithm is that it is computationally intensive. In [10], the authors propose a compression algorithm that uses median predictor to d-ecorrelate the sensed data. The proposed algorithm is simple and can be implemented in a few lines of code and uses the LEC compression table. The algorithm has similar compression complexity as LEC but lower compression efficiency. \\

In [11], the authors proposed a scheme called two-modal transmission (TMT) for predictive coding. In the ﬁrst modal transmission, called compressed mode, the compressed bits of error terms falling inside the interval [−R, R] are transmitted. In the second modal transmission, called non-compressed mode, the original raw data of error terms falling outside the interval [−R, R] are transmitted without compression. The sink node is responsible for computing the coefficient values of the linear predictor. Arithmetic coding is chosen as the coding scheme and an optimal M-based alphabet is applied. The drawback of this compression algorithm is that it is computationally intensive. As such, to implement the scheme in WSNs, the sink node, which is not energy-limited, searches for the optimal predictor’s coefficients, the optimal bound R and the optimal M for M-based alphabet coding. These optimal parameters are then transmitted to other sensor nodes to enable them to perform predictive coding based on the two-modal transmission algorithm.\\

Authors in [12] propose a new lossless data compression algorithm for WSNs called Adaptive Lossless Data Compression (ALDC) algorithm. This algorithm adapts to changes in the source data statistics to maximize compression performance. ALDC algorithm operates in one pass using multiple code options adaptively and can be applied to multiple data types. With this improvement, their proposed ALDC algorithm outperforms the LEC algorithm. ALDC scheme performs compression losslessly using two adaptive lossless entropy compression (ALEC) code options adaptively. The two ALEC code options are called 2-Huffman table ALEC and 3-Huffman table ALEC. The 2-Huffman table ALEC and the 3-Huffman table ALEC are both adaptive coding scheme that adaptively uses two Huffman tables and three Huffman tables, respectively. The proposed algorithm can be used in monitoring systems that have different types of data and still provides satisfactory compression ratios. Furthermore, the proposed ALDC algorithm takes into account the different timeliness requirements on data compression. Thus, this algorithm is suitable for both real-time and delay-tolerant transmission. It achieves compression performance up to $74.02\%$ for real-world data sets. Table \ref{table:Entropy} summarizes the performance evaluation of mentioned entropy based approaches.\\

\begin{table}[h]

\centering

\caption{Pefromance Comparison of Entroy based Compression Methods}

**\label{table:Entropy}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|c|}{8} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{9} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{10} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{N} \\ \hline

\multicolumn{1}{|c|}{11} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{12} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

**\paragraph**{ Run-Length}

The simple and one of the most popular data compression algorithms is the Run-Length Encoding (RLE) in which the runs of data, i.e. the sequence of similar data elements in the input data stream (repeating string), are replaced by a single data element value. The RLE plays a vital role where the data stream contains many runs. RLE replaces \textit{n} consecutive times occurrence of \textit{d} data item in the input stream, with the single pair \textit{nd} [2]. This approach is useful when repetition often occurs inside data. \\

However, because RLE is based on the same consecutive input stream, its results depend on the data source. In this way, in order to perform RLE with different data sources statistics, authors in [] introduce a new compression algorithm inspired from RLE and named K-RLE which means RLE with a K-Precision. The K-RLE improves the compression results with different statistics of data sources. It shows increased compression ratios compared to RLE, but with certain amount of error. The performance of this algorithm depends on the choice of the value of the parameter \textit{K} which represents the precision. This algorithm emphasizes on processing the data locally at node level. In this way, if a data item \textit{d} or a data item between \textit{d+K} and \textit{d-K} occurs for \textit{n} consecutive times then the occurrences are replaced by a single pair \textit{nd}. If $K=0$, then K-RLE is RLE. This \textit{K} value makes the K-RLE lossy compression algorithm, leaving RLE a lossless algorithm. The choice of \textit{K} also influences the percentage of data and the extent to which it is modified by this algorithm.\\

The main drawback of K-RLE is that the compression ratios depend on the data sources. The user chooses the \textit{K} value depending on the desired compression ratio. Generally, mathematical parameters like standard deviation and allen deviation are used. K-RLE can achieve higher compression ratios at the cost of data precision when \textit{K} increases. Thus, the value of \textit{K} provides an indication about the data loss resulting from the process. In K-RLE, one can see that compression ratios fall down as the precision requirements are high. This limits K-RLE from adoption by applications that require high precision. Hence, there is an obvious requirement for a better lossy compression algorithm in WSN, which can result in higher compression ratio with minimal or little higher loss depending on the constraints.\\

\begin{table}[h]

\centering

\caption{Pefromance Comparison of Run Lenght based Compression Methods}

**\label{table:RunLenght}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|c|}{8} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{N} \\ \hline

\end{tabular}

\end{table}

**\subsubsection{Distributed Methods}**

In dense sensor networks, measurements from neighboring nodes often exhibit a certain degree of correlation as nodes usually monitor common phenomena. Therefore, transmitting all these information is highly redundant. Spatial compression techniques exploit existed redundancy and spatial correlation in order to save energy by (i) reducing the number of transmissions made to the fusion center, (ii) reducing the amount of each transmission, or (iii) a combination of the two.

In addition to the spatial correlation, the sensing data collected by a single node may also present high temporal correlation when its surrounding remains stable or changes slowly. Most of the spatial correlation based compression techniques are distributed which helps them to utilize densely deployed structure to compress data. Considering distributed nature of these compression techniques, Distributed Source Coding (DSC) is introduced.

**\paragraph**{Distributed Source Coding}

One way of exploiting the spatial correlation between sensor nodes is to use a DSC approach, which is fundamentally based on the Slepian-Wolf theorem [4][5]. We refer to DSC as the compression of multiple correlated sensor readings from neighboring sensor nodes. These nodes do not communicate with each other and directly send their compressed readings to a central node, which performs joint decoding [6][7]. Hence, it reduces computational complexity of the sensor nodes and increases the computational complexity of a usually more powerful central node without sacrificing performance. Due to the severe aforementioned constraints in WSNs, there is a strong need for utilization of energy-efficient processing capability thus making practical DSC techniques one of the most supportive technologies for sensor networks.\\

Since DSC tries to remove the redundancy in the information, it is especially promising when data correlation exists.. Existing results from information theory show that DSC can be executed in a distributed fashion and without any performance degradation in comparison with the centralized approach. There are two types of nodes in distributed source coding for WSNs: (i) source-coding node (ii) data gathering node (remote sink). Source coding node should perform sensing, transmission and encoding. It encodes the quantized version of the sensing data and transmit the encoding data when the data-gathering node sends the query. Data-gathering node which is assumed to be a powerful node, has the responsibility to determine the source-coding rate for each source coding node. After receiving the encoded data from source coding node, it decodes the encoded data to get the prediction of the quantized measurement.\\

Recently, there are more and more researches discuss how to exploit the richness of information provided by spatial correlation of measurements taken from nodes of wireless sensor networks, and further compress the measurements without inter-node communications which is the feature of distributed source coding [3–5]. Unlike the traditional direct compression, distributed source coding in wireless sensor networks is the remote compression problem. The encoder can only access the side information indirectly through a noisy observation process. In WSNs, the noise and interference in the error prone channel may cause the information collected from WSNs distortion or loss. Therefore, some schemes must be implemented to protect the encoding version of quantized measurements.\\

The early work that focused on energy-efficient distributed source coding schemes for wireless sensor networks is [Tang et al. (2007)]. The proposed method is an energy-efficient adaptive distributed source coding (EEADSC) technique for sensor clusters which exhibit a high spatial correlation. This was designed for the specific application of sensor networks that detect a target using acoustic signals, e.g., automatic target recognition (ATR). Find a bit sequence that conveys information about the coset index [] is the building block of this approach, which minimizes (i) the Lagrangrian cost, (ii) the cost function of bitrate, (iii) the distortion and (iv) energy cost used in decoding algorithm and transmission. This technique is practical in cases of single cluster topology. They study the problem of a random-binning based DSC scheme for remote source estimation in WSN. They design a DSC scheme and analyze its performance on the estimated signal to distortion ratio (SDR), in which observation noise, quantization distortion, DSC decoding errors and network packet losses are all taken into account. Simulation results show the proposed adaptive DSC scheme either consumes up to $31.6\%$ less energy without decreasing the SDR or maximizes the SDR with up to $9.4\%$ energy saving.\\

The DSC compression rate is directly dependent on the level of correlation among sources, which is not constant over time. This implies that it is beneficial to apply multi-rate distributed source coding in WSNs in order to enhance power saving. The authors in [Rezayi and Abolhassani (2009) and Wang et al. (2009a)] propose the multi-rate DSC compression schemes. The authors in [Rezayi and Abolhassani (2009)] propose a technique to apply multi-rate distributed source coding using low density parity check (LDPC) codes. This is to reduce energy consumption between source outputs with high correlations and to decrease the bit error rate (BER) value in low correlations rather than using single rate DSC. On the other hand, the energy consumption of their multi-rate scheme was better than a single rate coding at similar maximum BER value. \\

Although there have been several literature works studying on the DSC approach in recent years, the power consideration of those works is often highly abstracted and does not consider practical issues sufﬁciently (Oldewurtel et al., 2010). For example, the power consumption of the processing unit is not taken into account in the analysis. This motivates the authors in Oldewurtel et al. (2010) to study the energy consumption of DSC in various and more realistic topologies together with the cluster head selection schemes of WSNs. Three topological models and three cluster head selection schemes were evaluated. The ﬁrst topological model is random point process (PP) (Stoyan et al., 1995) which is based on Poisson distribution, and a rather unrealistic model was used in this evaluation for reference and comparison purposes. The second topological model called Thomas PP is an extension of the PP model. The authors argued that the Thomas PP model led to a more realistic deployment model in the context of WSNs since sensor nodes were often assumed to be clustered. The ﬁnal topological model is a grid in which sensor nodes are placed on the vertices of a rectangular grid. In their simulation, sensor nodes in these topologies were formed into clusters. The cluster heads were selected by three selection schemes: random selection, closest-to-center of gravity, and closest to sink. To perform DSC, the compressing node is one of the cluster members and the reference node is its cluster head. Using a power consumption model that makes use of measurements obtained from real experiments, the results of simulation in Oldewurtel et al. (2010) pointed out that adopting closest-to-center of gravity scheme on Thomas PP topology strongly outperformed other combinations. It could save power up to $50.7\%$ and the lifetime was extended up to $34.9\%$. Additionally, the work also described optimal parameters that could maximize the energy saving using DSC.\\

Authors in [4] propose a jointly opportunistic source coding and opportunistic routing (OSCOR) protocol for correlated data gathering in wireless sensor networks, which exploits the broadcast nature of wireless transmission. OSCOR broadcasts each packet, which is received by possibly multiple sensor nodes, and opportunistically chooses a receiving neighbor to forward the packet, with the goal of obtaining a path online with highest possible compression and best possible link quality. Opportunistic forwarding with opportunistic compression allows OSCOR to exploit multiuser diversity in packet reception, data compression and path selection, result in high-expected progress in transmission. They propose a practical distributed source coding that combines and takes advantage of both Lempel-Ziv []code and network coding. Lempel-Ziv code does not require the knowledge of the statistics of the data, while network coding is well-suited to a distributed compression of information in networks. They use expected transmission count discounted by node compression ratio (cETX) and expected opportunistic transmission power discounted by node compression ratio (cOETP) along a path as the path metrics for routing. They propose modified Dijkstra's algorithms to update the path metrics cETX and cOETP from a node to the sink and then select the shortest path accordingly. Basically, these two metricsare used to prioritize the neighboring nodes and update the forwarding candidate set of a node. Computational complexity of each sensor node has exponential relation with number of forwarding nodes $(O(nlog2(L))$.\\

In [5], authors propose an approach to improve the DSC decoding quality in terms of loss factor, while still satisfying data delivery latency requirements. Also, a novel coding topology, hierarchical coding topology, has been introduced. With a given network topology and correlation structure of the sensing area, they can easily construct the hierarchical coding topology.. The DSC decoding quality in WSNs is further improved by adjusting the maximum retransmission limit of DSC packets, which is an effective way to protect the DSC packets of different importance. They also applied a particle swarm optimization[] based evolutionary algorithm (APSO) to solve the optimal DSC packets transmission-scheduling problem with regard to a practical wireless environment. Simulation shows that their proposed hierarchical coding topology performs efficient in both high and low data correlated environments. Simulation results also reveal that the APSO eases the decreasing of decoding quality caused by packet loss.\\

Table \ref{table:DSC} summarize and compares the performance of mentioned DSC methods.

\begin{table}[h]

\centering

\caption{Performance Comparsion of Distributed Source Coding Methods}

**\label{table:DSC}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net of Saving \\ \hline

\multicolumn{1}{|c|}{1} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{2} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{3} & \multicolumn{1}{c|}{N/A} & \multicolumn{1}{c|}{N/A} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{N} \\ \hline

\multicolumn{1}{|c|}{4} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{5} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

**\subsection{Lossy Data Compression Techniques}**

In information technology, "lossy" compression is a data encoding method that compresses data by discarding (losing) some of it. The procedure aims to minimize the amount of data that needs to be held, handled, and/or transmitted by a sensor node. In lossy compression, the encoder often transforms the data into a new domain using appropriate basis functions. For example, the data can be transformed into the frequency domain using Fourier basis functions. In the new domain, the information content is concentrated in a small number of coefficients that contain the data values projected using the basis functions. Therefore, the encoder can reduce the data size by selecting only those coefficients.\\

In this section, we classify lossy compression techniques as figure, and describe the characteristics of each class of compression methods.

\begin{figure}[h]

\centering

**\includegraphics**[width=1\textwidth]{Lossy}

\caption{Lossy Data Compression Techniques}

**\label{fig:LosslessMethods}**

\end{figure}

**\subsubsection{Local}**

**\paragraph**{Entropy based}

Differential pulse-code modulation (DPCM) scheme widely used for compressing signals, especially in speech and video coding [31] and is a member of the family of differential compression methods. DPCM is a signal encoder that uses the baseline of pulse-code modulation (PCM) but adds some functionality based on the prediction of the samples of the signal. Basically, this method exploits the high correlation that typically exists between neighboring samples of smooth digitized signals, achieving compression by appropriately encoding differences between these samples. In the context of WSNs, different DPCM-based methods have already been employed. For instance, in [32], the authors propose an algorithm based on DPCM to compress data collected by vibration sensors and discuss the effects of signal distortion due to lossy data compression on structural system identiﬁcation. In their scheme, they use the least squares method to derive the linear predictor coefﬁcients, a Jayant quantizer [] for scalar quantization and an arithmetic coding as entropy encoder.\\

Some studies [33,34,35] show how DPCM techniques can be used to enable audio signal compression over WSNs. In particular, in [33], the authors show a very interesting experiment on how to implement a networking platform for supporting real-time voice streaming over a WSN in a coal mine. In [34], the authors describe how to implement streaming services for supporting military surveillance applications. The need to transmit understandable speech over WSNs along with energy consumption constraints are important in [33][34]. To address these two issues microphone sample rates are set lower than the normal sample rates and an adaptive DPCM is used to encode the data and thus reduce the transmission data rate. In [35], a distributed adaptive DPCM scheme was proposed in order to solve the problem of the low sample rates which affected the quality of the speech at the receiver in [33] and [34].\\

In [36], the authors introduce a two-stage distributed DPCM coding scheme for WSNs, consisting of temporal and spatial stages that compress data by making predictions based on samples from the past. The interesting feature of this approach is that, since it continuously monitors the additional gain provided by samples collected from other sensors, it can be combined with data-centric routing algorithms for joint compression/routing optimization. The novelty introduced in this paper is not the use of a DPCM compression scheme in WSNs, but rather an optimization method which allows using a classical DPCM scheme for reducing the information entropy at the encoder, resulting in a reduced noise after reconstruction at the decoder.\\

Authors in [37] propose an approach to perform lossy compression on single node based on a differential pulse code modulation scheme with quantization of the differences between consecutive samples. The quantization process affects both the compression rate and the information loss. To generate different combinations of the quantization process parameters corresponding to different optimal tradeoffs between compression performance and information loss, they have applied NSGA–II[], a popular multi-objective evolutionary algorithm, on a subset of samples collected by the sensor. The user can therefore choose the combination with the most suitable trade-off for the speciﬁc application. They tested their lossy compression approach on three datasets collected by real WSNs, obtaining high compression rates at very high signal-to-noise ratios. Like as other compression categories, we summarize the characteristics of discussed approaches in table.

\begin{table}[h]

\centering

\caption{Performance Comparison of Lossy Entropy based Compression Methods}

**\label{table:LossyEntropy}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|l|}{35} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{37} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

**\paragraph**{Run-Length}

The LTC algorithm proposed in [38] is an efficient and simple lossy compression technique for the context of habitat monitoring. The algorithm exploits the fact that the captured readings for microclimate data, in a small window of time, are linear in nature. It identifies such windows and generates a series of line segments that accurately represent the data. This scheme performs compression by introducing error bounded by a control knob, which is in the order of the error specified on the hardware. This algorithm attempts to represent a long sequence of similar data with a single symbol. It is effective on a data set, which is largely continuous, and changes in readings are infrequent. Thus, the results of LTC show that it performs better on the data related to temperature than on humidity or wind speed. This shows that the compression ratio in LTC is highly dependent on the nature of the data. The LTC algorithm is designed for mica motes with 8-bit processor, which has no hardware to handle floating-point values. This limits the applications of LTC to compression of integer data only.\\

Basically LTC is similar to RLE in the sense that it attempts to represent a long sequence of similar data with a single symbol. The difference with RLE is that while RLE searches for strings of a repeated symbol, LTC searches for linear trends. LTC scheme is designed by considering two key points: (i) the observations of microclimate data over a small enough window of time are linear (ii) the noise exists in sensors. Based on these observations, LTC was designed for climate monitoring applications by ﬁtting microclimate data during a short range of time (window) with a sub-linear model. LTC was also designed to compress data when sensor accuracy was expressed as an error margin and when the probability distribution of error was either uniform or unknown.\\

In [39], a new lossy data compression algorithm called IR-LDCA is proposed to achieve better compression ratios at minimal loss. IR-LDCA for WSNs is proposed to achieve energy efficiency in transmission. The proposed algorithm effectively exploits the natural correlation that exists in sensory data. The objective of the work is to exploit the commonality existing in the continuous data stream and also to eliminate redundancy. This property is exploited through inducing redundancy in the data set and converting them to their integer representation. This lossy data compression scheme allows the user to control the compression ratio and the data loss during the compression, facilitating higher compression ratios with minimum loss of data. Also, the algorithm performs in a more flexible and optimized manner than the existing K-RLE algorithm with respect to the RMS error (data loss). This proposed scheme reduces the volume of data for transmission at a sensor node (with minimal loss). Thus, resulting in reduced energy consumption and enhanced lifetime of the network.\\ Table represents a summery of discussed techniques.

\begin{table}[h]

\centering

\caption{Pefromance Comparison of Lossy RunLenght based Compression Methods}

**\label{table:LossyRunlenghtl}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|l|}{38} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{N} \\ \hline

\multicolumn{1}{|c|}{39} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

**\subsubsection{Distributed Methods}**

Distributed lossy compression techniques are classified in three categories: (i) Distributed Source Modeling (DSM) (ii) Distributed Transform Coding (DTC) (iii) Compressive Sensing/Distributed Compressive Sensing (CS/DCS).

**\paragraph**{Distributed Source Modeling (DSM)}

The DSM technique aims to search for a function/model that best ﬁts a set of input measurements acquired by a speciﬁc group of sensor nodes using parametric modeling and non-parametric modeling. By the means of parametric modeling, an algorithm treats sensor data as a random process that has to be optimally estimated when knowing its statistical parameters, such as mean and variance. Parametric modeling could yield superior performance when the statistical structure of the random process (sensor data) being observed is known. On the other hand, non-parametric modeling utilizes kernel-based regression [] to represent the sensor data where the regression coefﬁcients are learnt by treating the sensor data as input–output example pairs of some deterministic function observed in noise. In this case, it requires very little prior information about the nature of the data in the random process and is considered to be very robust (Oka and Lampe, 2008). We have already described this model in Adaptive Sampling chapter. Therefor we will not describe this model in this chapter.

**\paragraph**{Distributed Transform Coding (DTC)}

The transform coding technique decomposes a source output based on transform theories into components/coefficients that are then coded according to their individual characteristics. DTC includes several well-known approaches, such as Karhunen–Loeve [] transform, Cosine transform [] and Wavelets transform []. In non-WSN applications which do not suffer from power constraints, DTC-based compression approaches are widely used, especially in image, video and audio compression algorithms.

Transform coding is used to convert spatial image pixel values to transform coefficient values. Since this is a linear process and no information is lost, the number of coefficients produced is equal to the number of pixels transformed. The desired effect is that most of the energy in the image will be contained in a few large transform coefficients. If it is generally the same few coefficients that contain most of the energy in most images, then the coefficients may be further coded by lossless entropy coding. In addition, it is likely that the smaller coefficients can be coarsely quantized or deleted ( lossy coding ) without doing visible damage to the reproduced image.\\

Not many types of transforms have been tried for image coding, including for example Fourier [], Karhonen-Loeve [], Walsh-Hadamard [], Lapped orthogonal [], Discrete Cosine (DCT) [], and recently, Wavelets. The various transforms differ in three basic ways that are of interest in image coding:\\

1) the degree of concentration of energy in a few coefficients;\\

2) the region of influence of each coefficient in the reconstructed image;\\

3) the appearance and visibility of coding noise due to coarse quantization of the coefficients.\\

Karhunen-Loeve[] is a statistically based transform method that can be tailored to one image or group of images, and therefore has the optimum energy concentration. However, it generally will not have this optimum concentration for images not in the basis set.

Fourier transforms (discrete) have good energy concentration characteristics, but becomes inefficient when dealing with large images requiring large numbers of coefficients. Block transforms, which work on a small portion of the image at a time, are therefore preferred. The discrete Fourier transform may be applied to a block of pixels. Other transforms, which fall in this category, are Walsh-Hadamard, and the DCT. The lapped orthogonal transforms are a special case in which the coefficients' influence is confined to a few adjacent blocks, with a tapering-off influence toward the edges.\\

Because of ease of hardware computation and generally very good energy concentration for a wide range of natural images, the DCT has become the transform of choice for many image-coding schemes, including MPEG. The DCT, unlike the Fourier transform, is spatially variant. A portion of a sine wave coded with a Fourier transform has all the energy concentrated at the same frequency coefficients regardless of the phase of the sinusoid (although the energy will be apportioned differently between the sine and cosine components). The DCT, on the other hand, is sensitive to phase, so that an object moving across the screen will have different frequency content from frame to frame. This also means that the visibility of coding artifacts due to coefficient quantization will vary somewhat depending on the position of an object (edge) in the image. Also, because the DCT is a strictly bounded block transform, lossy coding will produce block-edge mismatch, which will be visible at some level of quantization even if there is only low frequency content in that area.\\

An image is usually composed of many small pixels and a matrix whose elements are the values of these pixels. By conducting some wavelet transformation over the aforementioned matrix, we can extract the important features from the image in the frequency domain [16]. Then, storing only these important features of the image can significantly reduce the image size.

Generally, it is difﬁcult to implement a full version of those popular transform-based compression algorithms in WSNs. The main reason is that they often require one of the sensor nodes to have knowledge of all measurements in a network in order to calculate the transform coefﬁcients. This requirement could potentially increase the volume of inter-node communication which affects communication costs and results in higher power usage. Due to this reason, there are several works in the literature[][][] that aim to approximate or modify those classic transform-based algorithms in order to be applicable to WSNs. \\

In transform coding, the compression scheme acts on a linearly transformed version of the data. Specifically, for the sampled data \textit{f}, we typically use an orthonormal transformation matrix \textit{J} that yields a coefficient vector $w=J^{-1}f $(so that $f=Jw$) and achieves compression by encoding \textit{w}, dividing the bit budget among the entries of \textit{w} according to their impact in signal distortion. For example, the bit budget may be partitioned among the entries of\textit{w} according to their expected magnitudes. In certain cases, it might be optimal to discard small coefficients when the savings in bitrate outweigh the distortion incurred by ignoring them. The signal is then decompressed by decoding the bit stream and populating an approximate coefficient vector $w$ with only the preserved coefficients (setting all others to zero), and then inverse transforming to obtain the estimate $f=Jw$. The amount of distortion introduced by this type of compression is owing to the quantization error during coding and the magnitude of the discarded coefficients; thus, efficient transform coding relies on a suitable transform and coding scheme pair that minimizes these two sources of distortion.\\

We classify these approaches based on the transform coding metrics as follows:

**\subparagraph**{Karhunen–Loève Transform (KLT)}

KLT is an applicable transform coding to approximate and compress random variables, which minimizes the total mean square error. When the entire variables are available, it can reconstruct un correlated random variables. However, in case of sensor networks which each sensor observes only part of the signal, KLT must be applied in distributed format which results in distributed KLT.

(i) Distributed Karhunen–Loève Transform

In case of such WSNs which require to restrict their communication to only local transmissions, the KLT coefficients should depend only on sensor readings from communication graph neighbourhoods. In case of sensor networks, multiple sensor nodes sample only a part of the original signal X and transmit their observation to the base station. Since sensor nodes can only communicate with neighbor nodes, to reconstruct the original signal, therefore, a distributed KLT approach suggested in []. In this approach, each sensor node utilizes its measurement matrix to produce a linear encoded data and transfer it to the base station. The size of encoded data is defined based on setting an upper and lower bound for mean square error parameter. For the upper bound sensor nodes utilizes their own data to compute a KLT. However, in case of lower bound, sensor node employs correlation among different sensor nodes measurements to solve joint KLT. For the joint KLT, base station runs an iterative reconstruction procedure. In each iteration, base station considers one of the sensors and assumes that the encoding matrices of all the other sensors are fixed and known to that sensor. Then it finds the optimal encoding matrix for that sensor in a way that minimizes the mean square error. The DKLT also follows the routing-over-compression design scheme, since the communication network neighbourhoods determine the constraints of the DKLT compression basis \textit{J}. \\

ؤA greedy based recursive DKLT algorithm proposed in [22] which terminates reconstruction in a finite number of steps. Against previous approach, this technique assumes signal is noisy. In each iteration, base station updates the encoding matrix of a sensor node which represents maximum reduction in the mean square error. When encoding matrix of all sensor nodes reach their predefined row dimensions, this algorithm terminates. Compared with [], it represents same accuracy in terms of mean square error, however, its complexity is less.

. \\

\begin{table}[h]

\centering

\caption{Performance Comparison of DKLT Compression Methods}

**\label{table:DKLT}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|c|}{21} & \multicolumn{1}{c|}{N/A} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{N} \\ \hline

\multicolumn{1}{|l|}{22} & \multicolumn{1}{c|}{N/A} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

(ii)Tree Karhunen–Loève Transform

[] suggests a tree KLT as a distributed unidirectional transform coding which employ spatial model to achieve maximum data de-correlation. TKLT is basically a routing scheme over compression, which has minimum learning cost. For transmitting data, each leaf node forwards its data to the parent node. Parent nodes only observe its data and its descendant nodes; therefore, it applies TKLT on its data and data received from descendant nodes and produces KLT coefficients. This procedure repeats along the routing tree till base station receives data. It is necessary for each sensor node to forward its KLT encoding matrix to the parent node or has access to the statistical correlation structure of nodes. To do so, each senor nodes finds two types of covariance matrices: (i) covariance matrices of its sub tree and (ii) the covariance matrices of its children nodes. Since thie approach is unidirectional, base station is not allowed to have any backward communications with sensor nodes. This method can serve mostly as an upper bound method. Table compares the performance of aforementioned techniques. \\

\begin{table}[h]

\centering

\caption{Performance Comparison of TKLT Compression Methods}

**\label{table:TKLT}**

\begin{tabular}{lcccc}

\hline

\multicolumn{1}{c}{} & CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|l|}{22} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{N/A} & \multicolumn{1}{c|}{N/A} \\ \hline

\end{tabular}

\end{table}

**\paragraph**{Wavelet Transform}

Many classes of signals have been shown to exhibit sparsity in particular bases. For example, the Fourier basis provides sparse representations for smooth signals, while wavelet bases provide sparse representations for piecewise smooth signals. In this subsection, we focus on the case of wavelet transforms, which are natural for (i) WSN settings such as anomaly detection in smooth signal fields, (ii) providing compact representations for smooth fields with few discontinuities. Different class of wavelet transform are introduced and discussed. However, Irregular sampled wavelet and graph-based wavelets are two categories, which are discussed in this section.

(i) Irregularly sampled wavelets

The irregular wavelet transform [] is well suited for settings in which the communication neighborhoods are small, and multiple hops are involved in the paths to the sink, which increases the communication savings owing to compression. In this approach filters used for compression are chosen according to the geographical neighborhoods, which are not necessarily dependent on the network topology of the WSN. It means that the method can be placed in the compression-over routing or separate design classes. The irregular wavelet transform requires communication in local neighborhoods to compute scaling and wavelet coefficients. It has been shown experimentally that this transform provides a more balanced distribution of the communication expense needed to receive all sensed values, when compared with a data dump approach that stresses the communication bottleneck around the sink. This approach falls in the spatial-only class and do not support temporal cases.\\

In order to apply wavelet transform (WT) in sensor networks, earlier works assume gird topology for this networks. Lifting Scheme based Wavelet Transform (LSWT)[2005] is one of these methods, which combine WT with routing to reconstruct sensor node readings in a distributed fashion. Each sensor node computes its partial coefficients of WT and forwards it to the next sensor node in the forward path towards base station. This procedure repeats by sensor nodes along the forward path till full coefficients of WT computes in base station. Another version of LSWT is suggested in [] for random sensor nodes deployments where WT is implemented in a minimum spanning tree-based path. However, the approach is not scalable and can not capture the higher dimension of spatial correlation among measurements. This work considers the tradeoff between transmission and computation costs which led to energy efficient data gathering solution.

A distributed wavelet base adaptive compression method is proposed in [] which changes its data gathering strategy according to the properties of the signal field.

Irregular WT is proposed in [Acimovic et al. [2005]] develops piecewise-constant multiscale approximation based transform coding which combines this transform with multi scale routing. Authors address a new version of this techniques in[Wagner et al. [2005a]] and introduce a WT based protocol which can perform piecewise planar multiscale approximation.

A robust information driven data compression architecture (RIDA) is introduced in []. This architecture is consist of three main modules: logical mapping, compression algorithms , and a resiliency mechanism . In logical mapping module, sensor nodes are reorganized and grouped in the clusters according to their spatial correlation which helps to achieve higher compression ratio. Then in compression module, two different transformation apply on the sensor data: DCT and DWT. Sensor nodes inside each cluster exchange their data with its neighbor nodes and then utilize one of transformation method to calculate its coefficients and then in quantization module, each sensor weighs

its coefficient using a quantization matrix. In order to transfer coefficients to the base station, sensor node employs a threshold to select which coefficients must be transfer to the base station. RIDA is suitable for sensor networks that are dense, have a cluster topology, and have slowly changing environmental conditions. Their experiments show that using RIDA can double the compression performance of a state-of-the-art distributed data compression method, Wagner- distributed wavelet transform , and eliminate 80\% of the energy consumption compared to a sensor network not using RIDA. In addition, RIDA can perform well even when half the sensor data is missing.\\

Table \ref{table:IWT} compare proposed methods performance.

\begin{table}[h]

\centering

\caption{Performance Comparsion of Irregualr WT}

**\label{table:IWT}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|l|}{23,24} & \multicolumn{1}{c|}{Depends on SNR} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{25,26} & \multicolumn{1}{c|}{Depends to SNR} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{27} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

(ii) Graph-based wavelets

While the irregular wavelet transform makes use of node proximity regardless of network topology (compression over routing), graph-based wavelets [28-30] use a transform constructed on the network communications graph (routing over compression). The partitioning step chooses odd nodes that provide maximal de-correlation. A lifting transform can be inefficient in terms of the overall number of communications, especially if even nodes must first transmit raw data to odd neighbors and then wait for odd neighbors to transmit transform coefficients back to them before they can compute their own transform coefficients and forward them to the sink. This may produces significant communication overhead, since many nodes are transmitting data twice, and data may be transmitted away from the sink. Graph-based wavelets strive for low communications cost by requiring nodes to transmit data just once and to do so in the direction of the sink. In fact, it is possible to transmit data only once by computing the transform as data is forwarded to the sink along a given routing tree. Such transforms have unidirectional operation [28,29], since each node only transforms its own data using data from neighbors along a one-dimensional routing path to the sink. This is achieved by constructing the transforms along an arbitrary routing tree while also utilizing data received via broadcast from neighbors not in the tree. It is possible to optimize the even–odd splitting under uni-directionality constraints. An even–odd splitting can be viewed as a constrained set-covering problem, where every odd node needs to be covered by at least one even node and the communications are constrained by the transmission ranges used in the routing tree; i.e. a node can only cover neighbors whose distance is less than the distance from itself to its parent in the routing tree. The heuristic-constrained minimum set covering algorithm of Narang et al. [31] has been shown to provide lower energy consumption than other graph-based wavelet methods for a fixed reconstruction quality.\\

The lifting scheme is an alternative method to compute biorthogonal wavelets. It allows a faster implementation of the wavelet transform along with a full in-place calculation of the coefﬁcients [23,24]. The scheme consists of three steps: (a) split, (b) prediction and (c) update. In the split step, the signals s(n) are split into even signals and odd signals. Both types of signals are then processed in prediction and update steps. Finally, the detail coefﬁcient d(n) and smooth coefﬁcients y(n) are obtained. They use distributed wavelet transform. A sensor in the routing path partially calculates the transformed coefficients and forwards the coefficients to the next sensor in the routing path. The sensor at the end of the routing path will receive all the coefficients of the discrete wavelet transform of sensors along the routing path. Ciancio’s algorithm performs a wavelet transform on the raw data along a path. However, it does not explore methods which improve the compression radio obtained with wavelet transform on the same set of data. Since this scheme depends on the routing tree, temporal and spatial correlations between sensor data are not explicitly exploited..\\

In [25,26] a baseline wavelet lifting approach is designed for signals f that have been sampled on a uniform grid. They extend the lifting scheme to 2-D wavelets transform, which is then applied with the routing tree. Additionally, the prediction and update filter designs assume that the cost of computing a wavelet coefficient depends only on the number of samples involved. This set-up does not map very well onto a WSN, since a WSN sampling is very often performed irregularly in the space domain according to the particular deployment of the sensors in the field. Nonetheless, it is possible to adapt wavelet transforms designed for regularly sampled data to irregular sampling [25]. In this case, the definition of the neighborhoods must be modified. This approach uses location information to define local neighborhoods of nodes at multiple scales that exchange information, forming multi-resolution geographical neighborhoods. This also forces the neighborhoods to be calculated in sequence from finest (smallest) to coarsest (largest) scales. For a local neighborhood at a particular scale, the partition stage arbitrarily selects a node to be placed in the even subset and moves all nodes within a smaller geographical neighborhood of the new even node to the odd subset. This process is repeated with the nodes remaining in the neighborhood until each node has been labeled as even or odd. The prediction and update filters are designed to perform optimal approximation of the corresponding sampled value by fitting a plane to the readings from sensors in the even/odd neighborhood partitions, correspondingly [25].\\

\begin{table}[h]

\centering

\caption{Performance Comparison of Graph WT methods}

**\label{table:GWT}**

\begin{tabular}{ccccc}

\hline

& CR & Cmm\\_Cost & Comp\\_Cost & Net\\_Saving \\ \hline

\multicolumn{1}{|l|}{29} & \multicolumn{1}{c|}{Depends on SNR} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{30} & \multicolumn{1}{c|}{Depends on SNR} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Y} \\ \hline

\end{tabular}

\end{table}

**\paragraph**{Compressive Sensing/Distributed Compressive Sensing}

Existing compression techniques suffer from some essential disadvantages, which make them inefficient. Sampling and processing high-dimensional signals are inappropriate for resource restricted sensor nodes. Upon receiving such a high dimensional signal parameters, encoder discards most of the samples and only transfers most informative ones, which causes wasting energy and processing resources. In other hand, encoder is required to send the location of most informative signals’ coefficients as well as amplitudes. There is an emerging compression technique called compressive sensing which is able to compress and reconstruct sparse or compressible signal from small number of measurements without requiring any apriori knowledge about the signal structure.

Against existing traditional compression techniques, compressive sensing based approaches sample signal with sub-Shannon-Nyquist rate and recover this signal with high accuracy if certain condition are satisfied. In fact compressive sensing utilizes information rate instead of sampling rate to sample and recover the signal.

compressive sensing is advantageous whenever signals are sparse in a known basis, measurements (computation at the sensor end) are expensive and computations at the receiver end are cheap [3]. These characteristics completely match wireless sensor networks, which have asymmetric computational nature. Temporal, spatial and spatio-temporal correlation among sensor nodes readings satisfies sparsity or compressibility requirements of compressive sensing . A complicated decoding task takes place in sink or central node while sensor nodes run a simple encoding procedure. In addition, it minimizes the abnormal sensor reading as well as sensitivity to packet loss. According to the type of correlation which compressive sensing techniques utilize, we classify them to three category: Temporal, Spatial and Spatio temporal.

**\subparagraph**{Temporal Compressive Sensing}

Haupt et al. discuss how to makes encoding process more efficient in term of resource management strategy. Proposed approach considers temporal correlation among sensor node readings to recover the original signal. They combine compressive sensing with routing method to recover a sparse signal in a multi-hop network. Gossip based routing protocol helps data gathering approach to use the benefits of compressive sensing in storing and recovering data form multiples points of network instead of BS. Each sensor calculates measurement and transfer it to the subset of sensor nodes using gossip based routing. \\

TC-CSBP [20] is a belief propagation based compressive sensing technique, which employ temporal correlation among sensor node readings to reconstruct the signal. This approach only focused on temporal correlation based reconstruction algorithm and dose not mentioned energy efficiency of the approach. This reconstruction method builds upon compressive sensing via belief propagation (CSBP) [] technique. They employ the time-correlation model of signal as a priori information and improve reconstruction accuracy. Further, an integration of online model estimation into TC-CSBP was studied for more accurate model estimation.\\

Authors in[] address three different temporal correlation based compressive sensing techniques. The first approaches employ temporal correlation to have more sparsity level which results in less number of measurements. Other approach emphasizes on using temporal correlation as prior information to reconstruct the original signal. The last approach makes multiple measurement vectors by combining several single time measurement vectors to achieve sparser representation. They evaluate the performance of their methods at different noise levels.\\

[23] suggest a compressive sensing based data gathering method to monitor ECG signals in Wireless body area networks. They proved that existing transform based compression techniques are too complex and time consuming solutions. Proposed technique employs binary sparse sensing matrix and achieve low complexity and energy efficiency. Compared with discrete wavelet transform based compression algorithm, it represents a very low complexity and CPU execution time which leads to 37% extension in node lifetime.\\

Proposed adaptive compressive sensing techniques in [22,44] utilizes a feedback control mechanism to adjust sensor nodes sampling rate. They employ a random sampling measurement matrix and attempts to make a tradeoff between their randomization procedure and system performance. The parameters of feedback control mechanism address the expected scheduling of sampling times. Sensor nodes do not require any prior information of monitored signal and can adopt their sampling rate to the time-varying sparsity level of signals.

**\subparagraph**{Spatial}

In [], a compressive wireless sensing (CWS) approach is introduced which utilizes an ensamples of spatially distributed sensor nodes to recover the target signal. This approach is a decentralized approach which transfer sensor node readings in a distributed and energy efficient manner to the base station. In CWS data processing and communication are combined into one distributed operation without any in-network communication and processing requirements. In addition, consistent filed signal reconstruction without any prior information of the sensed data makes CWS a universal approach. However, this universality increases the cost of optimization in terms of a less favorable power-distortion-latency trade-off, which is a direct consequence of not having sufficient prior knowledge of the sensed data. Difficulties in synchronization are the main disadvantage of this technique that make it in scalable. \\

Luo et al[] claim that their compressive data gathering method (CDG) converts the traditional compress and transmit process to the compress with gathering process. CDG utilizes routing in conjunction with compressive sensing to collect data. Suggested mechanism mitigates data traffic and provides energy balance in order to prolong network lifetime. However, this method is applicable in scenarios which we are dealing with large scale networks with static and stable routing architecture. Control overheads and lower sparstity level overcomes compression advantages in case of (a) frequent node failures, (b) dynamic route allocations and (c) low scale networks.\\

In [] a joint sparse signal recovery method is suggested which utilizes spatial correlation to make a tradeoff between energy consumption and the accuracy of the data recovery. Considering optimized clustering technique, sensors have been distributed in different clusters and a localized data projection techniques is utilized to gather data within each cluster, which results in, minimizing data transmission costs. In addition, utilized joint sparse signal recovery instead of independent reconstruction increases the reconstruction accuracy. \\

A compressive sensing based event detection approach introduced in []. This work relies on this assumption that in large scale networks, number of sensors monitoring the area are much more than number of events, which shows the sparsity of events. Inspired by this fact, the number of active sensors monitor the events are minimized to the number of sparse events which is much less than total number of sensor nodes. To recover the original signal, a Bayesian reconstruction algorithm is proposed. This reconstruction method improves network performance in terms of data accuracy and energy efficiency where it significantly reduces the number of samples and provides higher level of event detection probability. To achieve such a high detection probability, marginal likelihood maximization algorithm and a heuristic algorithm for the Bayesian framework is utilized.\\

In spite of existing joint sparsity based signal recovery methods, distributed compressive sensing (DCS) based approach addressed in [39] utilizes different basis functions for common and innovate components. This method uses an efficient projection approach to provide an optimized measurement matrix. Orthogonal Matching Pursuit (OMP) based joint recovery method provides significant recovery accuracy rather than independent recovery case.\\

A random access compressive sensing technique is proposed in [24] for underwater sensor networks. Combining random access and compressive sensing concepts results in an energy efficient and simple algorithm. For measurement, sensor nodes utilize compressive sensing based random sampling procedure. In order to transfer measurements to the center, sensor nodes use random access policy to have access to the channel. As in any random access, it is more likely some of sensor nodes measurements collide at the center. Being aware of data collision, base station discards the collide packets and employ compressive sensing based signal recovery algorithm to reconstruct the original data. However, to satisfy the quality of reconstruction, sufficient measurement probability is introduced which defines the minimum number of active sensor nodes.

**\subparagraph**{Spatio-Temporal Correlation based Compressive Sensing}

Against existing DCS methods which exploits spatial correlation to recover the original signal, Baron et al [] address a new DCS technique for multiple signal ensembles which is based on spatio-temporal correlation. To evaluate the performance of proposed method, three different types of CS named single signal, joint sparse based and distributed method is considered. For each of these methods, they provide three examples to study the performance of joint sparse signal recovery and compare with a real developed joint sparse signal recovery approach. Results prove that for two of three methods, the prediction accuracy is completely similar to the result of developed algorithm.\\

An energy efficient distributed data storage is addressed in [] which is based on compressed sensing theory and network coding technology. Authors introduce Compressed Network Coding based Distributed Data Storage (CNCDS) scheme which employs the correlation among sensor node readings. This method achieves energy efficiency through minimizing number of data transmission and data reception. Theoretical analysis is utilized to prove that CNCDS can provide acceptable performance in compressive sensing based data recovery procedure. In addition, simulation results show improvement in the number of data transmission, data reception and compressive sensing recovery up to $55\%, 74\%, and 76\%$ respectively.\\

A DCS based data transmission scheme addressed in [] which improves the tradeoff among energy usage, reconstruction error and resource utilization. This method, which is called as amplify-and-forward compressed sensing (AF-CS), exploits both temporal and spatial correlations. In terms of temporal correlation, it produces a sparse signal vector which is consist of transmitted signals of all sensor nodes. Different sensor nodes employ Conditional Downsampling Encoder (CDE) - Predictive Decoder (PD) pair [10] to produce sparse representation of sensor nodes readings. In addition, a two hop communication method is used to transfer the readings of active nodes to the base station through a subset of relaying nodes. Thanks to this way of configuration, they achieve R projections by using a single channel lead to cope the multi-user interference in its first hop transmission. To select the number of active nodes and relay nodes, a cost function is defined which controls the tradeoff between reconstruction error and energy consumption. Simulation results show that the AF-CS outperforms other techniques in terms of distortion and number of transmissions, providing simultaneously, energy savings and a significant reduction in the number of channel uses.\\

[] proposes a hybrid compressive sensing method which utilizes sensor nodes locations to design a cluster based data gathering method. Network is consist of several clusters in which sensor nodes are distributed. These nodes transfer their data without any compression to the cluster-heads. Cluster-heads utilize compressive sensing approach to project these readings and transfer them to the base station. Simulation results show that the proposed approach significantly reduces number of data transmissions. Considering the relationship between the cluster size and number of data transmissions in the hybrid compressive sensing method, this technique discuss on finding the optimal size of clusters that can lead to minimum number of transmissions. In addition, simulation results shows when the number of measurements is 10 times of the number of nodes in the network, their method can reduce the number of transmissions by about 60 percent compared with clustering method without using compressive sensing.\\

Unlike other approaches, [21] develops a non-uniform compressive sensing approach for data gathering from . Non-uniform compressive sensing addresses the challenge of heterogeneous sensor node sample scheduling and signal reconstruction. In this way, a joint sparse signal recovery method is proposed which exploits spatial-temporal correlation as well as heterogeneity to recover the original signal. Compared with traditional compressive sensing methods, simulation results show this technique holds same reconstruction accuracy level with significantly less number of samples. However, it has very little communication overhead.

**\subsection{ Summerizing Compression Techniques}**

In this section we summarize and provide a table about most of compression techniques which are introduced in state of the art section. Table \ref{Compression} represent different features of thses compression methods which ae already addressed in the text. Since studied approaches has their own limitation and implementation enviroment, we can not compare all these approach in a vey uniform way. However there is survey papers which tried to addressed numeriacl comparisiion for these techniques [][].\\

Parameteres which are utilized to evaluate the performance of compression techniques are defined as follows:

\begin{table}[h]

\caption{Compression Techniques}

**\label{Compression}**

\resizebox{\textwidth}{!}{%

\begin{tabular}{ccccccc}

\hline

& DSC & DTC & CS/DCS & Lossy Entropy & Lossless Entropy & Lossy Run-Lenght \\ \hline

\multicolumn{1}{|c|}{\begin{tabular}[c]{@{}c@{}}Compression\\ Technique\end{tabular}} & \multicolumn{1}{c|}{DC,CC} & \multicolumn{1}{c|}{DC,CC} & \multicolumn{1}{c|}{SC,CC,DC} & \multicolumn{1}{c|}{-} & \multicolumn{1}{c|}{-} & \multicolumn{1}{c|}{-} \\ \hline

\multicolumn{1}{|c|}{Correlation Type} & \multicolumn{1}{c|}{S} & \multicolumn{1}{c|}{ST} & \multicolumn{1}{c|}{ST} & \multicolumn{1}{c|}{ST} & \multicolumn{1}{c|}{T} & \multicolumn{1}{c|}{ST} \\ \hline

\multicolumn{1}{|c|}{Complexity} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{\begin{tabular}[c]{@{}c@{}}H(Decoging) \\ L(Encoding)\end{tabular}} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Med} & \multicolumn{1}{c|}{M} \\ \hline

\multicolumn{1}{|c|}{Reliability} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{N} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{Robustness} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{L} \\ \hline

\multicolumn{1}{|c|}{Scalability} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{H} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{-} \\ \hline

\multicolumn{1}{|c|}{Compression Rate} & \multicolumn{1}{c|}{L} & \multicolumn{1}{c|}{Depends on SNR} & \multicolumn{1}{c|}{Depends to the Signal Sparsity} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} & \multicolumn{1}{c|}{M} \\ \hline

\multicolumn{1}{|c|}{Communication Saving} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{Single/Multiple Data Types} & \multicolumn{1}{c|}{Si} & \multicolumn{1}{c|}{Si} & \multicolumn{1}{c|}{Si/Mu} & \multicolumn{1}{c|}{Si} & \multicolumn{1}{c|}{Si/MU} & \multicolumn{1}{c|}{Si} \\ \hline

\multicolumn{1}{|c|}{Net of Power Saving} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} & \multicolumn{1}{c|}{Y} \\ \hline

\multicolumn{1}{|c|}{Enviroment} & \multicolumn{1}{c|}{St} & \multicolumn{1}{c|}{St,Dy} & \multicolumn{1}{c|}{St,Dy} & \multicolumn{1}{c|}{-} & \multicolumn{1}{c|}{-} & \multicolumn{1}{c|}{-} \\ \hline

\multicolumn{1}{|c|}{Limitation} & \multicolumn{1}{c|}{Only Star Topology} & \multicolumn{1}{c|}{Inter Node Communications, Topology Dependent} & \multicolumn{1}{c|}{Transform basis Dependent} & \multicolumn{1}{c|}{-} & \multicolumn{1}{c|}{-} & \multicolumn{1}{c|}{-} \\ \hline

\end{tabular}

}

\end{table}

**\section{Why Compressive Sensing}**

Compressive sensing is an energy efficient data gathering solution, which surprisingly can reduce energy cost in both computation and transmission units. However, most of the existing compression techniques reduce transmission energy in cost of increased computation energy.\\

Minimizing energy consumption is a primary challenge in resource limited WSNs which is addressed in different solutions. Data compression techniques are one of the suggested solutions, which compress and transmit less information at the cost of more data processing. Radio transmission consumes much more energy rather than data processing. However, data compression is an energy efficient solution as long as execution time of processing algorithm is short. Sampling interval, sampling frequency and energy hungry based sensor applications are another energy consumption parameters, which are ignored in many scenarios. Nevertheless, there are applications wherein sampling frequency is very high, they utilizes energy hungry sensors like, as CO2 or the sampling interval is long, so the power consumption for sampling is compatible with transmission power. Long sampling intervals are another concern in energy consumption challenges, which can be handled through scheduling sensor nodes to sample in requisite amount of time.\\

Most of proposed approaches only consider effectiveness of compressive sensing in minimizing transmission cost while a few of them take sampling and computation costs into attention as well. In this section, we will review and compare the performance of compressive sensing approach in contrast with other compression techniques in terms of sampling and communication energy cost.\\

In light of these concerns, compressive sensing is a prefect solution to handle these challenges. Compressive sensing allows sensor nodes to sample and compress data at the same time and each sensor is able to sample signal with lower frequency rather than Sahnon-Nyquist rate. As a result, sensor nodes transmit less data, which minimizes data transition cost. In contrast to other compression techniques, sensor nodes are free of communication data among each other to compress data. Therefore, the sample and compress data in an independent fashion which removes communication redundancy among sensor nodes. \\

It is a defult assumption which says sensing energy is not comparable with transmission energy. This assumption does not hold for all applications. Authors in [] consider applications, which utilizes energy hungry sensors. Numerical experiments show that for some applications compressive sensing is more energy efficient solution rather than data compression techniques.

\begin{figure}[h]

\centering

**\includegraphics**[width=0.85\textwidth]{survey1}

\caption{Comparison of $E\_{comm}, E\_{sens} and E\_{comp}$}

**\label{fig:survey1}**

\end{figure}

\begin{figure}[h]

\centering

**\includegraphics**[width=0.85\textwidth]{survey2}

\caption{$E\_{comm}, E\_{sens} and E\_{comp}$}

**\label{fig:survey2}**

\end{figure}

In [][] authors did comprehensive numerical and simulation analysis to find out the advantage of compressive sensing rather than other data gathering solutions. For evaluation they consider different type of sensors like as $CO\_2$, acceloremeter, temperature and different sensor nodes like as TelosB. Figure \ref{fig:survey2} and \ref{fig:survey3} compare the energy efficiency of raw data gathering and compressive sensing based approach for SHT1X and CO2 sensors. In their evaluation they consider applications, which require energy hungry sensors. As it is depicted in figure, sampling energy is most dominant energy consumption parameter while using compressive sensing we can save around 75$\%$ rather than using only raw data gathering. Figure shows that for CO2 the main energy consumption parameter is sampling operation which consumes 99$\%$ of sensor node energy. However, employing compressive sensing technique reduces this cost to 25$\%$. For same application, they also compare compressive sensing approach with other compression techniques like as transform coding. According to [][], table shows that compressive sensing outperforms other compression techniques in terms of sampling and overall energy consumption. In general the overall number of operation in case of transform coding is equal to$2N^2+ 2N\log(N) + N + 2K$ where N is the overall number of signal coefficients and K is sparsity. Compressive sensing requires less number of computational operations rather than transform coding and its equal to $MN + N + M$ where M is the number of measurements.

\begin{table}[h]

\centering

\caption{Numeric Comparison berween CS and TC}

**\label{table:t1}**

\begin{tabular}{ccccc}

\hline

& E\\_Sampl & E\\_Saving & R\\_mean & Compelexity \\ \hline

\multicolumn{1}{|c|}{CS\\_temp} & \multicolumn{1}{c|}{75\%} & \multicolumn{1}{c|}{74.9\%} & \multicolumn{1}{c|}{0.06} & \multicolumn{1}{c|}{O(M)} \\ \hline

\multicolumn{1}{|c|}{TC\\_temp} & \multicolumn{1}{c|}{0\%} & \multicolumn{1}{c|}{34.3\%} & \multicolumn{1}{c|}{0.022} & \multicolumn{1}{c|}{O(N)} \\ \hline

\multicolumn{1}{|c|}{CS\\_CO2} & \multicolumn{1}{c|}{50\%} & \multicolumn{1}{c|}{49.4\%} & \multicolumn{1}{c|}{0.5} & \multicolumn{1}{c|}{O(M)} \\ \hline

\multicolumn{1}{|c|}{TC\\_CO2} & \multicolumn{1}{c|}{0\%} & \multicolumn{1}{c|}{0.06\%} & \multicolumn{1}{c|}{0.37} & \multicolumn{1}{c|}{O(N)} \\ \hline

\end{tabular}

\end{table}

Most of existing compression techniques is applicable for scenarios wherein computational energy cost is insignificant comparable to communication cost. The energy requirements of data transmission mainly depend on packet size to be transmitted. Authors in[][], address the effectiveness of compressive sensing in term of communication cost. Their simulation results prove that compressive sensing still is useful for the scenarios which data sampling cost is not comparable with communication cost. Simulation results shows that compressive sensing provides better lifetime rather than transform coding approach.