IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 August 2023, accepted 12 September 2023, date of publication 18 September 2023,
date of current version 22 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3316616

RESEARCH ARTICLE

DRDE: Dual Run Distribution Based Encoding Scheme for Sustainable IoT Applications

PRATHAM MAJUMDER[1], (Member, IEEE), PUNYASHA CHATTERJEE[2], (Senior Member, IEEE), SAURAV MALLIK[3], (Member, IEEE), AMAL AL-RASHEED[4], MOHAMED ABBAS[5], MALAK SAEED M. ALQAHTANI[6], AND BEN OTHMAN SOUFIENE[7]

[1]Department of Computer Science and Engineering, University of Calcutta, Kolkata 700073, India
[2]School of Mobile Computing and Communication, Jadavpur University, Kolkata 700032, India
[3]Department of Environmental Health, Harvard T. H. Chan School of Public Health, Boston, MA 02115, USA
[4]Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh 11671, Saudi Arabia
[5]Department of Electrical Engineering, College of Engineering, King Khalid University, Abha 61421, Saudi Arabia
[6]Department of Computer Engineering, College of Computer Science, King Khalid University, Abha 61421, Saudi Arabia
[7]Prince Laboratory Research, ISITCom, University of Sousse, Sousse 4002, Tunisia

Corresponding author: Ben Othman Soufiene (soufiene.benothman@isim.rnu.tn)

This research was financially supported by Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2023R235), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University (KKU) for funding this research through the Research Group Program under the Grant Number (R.G.P.2/516/44).

ABSTRACT Nowadays, data is ubiquitous and has a significant influence on our day-to-day activities due to the emergence of high speed internet and widespread use of sensor-enabled Internet of Things (IoT) devices. Owing to major improvement in *sampling rate* of sensors in recent years, a low variation of the sensed physical parameters is witnessed during a small time interval of observation, which in turn shows high correlation in time domain. Data-critical applications like personal healthcare monitoring, video surveillance, and other applications where data dropping creates significant barriers are attracted by high correlation data. However, due to their power-constrained nature, such applications do not benefit much from the transmission of redundant data. The ideal solution to this problem could possibly be achieved by adopting typical lossless source coding strategy with low-complex design. This paper presents a novel encoding scheme termed as *Dual Run Distribution based Encoding* (DRDE) scheme by exploiting high correlation of sensor data to suitably encode them using symbol run statistics, leading to a reduced length of data with a very large percentage of 0s. Employing silent symbol based communication, the transmitter can be kept in silent state during periods of the most dominant symbol '0' in the encoded messages and using a hybrid $FSK - ASK$ modulation/demodulation technique for communication with a non-coherent receiver results in a significant reduction in transmitter energy. We simulate the proposed sensor data encoding technique on real-life data with low-cost, low data-rate transceivers like CC2420. Simulation results show about 88% (theoretical) and 79-82% (practical) savings in transmitter and 12% (theoretical) 23.5% (practical) savings in receiver energy over conventional BFSK with real-life sensor dataset. Furthermore, our proposed method outperforms in terms of overall energy savings and reduction of CO_2 footprint, generating $1.48 - 0.041$ mg/day, which is 78% lesser than conventional BFSK modulation scheme, making our proposed scheme suitable for sustainable IoT applications in WSN compared to existing schemes. Furthermore, we investigate the influence of various data compression algorithms on computation time, CPU power consumption, and transmission cost on an LPC2148 microcontroller built upon a 16-bit/32-bit ARM7TDMI chipset.

INDEX TERMS Energy-efficient communication, wireless sensor networks, Internet of Things, sustainable computing, sensor data correlation.

I. INTRODUCTION

With the tremendous proliferation of cutting-edge communication network with lightning-speed 5G technology in the

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Feng.

field of Internet of Things (IoT), practically every individual in a metropolitan or urban region is linked to the internet via one or more communication devices. According to CISCO's global traffic forecasts research [1], yearly global IP traffic would reach up to 4.8 ZB per year by 2023, a 220% increase over 2017. Around 71% of the IP traffic would be generated by wireless and mobile devices. Under these circumstances, 82% of IP traffic will be IP video traffic (a 75% increase from 2017), along with video surveillance (3%), virtual reality and augmented reality (6%), consumer Video-on-Demand (VoD) (∼5%), and online gaming (∼3%) due to the availability of wireless communication infrastructure with a vast amount of low-cost, powerful resource forms *Wireless Multimedia Sensor Networks* (WMSN). Smart healthcare systems [2], energy [3], tele-medicine [4], retail [5], transport [6], video streaming (on-demand and live), monitoring through surveillance cameras, and so on are exemplifications of IoT-based applications.

A wireless sensor network is a collection of a large number of low-cost nodes and is highly resource-constrained in terms of computational power and energy capacity-based mobile or stationary sensor nodes. With the exponential expansion of smart applications, the sensor data is used to make suitable judgments using a cloud/fog computing paradigm [7], [8], [9], [10], [11], [12], [13], depending on the specific application. The essential prerequisite for sustainable computing with aggregated sensor data on a cloud/fog computing paradigm is that these energy-constrained sensor devices must consume minimum energy. Apart from computing and sensing energy needs, communication energy is the principal source of energy drain in sensor devices deployed in WSNs. Therefore, incorporating *fair energy-efficient* communication schemes in these sensor devices helps improve effective lifetime of broad range of intricate sensing-driven IoT applications.

Much research in the WSN domain has been conducted in order to improve network lifetime by reducing the energy requirement of sensor devices. Several approaches for ensuring energy-efficient communication in WSNs have been investigated in the literature, encompassing several layers of the communication protocol stack. Sleep scheduling in TDMA [14], preventing and minimizing packet collisions, for example, to reduce packet re-transmissions [15], [16], [17], is a common strategy in MAC layer-based solutions for lowering energy consumption. Physical also plays potential role in this issue.

WSN-based IoT applications often use low-cost, low-power modulation techniques e.g., *On-Off Keying* (OOK), *Amplitude Shift Keying* (ASK), and *Phase Shift Keying/ Frequency Shift Keying* (PSK/FSK) [18], [19] at the physical layer. Sharma et al. [20] provide a performance review of numerous narrow-band digital modulation methods used in WSNs. According to several researchers, performance of the *bit error rate* (BER) in an *Additive White Gaussian Noise* (AWGN) channel utilizing a variety of narrow-band digital modulation schemes i.e., Binary Phase Shift Keying (BPSK), Quadrature Phase Shift Keying (QPSK), and 8,16- PSK and QAM, is the key factor for reducing transmission energy. Their study demonstrates that while BPSK has the highest BER performance, it has the lowest spectral efficiency, making it suitable only for low data rate WSN applications.

Traditionally, every bit in the message (represented in binary), whether it be a 0' or '1', is modulated with respective carrier signal to form the modulated signal. Thereby, predetermined measure of energy is incurred in transmitting every bit of the message, which is known in the literature as the *Energy-based Transmission* (EbT) strategy [21]. But this method suffers from exponential growth of energy depending on the number of bits representing per transmitted encoded string. In comparison to EbT, one can choose specific bit intervals during which the transmitter does not transmit at all, saving some energy, which is known in literature as *communication through silence* (CtS) proposed by Zhu and Sivakumar [27]. Unfortunately, this strategy has the limitation of experiencing exponential communication time. Based on this novel idea, a multitude of energy-efficient communication methods have been devised [28], [29], [30], [31].

These techniques are essentially based on universal strategies in which the message to be delivered is first encoded using a suitable source coding technique to yield a highly asymmetric frequency of occurrence of the symbols in the encoded data string. The transmitter will remain silent during periods of dominating symbol and will be active during the occurrences of other low-frequent encoded symbols during transmission.

In this work, our goal is to design an efficient encoding scheme that achieves two objectives: reducing the data length and minimizing the most power consuming active/transmission time interval of the transceiver and to keep the transceiver in low-power silent state. To accomplish this, our proposed encoding scheme employs a highly asymmetric distribution of encoding symbols in the encoded data strings, ensuring that one symbol has a significantly higher frequency of occurrence compared to the others.

A. RELATED WORKS

Energy utilization within Wireless Sensor Networks (WSNs) occurs during various stages, including data sensing, processing, and communication. Among these stages, data reception and transmission consume more energy in comparison to sensing and processing. To mitigate potential data conflicts stemming from data collection across diverse sensors, the adoption of a efficient data management framework becomes crucial. The imperative of minimizing data loss during this process is underscored. Given the substantial sensor volumes and their sometimes inaccessible locations, the replacement or recharging of low-power sensor batteries is unfeasible. This highlights the need for vigilant monitoring of energy consumption and management. Simultaneously, it accentuates the importance of formulating strategic approaches to conserve energy within the realm of Wireless Sensor Networks (WSNs).

This paper endeavors to confront the issue of energy consumption, a formidable threat to the sustained operation of Wireless Sensor Networks (WSNs). The approach adopted by numerous researchers to combat the energy consumption challenge involves reducing data size prior to transmission and employing streamlined methodologies for seamlessly conveying this data throughout the network to its ultimate destination or sink. This objective can be achieved through techniques such as *data compression*, the deployment of efficient *data aggregation* approaches, and the utilization of optimized *routing techniques*. Among these strategies, *data compression* stands out as a potent energy-conservation tactic, effectively reducing the volume of data transmitted across the network and consequently curtailing transmission time.

For simplicity, if we assume the uncompressed data size is n-bits and e_b units represent the transmission energy associated with each bit, proposed by Zhan and Dai [21], the total energy consumption by the device can be approximated as ne_b units. Now, if any data compression strategy is employed within this transmission scheme, utilizing a compression factor denoted as k bits, the total transmission energy can be reduced by a factor of k i.e., $e_b\lceil\frac{n}{k}\rceil$ units. Unfortunately, this data compression strategy will lead to a proportional increase in hardware complexity by a factor of k. This arises due to the fact that encoding a larger number of bits within a single symbol reduces the transmission time but concurrently escalates hardware complexity by a factor of 2^k for binary data. This phenomenon arises from the presence of 2^k distinct potential values in a k-bit encoding scheme, each requiring a unique means of identification. As a result, achieving an optimized overall communication cost necessitates the optimization of both the transmission cost (ne_b) and the hardware cost ($e_b\lceil\frac{n}{k}\rceil$). The goal is to design an effective data compression strategy tailored specifically for Wireless Sensor Networks (WSNs).

Uthayakumar et al. [22] introduced a two-stage data compression framework. In the initial stage, the *neighborhood correlation sequence* (NCS) algorithm is employed for *bit reduction*. This algorithm generates two distinct code words for each sensor data information, based on two traversal processes: one using 0s and the other using 1s. The second stage involves encoding the code words using the *Lempel-Ziv-Markov chain algorithm* (LZMA) for data compression. The codeword with the fewest bits is chosen as the optimal codeword. Let $\mathcal{G}$ represent the raw sensor dataset, consisting of n bits. The total number of bits required to store the optimal codeword of the final dataset is denoted as $\mathcal{G}_{\text{opt}} = \sum_{i=1}^{n}\phi(i) + c(i)$, where ϕ represents the NCS bits and c signifies the additional eight control bits to account for the optimal number of bits in the compressed data. The LZMA algorithm is then employed to encode the optimal codewords generated by the NCS algorithm (ϕ).

The paradigm of Distributed Source Coding (DSC) has been introduced by researchers based on the Slepian-Wolf theorem [23]. This theorem facilitates the compressed encoding of two statistically dependent signals, denoted as X and Y, through separate encodings using a joint decoding process. Slepian and Wolf's investigation delved into the challenge of independently encoding correlated sources. Consider two information sources, (X, Y), each producing sequences of outputs $X_1, X_2, X_3, \ldots$ and $Y_1, Y_2, Y_3, \ldots$. If each source is encoded/decoded individually without knowledge of the other, the minimum required rate to encode these two sources is clearly $H(X) + H(Y)$. However, when X and Y are not independent, joint source coding/decoding can reduce the rate to $H(X, Y) < H(X) + H(Y)$. This becomes particularly pertinent when separate encoding must be carried out for both X and Y. Csiszar later demonstrated that linear block codes can be employed to tackle this problem, and their performance is bounded by an error exponent [24]. DSC finds utility when there exists correlation among nodes. This holds true in conventional Wireless Sensor Networks (WSNs), where significant correlation often exists between neighboring nodes. In the DSC method, suppose $X_1, X_2, X_3, \ldots, X_N$ represent N distinct source nodes transmitting data to a sink node, and $R_1, R_2, R_3, \ldots, R_N$ represent their respective encoding rates. The overall encoding rate of the system can be expressed as $R_N \leq \left(\frac{X_N}{X_{N-1}}, X_{N-2}, X_{N-2}, \ldots, X_1\right)$. Let $R_{c1}, R_{c2}, R_{c3}, \ldots, R_{cN}$ denote the compression rates corresponding to sources $X_1, X_2, X_3, \ldots, X_N$, respectively. As per the Slepian-Wolf theorem for correlated sources, it holds that$R_{c1}, R_{c2}, R_{c3}, \cdots, R_{cl} \geq H(X_{i1}, X_{i2}, X_{i3}, \cdots, X_{il}/X_{j1}, X_{j2}, X_{j3}, \cdots, X_{jl'})$, where, $l + l' = N$ and $\{i1, i2, i3, \cdots, il\} \in \{1, 2, \cdots, N\}/\{j1, j2, j3, jl'\}$.

Ketshabetswe et al. [25] introduced the Adaptive Lossless Data Compression (ALDC) algorithm. In this algorithm, blocks of sampled data are compressed simultaneously using two Huffman Tables known as Adaptive Lossless Entropy Compression (ALEC) with a 2 Huffman-Table configuration and ALEC with a 3 Huffman-Table configuration.

The simulated methodology discussed in [26] showcases a reduction in Root Mean Square Error (RMSE) values and an increase in R2 values (higher coefficient of determination) across varying compression ratios. This study introduces an inventive approach to enhance networks through the utilization of the Distributed Source Coding scheme.

B. OUR CONTRIBUTION

In this paper, we propose a novel lossless block data encoding scheme exploiting temporal partitioning of the raw sensor data to achieve significant reduction in the redundant sensor data, resulting in notable mitigation in transmitter and/or receiver energy consumption. Our proposed method follows two major steps: i) incorporation of *Dual Run Distribution based Encoding* (DRDE) scheme on highly correlated sensor data to increase the high frequent silent symbols in the encoded data, and ii) application of hybrid ASK-FSK modulation technique on encoded data transmission to keep the transmitted data in the low-power silent state to reduce transmission energy consumption and overall CO_2 footprint in the environment.

It is evident that any raw sensed data is finally represented as string of binary bits of 0's and 1's in the physical layer. The transmission energy of the sensed data depends on the distribution pattern associated with the corresponding binary string. A good energy efficient source coding scheme not only reduces the transmission duration by means of compression of raw data, but also enhances the silent symbol duration in the transmitted data string. Both compression factor and silent symbol communication have large impact on energy savings in energy-constrained wireless sensor network-based applications. While the previous method [32] focused solely on runs of 1's in the data string, our proposed scheme addresses both runs of 1's and runs of 10's. As a result, we achieve a higher compression rate while still maintaining an acceptable error margin of approximately 4.5%.

A predetermined number of instances of consecutive k success in $\mathcal{N}$ Bernoulli trials has been proposed by Sinha and Sinha [32]. In this paper, we will investigate the problem of the occurrences of the exact i_{k_1} repetitions/runs of 1's of length k_1 and i_{k_2} repetitions/runs of 10's of length k_2 in a binary string of length $\mathcal{N}$. We analyze this run distribution problem and also derived the expression for the number of binary strings that contain exactly i_{k_1} runs of 1's of length k_1 and i_{k_2} runs of 10's of length k_2 in all possible binary string of length $\mathcal{N}$, $1 \leq k_1, k_2 \leq \mathcal{N}$, using generating function based approach.

Our novel two fold encoding approach takes use of high correlation characteristics of sensor data values to generate highly asymmetric numbers of 1's and 0's in the encoded bit string. The use of our proposed encoding system, together with the notion of silent communication during dominant symbol periods, helps to lower the device's energy usage. For data transfer from sensor nodes, we simulate our proposed system with various real-life sensor data using a common low-cost low-data rate commercial transceiver such as the CC1100 [33], CC2420 [34], Maxim2820 [35], and RFM TR1000 [36].

Simulations with real-world sensor data reveal our given block data encoding approach yields transmitter energy savings of up to $76-82\%$ and about 23.5% receiver energy savings over traditional binary FSK (BFSK). These simulation results demonstrate that the observed gains in transmitter energy savings, and thus the overall reduction in CO_2 footprint with our proposed technique, are always greater than those obtained by the existing best-known schemes for energy savings CNS [37] or PCA [38] that also employ temporal correlation of sensor data. Thus, our suggested approach outperforms existing strategies in terms of total energy-efficiency and CO_2 footprint reduction with generation of $1.48-0.041$ mg/day, making it the most effective and novel among comprehensive data-critical applications as well as in other applications.

The remaining sections of the article are arranged as follows. In section II, we discuss about the basic concept behind dual run statistics method and notations of various parameters used in the paper followed by encoding and decoding algorithms and a generating function based solution to the counting problem in Section III. In Section IV, we describe data aggregation model of sensor networks. Section V describes the effect of real-life transceiver characteristics involving in communication scenario. In section VI, we describe implementation of proposed scheme,s followed by simulation results in section VII and conclusion in Section VIII.

II. NOTATIONS AND BASIC IDEAS

In our proposed *Dual Run Distribution based Encoding* (DRDE) scheme, consider a binary string (an equivalent representation of raw binary data stream) of length $\mathcal{N}$. We define R_k and S_k, respectively, to denote a run of 1's and 10's of length k in the binary string. In order to distinguish any two successive identical or non-identical repetitive sequences patterns of varying length, let's say k_1 and k_2, $1 \leq k_1, k_2 \leq \mathcal{N}$ in a bit string must be separated by minimum one zero. For the ease of calculation of this problem, we recommend appending of two zeros at the extreme left and right bit positions of the given binary string. Furthermore, let's assume, any run R_k or S_k, $1 \leq k_1, k_2 \leq \mathcal{N}$ that is separated by another runs with an incorporation of an isolation bit, appended at the left of each runs is conveyed by $0R_k$ as Z_k and by $0S_k$ as Y_k.

Example 1: Assume, $\mathcal{N} = 10$ and 1111010101 be the binary string of length 10 contains both runs of 1's and 10's. R_4 and S_3 denote a run of 1's of length 4 and a run of 10's of length 3, respectively (after appending two 0's at the most significant bit (MSB) and least significant bit (LSB) positions of binary string), which is also separated by one zero at the 6^{th} position from left. Hence, this augmented binary string can be denoted by $0R_40S_3$, i.e., Z_4Y_3.

Observation 1: For some values of k, $1 \leq k \leq \mathcal{N}$, we need to append two zeros at the most significant bit (MSB) and least significant bit (LSB) positions to obtain the string formats Z_k and Y_k, respectively, if a binary string begins with a run of 1's or 10's at the most significant bit (MSB) position or ends with 01's at the least significant bit (LSB). Thus each such augmented string Z_k $(1 \leq k \leq \mathcal{N})$, will be the string of length $k+1$. Likewise, each such augmented string Y_k $(1 \leq k \leq \mathcal{N})$, will become the string of length $k+1$ bits. Therefore, regardless of the most significant bit position of a binary string, whether it starts with a run of 1's or 10's, by adding two extra zeros at MSB and LSB positions will provide one-to-one correspondence between augmented string and the given binary string. The final binary string, however, will be $\mathcal{N}+2$ in length.

Let, $\mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)}$ denotes the entire count of $\mathcal{N}$ length binary strings that contains ultimately i_{k_1} counts of R_{k_1}'s and i_{k_2} counts of S_{k_2}'s, $1 \leq k_1, k_2 \leq \mathcal{N}$, present in the string. Therefore, to determine such count of augmented string of length $\mathcal{N}+2$ bits that contains i_{k_1} runs of 1's i.e., Z_{k_1} alongwith i_{k_2} runs of 10's i.e, Y_{k_2}'s $(1 \leq k_1 \leq \mathcal{N})$ and $1 \leq k_2 \leq \mathcal{N}$ is analogous to $\mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)}$. To address this problem, we first extract i_{k_1} and i_{k_2} amount of Z_{k_1} and Y_{k_2} substrings, respectively from the augmented binary string of length $\mathcal{N}+2$ bits. Hence, the total count of left over bits is thus $(\mathcal{N}+2)-(k_1+1)i_{k_1}-(2k_2+1)i_{k_2}$. These left over bits may now be filled with any possible combination of 0's or 1's making random runs of 1's or 10's, however the substrings Z_{k_1} and Y_{k_2} will not be repeated further. Instead of being contiguous, it is anticipated that the remaining $(\mathcal{N}+2)-(k_1+1)i_{k_1}-(2k_2+1)i_{k_2}$ left over bits will be dispersed throughout all conceivable probable bit locations. Now, a contiguous j_1 and j_2 runs of 1's and 10's of length (j_1+1) and (j_2+1) bits creates a block of size Z_{j_1} and Y_{j_2}, respectively, where, $j_1 \neq k_1$ and $j_2 \neq k_2$. As a consequence, the left over bit position may be segmented into an arbitrary v blocks, where each block must represent either Z_{k_1} or Y_{k_2}, where, $j_1 \neq k_1$ and $j_2 \neq k_2$ and $0 \leq j_1, j_2 \leq \mathcal{N}$. This is a technique for calculating all possible binary strings using an ordered partitioning mechanism.

*Example 2: Consider binary string 1110011000000101 of length 16 bits, containing totally two different runs of 1's i.e., of length 3 and 2 bits, respectively, and a run of 10's (started with 0 i.e., 010) of length 5 bits. Appending MSB and LSB 0's on the given binary string makes the length of the augmented string (011100110000001010) of 18 bits. Apart from these two different runs, there are 18-12= 6 bit positions that are termed to be the left over bit positions and is represented in this augmented string as 0111*011*****01010, where the positions of left over bits are specified by the symbol '*'. Thereafter, either of $Z_{j'}$ or $Y_{j''}$, where, $j' \neq 2$ and $j'' \neq 1$ can be used to fill the successive left over bit positions and a block of 6 bits positions is formed with each such $Z_{j'}$ or $Y_{j''}$.*

III. COUNTING STRATEGY

We follow three basic steps to solve our counting problem to determine all possible combination of of $Z_{j'}$ or $Y_{j''}$, in the left over positions of a binary string of length $\mathcal{N}$ bits, in presence of a certain frequencies of Z_j or Y_j, where, $j' \neq j$ and $j'' \neq j$.

Step 1: Ordered partitioning of the left over bit positions into v-blocks (say). We first remove i_{k_1} and i_{k_2} blocks of size k_1+1 and $2k_2+1$ bits, respectively, from the binary string of length $\mathcal{N}+2$ (after appending 0's in MSB and LSB in the original binary string). Hence, the total number of ordered partitioning is calculated for the integer $t=(\mathcal{N}+2)-(k_1+1)i_{k_1}-(2k_2+1)i_{k_2}$ for exactly v blocks, where $v \geq 1$ for $k_1+k_2 < \mathcal{N}$ and $v \geq 0$ for $k_1+k_2 = \mathcal{N}$. Partitioning should confirm the presence of no blocks of size (k_1+1) and (k_2+1) in the left over bit string.

Step 2: Insertion of i_{k_1} and i_{k_2} number of Z_{k_1}'s and Y_{k_2}'s, respectively in the above partitioned block, i.e., after partitioning is obtained in Step 1 we insert either i_{k_1} number of Z_{k_1}'s or i_{k_2} number of Y_{k_2}'s in each block or both and compute the feasible distinct possibilities.

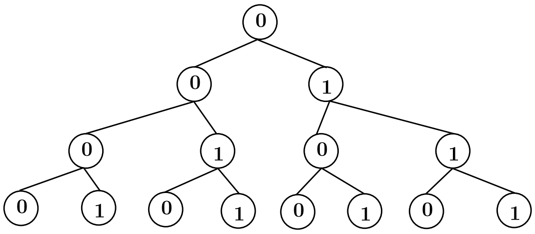


FIGURE 1. Tree representation with equal likelihood of binary digits with $\mathcal{N}=4$ and MSB '0'.

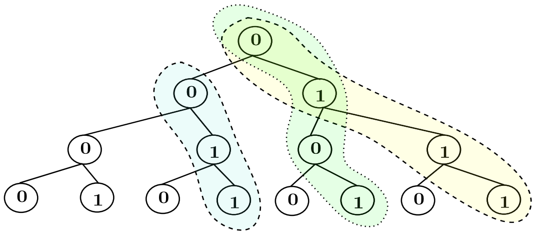


FIGURE 2. Identification of runs of 1's and 10's in DRDE scheme.

Step 3: Finally we sum over all possible values of v obtained in Step 2 to get $\mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)}$.

Fig. 1 represents a 4-bit tree, which will be useful in characterizing the generation of the DRDE encoding process by considering equal likelihood of all possible values of the bit strings with MSB to be '0'. This tree starts from a root node denoted by '0' at level 0, which will have two children at level 1 corresponding to the two different possible values - '0' and '1' at the $(\mathcal{N}-2)^{th}$ digit position of an $\mathcal{N}$-digit DRDE number. However, the LSB will be always 0. In order to compute runs of 1's or 10's from the tree, we have to find the start traversing from root node, which is the MSB '0'. At level 1, we have two children of the root node, i.e., 0 and 1, respectively, which in turn forms the sub-string '00' and '01', respectively. '00' sub-string is insignificant in terms of run analysis, whereas sub-string '01' gives a new possibilities of the beginning of runs of either 1's or 10's. At level 2, considering all possible combinations of sub-string formation, we have formed 4 different sub-strings, namely '000', '001', 1010' and '011', respectively. Out of all four sub-strings, we can form minimum runs of 1's, i.e., '011' and of 10's, i.e., '010'. However, new runs can be determined from sub-string '001' by examining the fourth bit. The construction of the tree is continued in the same way with nodes at the successive levels corresponding to digit values at the lower significant digit position in the DRDE encoding scheme. Therefore, for run analysis, the minimum criteria is to examine the first three digits in every sub-string. Fig.2 depicts the various possibilities of runs associated with different levels of the tree.

A. SOLUTION BASED ON GENERATING FUNCTION

We solve the counting problem using generating function [39], [40]. It is evident that, if either $k_1 = \mathcal{N}$ or $k_2 = \mathcal{N}$, there is only one type of run either be run of 1's or be 10's in the whole string, and therefore, $\mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)} = 1$. We use two different polynomial functions to represent the occurrence of

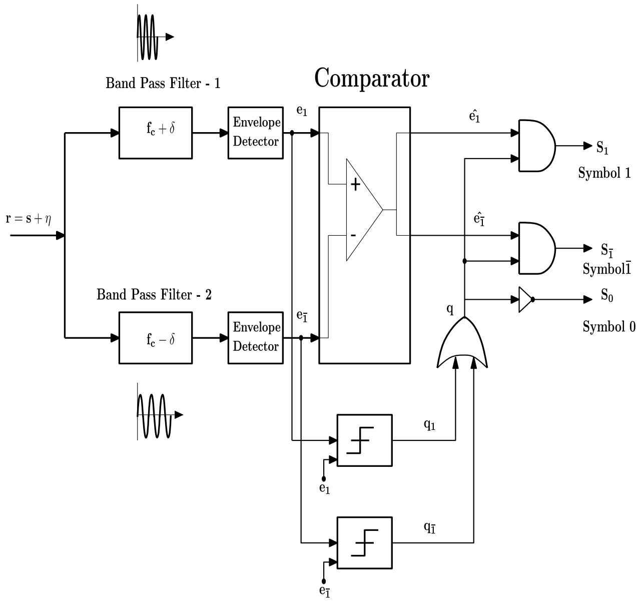


FIGURE 3. Schematic diagram of non-coherent *FSK* − *ASK* receiver.

Z_i and Y_i respectively, for any value of i, $0 \leq i \leq \mathcal{N}$ for any bit positions in augmented binary string. The polynomial maps the required length to represent any of such Z_i or Y_i at any given position of an augmented binary string to its corresponding representation in terms of number of bits. We use a bi-variate polynomial $f(x, z)$ to specify the required length for two different runs.

Thus simply, the run of 1's followed by a zero is represented by z^L, (L = overall length of the run). Hence, the minimum run of 1's to be encountered is '011' which is represented by z^3. Similarly, '0111' which is run of 1's of length 4, by z^4, and so on. In general, any run of 1's of length k requires $k+1$ numbers of bits (k ones are followed by 0), and can be represented by z^{k+1}.

Similarly, the run of 10's preceded by a zero is represented by x^L, i.e., the term '010' is represented by x^3 and 01010 is corresponds to x^5. Whereas, we start calculating run of 1's from 011 and onwards, hence, Thus, after encountering a '01', next step is to verify the succeeding bit, so that it can be a run of 10's or run of 1's. For v block partitioning of integer $t = (\mathcal{N}+2) - (k_1+1)i_{k_1} - (2k_2+1)i_{k_2}$, each block will corresponds to some runs, either be Z_i or be Y_i, therefore, product of such v polynomials represents total number of permutations of runs. Lastly, to exclude the occurrence of either of Z_{k_1} or Y_{k_2}, we need to subtract either z^{k_1+1} or x^{k_2+1}, respectively from each of these polynomials.

Hence, for v such blocks, the partition of the integer t, we represent the generating functions presented in eqn.1, as shown at the bottom of the next page. After that, number of partitions can be calculated in Step 1, taking the coefficient of $x^M z^N$ in $G(x, z)$ from eqn.4, as shown at the bottom of the next page.

Finally, we can represent $G(x, z)$ can be represented as: Thus, the coefficient of $(xz)^{(\mathcal{N}+2)-(k_1+1)i_{k_1}-(2k_2+1)i_{k_2}}$ is equal to the coefficient of $x^M z^N$ in eq. 4, such that, $M + N = (\mathcal{N}+2) - (k_1+1)i_{k_1} - (2k_2+1)i_{k_2} - v$. Therefore, we have, $x^M = x^{v-i-2j-2k_2} x^{2p}$ thus, $p = \frac{M-v+i+2j+2k_2}{2}$ and $z^N = z^{3i-2q+qk_1} z^t$ and thus, $t = N - 3i + 2q + qk_1$. Therefore, we can calculate the required values of $\mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)}$ and $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$ fov $k_1, k_2 \geq 1$ from eqn.5, eqn.6, respectively.

$$\mathcal{C}_{M,N} = \sum_{i=0}^{v} \sum_{j=0}^{v-i} \sum_{q=0}^{i} \binom{v}{i} \Big[(-1)^{j+q} \binom{v-i}{j} \\ \cdot \binom{v-i-j+p-1}{p} \binom{i}{q} \binom{i-q+t-1}{t} \Big] \tag{5}$$

$$N_n^{(i_{k_1},k_1)(i_{k_2},k_2)} = \sum_{v=1}^{(\mathcal{N}+2)-(k_1+1)i_{k_1}-(2k_2+1)i_{k_2}} \binom{v+i_{k_1}+i_{k_2}}{i_{k_1}+i_{k_2}} N_{M,N} \tag{6}$$

Substituting values of p and t in eq. 5 we have,
Hence, final expression of $\mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)}$ is obtained by substituting the value of $N_{M,N}$ in eq. 6,

B. ENCODING SCHEME

Algorithm 1 represents the encoding process of our proposed *Dual Run Distribution based Encoding* (DRDE) scheme including a special case analysis in Algorithm 2.

1) SPECIAL CASE ANALYSIS

In encoding algorithm (Algorithm 1, we have considered the two separate cases where homogeneous runs of 1's or 10's are appearing in the binary data string. In this special case analysis we propose the encoding process of a non-homogeneous runs of 1's and 10's in the binary string.

2) DECODING SCHEME

The decoding process will start from the leftmost bit position is presented in Algorithm 3.

IV. SYSTEM MODEL

We assume that there are m sensor nodes are deployed randomly over a given geographical region R to monitor certain physical parameters, e.g., temperature, pressure, humidity, etc. for quick identification of anomaly events. The i^{th} sensor being denoted by S_i, $1 \leq i \leq m$. Let S be the set of all these m sensors, i.e., $S = \{S_i | 1 \leq i \leq m\}$. This real-time monitoring application requires high efficient continuous monitoring process, so that observation interval (τ) of sensors are very small. We assume that throughout this time interval T, each sensor S_i, $1 \leq i \leq m$ node will produce n data values, so that $T = n\tau$. Therefore, data generated from sensor node S_i at j^{th} observation period is denoted by $\mathcal{D}_i(j)$, $1 \leq i \leq m$, $1 \leq j \leq n$ (which is assumed to be in floating-point format).

The aggregated raw sensor data from all the m sensors during the time interval T are collectively forming a $m \times n$ matrix $\mathcal{D}_{raw}$ is depicted below. The observation data in the matrix $\mathcal{D}_{raw}$ is represented by $D_i(j)$, $1 \leq i \leq m$, $1 \leq j \leq n$. The i^{th} row in $\mathcal{D}_{raw}$ represents the collective sensed data from sensor S_i for over all n time slots. Similarly, j^{th} column in $\mathcal{D}_{raw}$ represents collective sensor data generated from all deployed m sensor nodes in the j^{th} interval.

$$\mathcal{D}_{raw} = \begin{bmatrix} D_1(1) & D_1(2) & D_1(3) & \cdots & D_1(n) \\ D_2(1) & D_2(2) & D_2(3) & \cdots & D_2(n) \\ D_3(1) & D_3(2) & D_3(3) & \cdots & D_3(n) \\ \cdots & & & & \\ \cdots & & & & \\ \cdots & & & & \\ D_m(1) & D_m(2) & D_m(3) & \cdots & D_m(n) \end{bmatrix}$$

Due to effect of sufficiently low observation period (τ) of sensor nodes $S_i, 1 \leq i \leq m$, sensor data are highly temporally correlated. Moreover, due to relatively close and random placement of sensor nodes in geographic region R, most of the sensor data are highly spatially correlated. It is anticipated that, as a result of the asymmetric positioning of the nodes, a certain set of sensors will always be exposed to a specific event, and that their observation data will vary as a consequence [41].

V. EFFECTS OF DEVICE CHARACTERISTICS

The employment of radio devices to transmit sensor data incurs power consumption, even during silent symbol periods, diminishes the energy savings achieved by our proposed encoding approach. To evaluate the total power utilized by a

$$G(x,z) = \left[\left(x^3 + x^5 + x^7 + \cdots - x^{2k_2+1}\right) + \left(z^3 + z^4 + z^5 + \cdots - z^{k_1+1}\right)\right]^v \tag{1}$$

$$G(x,z) = \left[\left(\frac{x}{1-x^2} - x^{2k_2+1}\right) + \left(\frac{z^3}{1-z} - z^{k_1+1}\right)\right]^v \tag{2}$$

$$\begin{aligned} G(x,z) &= \sum_{i=0}^{v}\binom{v}{i}\left(\frac{x}{1-x^2} - x^{2k_2+1}\right)^{v-i}\left(\frac{z^3}{1-z} - z^{k_1+1}\right)^{i} \\ &= \sum_{i=0}^{v}\binom{v}{i}\left[\sum_{j=0}^{v-i}(-1)^j\binom{v-i}{j}\left(\frac{x}{1-x^2}\right)^{v-i-j}\left(x^{2k_2+1}\right)^{j}\right]\left[\sum_{q=0}^{i}(-1)^q\binom{i}{q}\left(\frac{z^3}{1-z}\right)^{i-q}\left(z^{k_1+1}\right)^{q}\right] \\ &= \sum_{i=0}^{v}\binom{v}{i}\left[\sum_{j=0}^{v-i}(-1)^j\binom{v-i}{j}x^{v-i-2j-2k_2}\left(1-x^2\right)^{-(v-i-j)}\right]\left[\sum_{q=0}^{i}(-1)^q\binom{i}{q}z^{3i-2q+qk_1}\left(1-z\right)^{-(i-q)}\right] \\ &= \sum_{i=0}^{v}\binom{v}{i}\left[\sum_{j=0}^{v-i}(-1)^j\binom{v-i}{j}x^{v-i-2j-2k_2}\left\{1 + \binom{v-i-j}{1}x^2 + \binom{v-i-j+1}{2}x^4 + \cdots\right\}\right] \\ &\quad \times \left[\sum_{q=0}^{i}(-1)^q\binom{i}{q}z^{3i-2q+qk_1}\left\{1 + \binom{1-q}{1}z + \binom{1-q+1}{1}z^2 + \cdots\right\}\right] \end{aligned} \tag{3}$$

$$\begin{aligned} G(x,z) = \sum_{i=0}^{v}\binom{v}{i}&\left[\sum_{j=0}^{v-i}(-1)^j\binom{v-i}{j}x^{v-i-2j-2k_2}\sum_{p=0}^{\infty}\binom{v-i-j+p-1}{p}x^{2p}\right] \\ &\times\left[\sum_{q=0}^{i}(-1)^q\binom{i}{q}z^{3i-2q+qk_1}\sum_{t=0}^{\infty}\binom{i-q+t-1}{t}z^{t}\right] \end{aligned} \tag{4}$$

$$\mathcal{C}_{M,N} = \sum_{i=0}^{v}\sum_{j=0}^{v-i}\sum_{q=0}^{i}\binom{v}{i}\left[(-1)^{j+q}\binom{v-i}{j}\binom{v-i-j+\frac{M-v+i+2j+2k_2}{2}-1}{\frac{M-v+i+2j+2k_2}{2}}\binom{i}{q}\binom{i-q+N-3i+2q+qk_1-1}{N-3i+2q+qk_1}\right] \tag{7}$$

$$\begin{aligned} \mathcal{C}_{\mathcal{N}}^{(i_{k_1},k_1)(i_{k_2},k_2)} = &\sum_{v=1}^{(\mathcal{N}+2)-(k_1+1)i_{k_1}-(2k_2+1)i_{k_2}}\binom{v+i_{k_1}+i_{k_2}}{i_{k_1}+i_{k_2}}N_{M,N} \\ &\times\sum_{i=0}^{v}\sum_{j=0}^{v-i}\sum_{q=0}^{i}\binom{v}{i}\left[(-1)^{j+q}\binom{v-i}{j}\binom{v-i-j+\frac{M-v+i+2j+2k_2}{2}-1}{\frac{M-v+i+2j+2k_2}{2}}\binom{i}{q}\binom{i-q+N-3i+2q+qk_2-1}{N-3i+2q+qk_2}\right] \end{aligned} \tag{8}$$

Algorithm 1 Encoding Algorithm

```
procedure Encode(Binary string $S$)
    Step 1: Convert all 01's to $\overline{1}$.
    Step 2: Two cases are considered:
        Case 1: Binary string ($S_{1,k}$) consists of $k$ consecutive 1's followed by a 0, i.e., $01^k$, where $1 < k$.
            Step 2.1.1: Repeat Step 1 in the sequence $S_{1,k}$ to get the augmented string $S_{\overline{1,k}}$, i.e., $\overline{1}1^{k-1}$.
            Step 2.1.2: Convert the leftmost $\overline{1}$ and the consecutive $(k-2)$ 1's followed by $\overline{1}$ in $S_{\overline{1,k}}$ to 1 and $(k-2)$ 0's,
                respectively, while keeping the least significant (rightmost) 1 unchanged. Thus, the modified augmented binary string $\overline{S_{1,k}}$
                is $10^{k-2}1$.
        Case 2: Binary string ($S_{01,k}$) consists of $k$ consecutive (01)'s, i.e., $(01)^k$, where $1 < k$.
            Step 2.2.1: Repeat Step 1 in the sequence $S_{01,k}$ to get the augmented string $S_{\overline{01,k}}$, i.e., $\overline{1}^k$.
            Step 2.2.2: Convert the $(k-3)$ consecutive $\overline{1}$'s between the leftmost $\overline{1}$ and the rightmost pair of $\overline{1}$'s in $S_{\overline{01,k}}$
                to $(k-3)$ 0's, while keeping the leftmost $\overline{1}$ unchanged. Convert the rightmost two $\overline{1}$'s to $1\overline{1}$. Thus, the modified augmented
                binary string $\overline{S_{01,k}}$ is represented as $\overline{1}0^{k-3}1\overline{1}$.
end procedure
```

Algorithm 2 Special Case Analysis: Observation 1

```
Input: Binary string $S_{ob:1}$ of the form $(01)^{k_1}0^{k_2}1^{k_3}$, where $1 < k_1, k_2, k_3$.
Output: Modified augmented binary string $\overline{S_{ob:1}}$.
Step 1: Start scanning from the leftmost bit position of string $S_{ob:1}$.
Step 2: Repeat Step 1 in the sequence $S_{ob:1}$ to get the augmented string $S_{\overline{ob:1}}$, i.e., $\overline{1}^{k_1}0^{k_2-1}\overline{1}1^{k_3-1}$.
Step 3: Repeat Step 2.2.2 and Step 2.1.2 in the augmented string $S_{\overline{ob:1}}$ to convert it to the modified augmented binary string
    $\overline{S_{ob:1}}$, represented as $\overline{1}0^{k_1-2}\overline{1}0^{k_2-1}\overline{1}0^{k_3-2}1$.
```

Algorithm 3 Decoding Algorithm

```
Step 1: When a string starts with 1 followed by multiple 0's and ends with another 1, i.e., $10^k1$ (where $1 < k$), the decoder
    considers the string as a run of 1's of length $(k+2)$. The whole string starts with $\overline{1}$ followed by $(k-1)$ 1's and is represented
    by $\overline{1}1^{k+1}$.
Step 2: When a string starts with $\overline{1}$ followed by multiple 0's and ends with $1\overline{1}$, i.e., $10^k1\overline{1}$ (where $1 < k$), the decoder
    considers the string as a run of $\overline{1}$'s of length $(k+3)$. The whole string is converted into $(k+3)$ $\overline{1}$'s and is represented by
    $\overline{1}^{k+3}$.
Step 3: When a string starts with $\overline{1}$ followed by multiple 0's and also ends with $\overline{1}$, i.e., $\overline{1}0^k\overline{1}$ (where $1 < k$), the decoder
    considers the string as a run of 0's of length $k$, hence, there is no change in the bit pattern.
Step 4: All $\overline{1}$'s in the string will be converted into 01.
```

sensor node's radio device, we consider commercially available radio devices such as the CC1100 [33] and CC2420 [34] chips, which are commonly used for low-power, low-cost WSN IoT applications. However, these devices exhibit significantly higher power consumption in TX (transmit or receive) states compared to their low-power operation mode, referred to as the "idle" or "active" state in subsequent discussions. Let I_{LOW} and I_{HIGH} be the currents drawn by the radio device in the idle state and TX state, respectively, while t_P denotes the duration of one symbol period.

The use of radio devices for transmitting sensor data introduces energy consumption, even during periods of silence, which diminishes the energy savings achieved by our proposed encoding approach. To estimate the total energy consumed by a sensor node's radio device, we consider commercially available radio devices commonly used for low-power, low-cost WSN IoT applications, such as the CC1100 and CC2420 chips [33], [34]. These devices exhibit significantly higher power consumption in TX (transmit or receive) states compared to their low-power operation mode, which we refer to interchangeably as the "idle" or "active" state. We quantify the energy consumption based on the currents drawn by the device in the idle state (I_{LOW}) and TX state (I_{HIGH}), and the duration of one symbol period (t_P). Additionally, energy is also consumed during transitions between the idle and TX states, characterized by the time taken to switch from idle to TX state (t_{RISE}) and the negligible transition time from TX to idle state (t_{FALL}).

Example 3: The CC1100 transceiver chip, draws 1.9 mA current (I_{LOW}) when the radio device is in the idle state. However, the device draws I_{TRAN} = 8.7 mA current while transiting from the idle state to the transmission (TX) state. Moreover, the current drawn by the device in the TX state is I_{HIGH} = 30.3 mA (for 10 dBm), 19.7 mA (for 5 dBm), 16.6 mA *(for* 0 *dBm), and* 14.0 *mA (for* -5 *dBm) output power, respectively. The amount of time required by the transceiver for switching from idle state to the TX state is known as* rise time, *which is* $t_{RISE} = 88.4\mu s$. *The data rate of CC1100 transceiver is* 2.5 *Kbps, refers to the duration of one symbol period is* $t_P = 400\mu s$.

Example 4: The CC2420 chip, the idle state current is $I_{LOW} = 0.426\,mA$, *whereas,* $I_{TRAN} \approx I_{LOW}$, *and the values of* I_{HIGH} *are* 17.4 *mA (at* 0 *dBm),* 14 *mA (at* -5 *dBm),* 11 *mA (at* -10 *dBm,),* 9.9 *mA (at* -15 *dBm), and* 8.5 *mA (at* -25 *dBm) output power, respectively. The time taken by the device to switch from the idle state to the TX state is* $t_{RISE} = 0.1\mu s$. *The data rate of CC2420 transceiver is* 2.5 *Kbps provides* $t_P = 400\mu s$.

We will estimate the energy consumed by the radio devices for different components, assuming a supply voltage of V_{DC} and defining $I_0 = I_{HIGH} - I_{LOW}$.

- E_{base} denotes the energy consumed during transmission when the radio device is in the idle state. It is calculated based on an n-bit binary data. For an encoded string of length n' with a compression factor of r_c (i.e., $r_{comp} = \frac{n'}{n}$), the value of E_{base} can be determined using the following equation:

$$E_{base} = n \times r_{comp} \times t_P \times I_{LOW} \times V_{DC} \tag{9}$$

- E_{trans} represents the energy required for transmitting non-silent symbols, specifically 1 or $\bar{1}$. It is calculated based on the percentage T_{ON}, which represents the ON period or the percentage of 1 and $\bar{1}$ in the encoded string. The expression for E_{trans} is given by:

$$E_{trans} = n \times r_{comp} \times T_{ON} \times (t_P - t_{RISE}) \times I_0 \times V_{DC} \tag{10}$$

- E_{switch} denotes the switching energy consumed during the transition from the *idle* state to the TX state for transmitting non-silent symbols (1 or $\bar{1}$). It is calculated by considering the probability of non-silent symbol periods as T_{ON}. The expression for E_{switch} is given by:

$$E_{switch} = n \times r_{comp} \times T_{ON} \times t_{RISE} \times I_0 \times V_{DC} \tag{11}$$

The total energy consumed in transmitting an encoded string of length n' can be determined by considering the sum of E_{base}, E_{trans}, and E_{switch}.

$$E_{TOTAL} = E_{base} + E_{trans} + E_{switch} \tag{12}$$

A. TOTAL ENERGY CONSUMPTION ACCOUNTING FOR DEVICE CHARACTERISTICS

Taking into account the power dissipation of the transceiver in different scenarios as discussed in Section V, we proceed to determine the total energy needed for transmission. As a result, we can express the total energy consumed by a wireless sensor network (WSN) in a single day as follows:

$$E_{wsn} = E_{TOTAL} \times m \times 86400 \times D_p, \tag{13}$$

Here, E_{TOTAL} represents the value obtained from Equation (12), m denotes the number of deployed sensors, a single day is comprised of 86400 seconds, and D_p signifies the total number of data packets transmitted per second. By considering the emission rate R_e associated with the specific energy source, we can calculate the equivalent daily CO_2 footprint of communication energy, as explained in [44].

$$CO_{2_{fp}} = R_e \times E_{wsn}. \tag{14}$$

For instance, if coal is used as the energy source, then $R_e =$ 0.9 Kg/KWh [44].

VI. IMPLEMENTATION OF PROPOSED COMMUNICATION SCHEME

we propose a modulation scheme that combines non-coherent Frequency Shift Keying - Amplitude Shift Keying ($FSK - ASK$) methods. This scheme allows us to transmit all three encoded symbols: 0, 1, and $\bar{1}$. Figure 3 illustrates our proposed hybrid non-coherent $FSK - ASK$ receiver. This receiver module utilizes two carrier frequencies: $f_c + \delta$ and $f_c - \delta$ for transmitting the two non-silent symbols, '1' and $\bar{1}$, respectively. During transmission, the transmitter remains in a deep-sleep silent state in the interim time of the high-frequency dominant symbol '0' appearing in the source-coded message.

In Fig. 3, the received signal r consists of the baseband signal (s) and the Additive White Gaussian Noise ($AWGN$) signal (η), where η is a random variable following a Normal Distribution denoted by ($\eta \sim \eta(\mu, \sigma^2)$). The proposed receiver module features three output ports: S_1 and $S_{\bar{1}}$, corresponding to the non-silent symbols 1 and $\bar{1}$ transmitted with the carrier frequencies $f_c + \delta$ and $f_c - \delta$, respectively.

On the other hand, the third output terminal S_0 corresponds to the silent symbol '0' specified during transmission. The receiver module comprises several sub-modules, including two bandpass filters (BPF) with center frequencies $f_c + \delta$ and $f_c - \delta$, two envelope detectors, one comparator, two threshold detectors (ED), two AND gates, one OR gate, and one inverter (NOT gate).

A schematic diagram of the hardware implementation of dual run detection process in the binary string is shown in Fig. 5. This design comprises two shift registers, 5 registers (temp1, temp2, temp3, temp4, temp5), 4 symbol comparators (comp1, comp2, comp3, comp4), one NOT gate, three delay circuits and one 2-to-1 multiplexer. Note that in the first clock signal, the LSB '0' of the input shift register is loaded directly to the LSB position of output shift register and loaded in the temp1 register. In the second clock signal, second data is extracted from the input shift register, which will be compared using comp1 symbol comparator with temp1 bit value. If the bit second bit is '0', i.e., the sub-string sequence has become '00'. This event indicates the non-confirmation of runs of 1's and 10's so far and therefore is to be loaded directly to the second bit position of the output shift register. However, if the second bit is '1' forming the sub-string '01', that event assures the start of runs of 1's or 10's. Hence, this

TABLE 1. Specifications of different Radios.

Radio Specification	CC1100 [33]	CC2420 [34]	Maxim 2820 [35]	RFM TR1000 [36]
Rise time (μs)	88.4	192	3	16
Symbol duration (μs)	400	400	20	40
Data rate (Kbps) with silent symbol strategy	2.5	2.5	50	25
Supply voltage (V)	2.5	2.5	2.7	3
Transmit state, I_{high}(mA)	30.3	17.4	70	12
Active state, I_{low}(mA)	1.9	0.426	25.0	0.0007

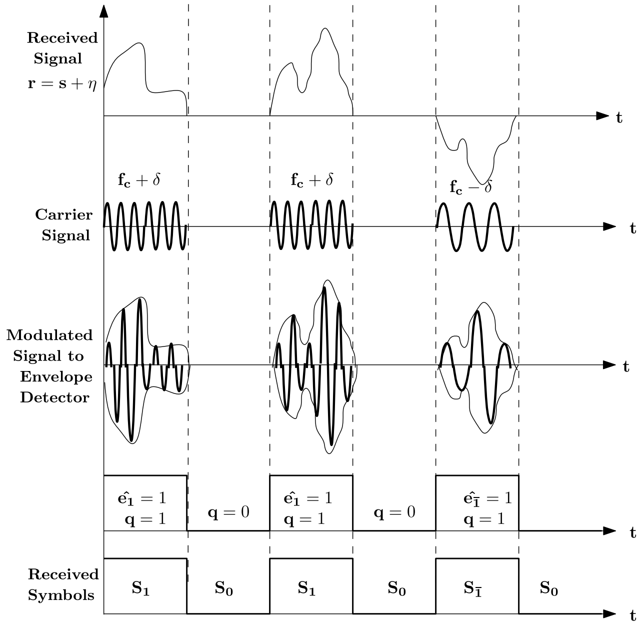


FIGURE 4. Working process of non-coherent *FSK* − *ASK* receiver.

bit is loaded into temp2 register. In the third clock signal, third bit is extracted, which decides the sub-string to be either a run of 1's or 10's. The validation is performed using symbol comparator i.e., comp2 and result is stored in the respective temporary registers as shown in Fig. 5. For the implementation of Algorithm. 1, we have to consider the fourth bit of the input shift register to toggle all the intermediate bits of run of 1's to '0' for the sequence '0111' or run of 10's to '0' for the sequence '0101'. To implement this argument, the fourth bit of the input shift register is first verified to be it a '0' or '1'. If the bit is '0', it makes the sequence '0110' or '0100'. Due to absence of runs in both cases, the third and fourth bit will be loaded as it is in the output shift register without alteration. However, if the fourth is '1', that forms the sub-strings '0111' or '0101'. This system identifies the presence of runs of either 1s or 10s in the sub-string, and Algorithm. 1 makes the third bit '0' (implemented using inverter) and forth bit either '1' or '$\bar{1}$'. The conversion of sub-string '01' to '$\bar{1}$' is trivial and not shown explicitly in the Fig. 5. Both the output lines are connected to a 2-to-1 multiplexer. The output line of the multiplexer will be '0' or '1' if the comp1 output is '0' or '1', respectively.

The received signal r is initially routed across dual band-pass filters sub-module to retrieve non-silent signals. The outputs of the two bandpass filters are applied to the two envelope detectors to obtain the estimated amplitude levels e_1 and $e_{\bar{1}}$, which correspond to the ON state symbols 1 and $\bar{1}$, respectively. In terms of system behaviour, when there is no noise ($\eta = 0$), either e_1 or $e_{\bar{1}}$ will have a non-zero value depending on the appropriate ON state symbols. Similarly, if a silent symbol (0) emerges within the interim symbol time, e_1 and $e_{\bar{1}}$ will become zero.

For appropriate detection of all three encoded symbols in the received message signal e_1, $e_{\bar{1}}$ will be forwarded to a comparator sub-module to determine which of the two values is the largest. Furthermore, received signal also passed across threshold detectors sub-module to determine whether any of the signals e_1, $e_{\bar{1}}$ are greater than a predetermined threshold value e_{th}. Two threshold detector outputs i.e., q_1 and $q_{\bar{1}}$, are *ORed* to yield the value q. If the transmitter is silent, the outputs of the threshold detectors q_1 and $q_{\bar{1}}$ are considered to be zero, making the *ORed* output to be zero, so that except at the output terminal S_0 output port, other outputs i.e., S_1 (or $S_{\bar{1}}$ will also be zero. Whereas, if the transmitter in ON state, then any of the two outputs S_1 (or $S_{\bar{1}}$ will become high depending on the output of the comparator, i.e., largest values among e_1, $e_{\bar{1}}$. The detailed process is shown in fig. 4. Following the identification of encoded symbols, the incoming data block is eventually decoded using the appropriate decoding Algorithm.3 Detailed description of the error analysis and energy calculation is described by Majumder et al. [41].

A. PERFORMANCE IN A NOISY CHANNEL

Next, we evaluate the performance of our proposed scheme in a realistic noisy environment. We assume that the channel noise follows an Additive White Gaussian Noise (*AWGN*) model. Referring to Fig. 3 depicting the hybrid non-coherent FSK-ASK receiver, the desired output $\hat{s}$ of the receiver can be expressed as:

1) When $q = 0$, the output $\hat{s}$ is equal to 0, indicating the presence of a silent symbol. This corresponds to the case where $S_0 = 1$ and both S_1 and $S_{\bar{1}}$ are equal to 0.
2) On the other hand, when $q = 1$ (representing an ON state symbol), the output $\hat{s}$ can be determined as

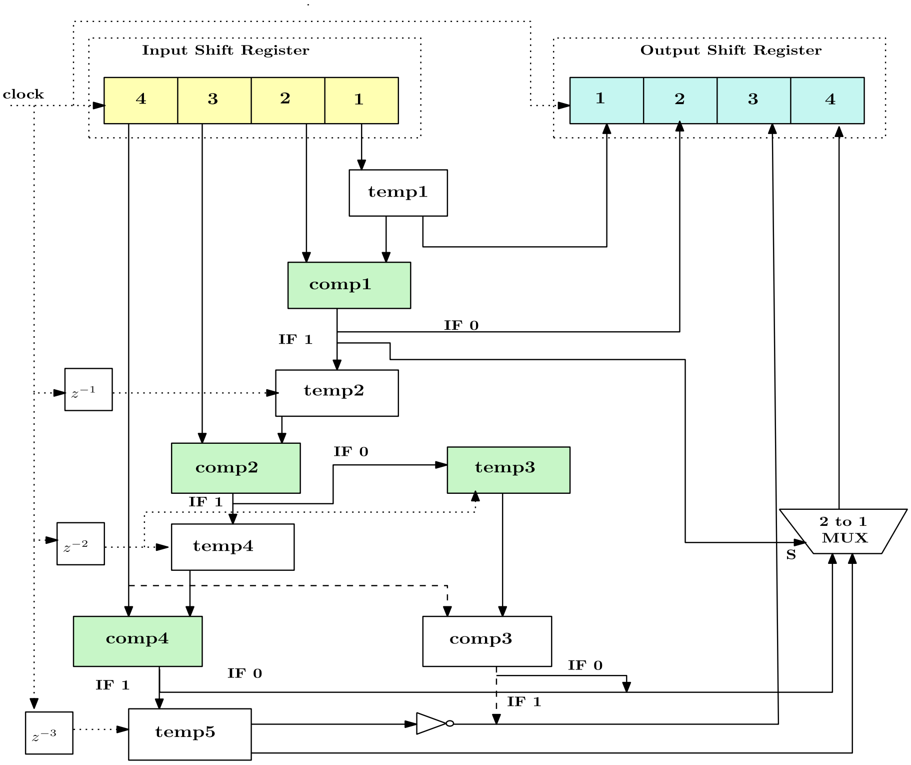


FIGURE 5. Schematic diagram of circuit for symbol run detection.

follows:

$$\hat{s} = \begin{cases} 1, & e_1 > e_{\bar{1}},\ (S_1 = 1, S_{\bar{1}} = 0). \\ \bar{1}, & e_{\bar{1}} > e_1,\ (S_{\bar{1}} = 1, S_1 = 0) \end{cases} \tag{15}$$

B. ERROR ANALYSIS

To assess the bit-error rate (BER), we assume that the transmitted data frame consists of *d Dual Run Distribution based Encoding* (DRDE) digits, where each digit can take one of three values: 0, 1, or $\bar{1}$. However, instead of directly calculating the conventional bit-error rate during message reception, we adopt a frame error rate (FER) approach, as utilized in [29], to derive the BER value. The reason behind this choice is that *Dual Run Distribution based Encoding* (DRDE) symbols exhibit some inter-dependency due to the nature of $\bar{1}$. Consequently, the erroneous reception of a $\bar{1}$ may affect several higher-order bits in the decoded binary data. Thus, FER serves as a more suitable performance metric for error analysis in RBNS-based number systems.

In the hybrid FSK-ASK modulation scheme, we define the probability α as the likelihood that a transmitted *Dual Run Distribution based Encoding* (DRDE) symbol is 0 and is correctly received as the symbol 0. Similarly, we define β (or δ) as the probabilities that the transmitted symbol is 1 (or $\bar{1}$) and the received symbol is also 1 (or $\bar{1}$) respectively. It is important to note that β and δ are symmetrical, meaning that $\beta = \delta$.

$$\begin{aligned} \alpha &= P\{q_1 = 0 \mid 0\}P\{q_{\bar{1}} = 0 \mid 0\} \\ &= \Big[P\{q_1 = 0 \mid 0\}\Big]^2, \end{aligned} \tag{16}$$

(by symmetry), where P denotes the probability. It is worth noting that both $P\{q_1 = 0 \mid 0\}$ and $P\{q_{\bar{1}} = 0 \mid 0\}$ follow the Rayleigh distribution. Assuming a zero mean with a standard deviation of σ for the *AWGN* channel, we can express this as follows:

$$\begin{aligned} P\{q_1 = 0 \mid 0\} &= \int_0^{e_{th}} \left(\frac{e_1}{\sigma^2}\right) \cdot e^{\left(-\frac{e_1^2}{2\sigma^2}\right)} de_1 \\ &= \left(1 - e^{\left(-\frac{e_{th}^2}{2\sigma^2}\right)}\right) .u(e_1) \end{aligned} \tag{17}$$

Hence, we have,

$$\alpha = \left(1 - e^{\left(-\frac{e_{th}^2}{2\sigma^2}\right)}\right)^2 \tag{18}$$

Now, β can be defined as,

$$\begin{aligned} \beta &= P\{q = 1 \,\&\, e_1 > e_{\bar{1}}\} \\ &= P\{q = 1 \mid 1\} \cdot P\{e_1 > e_{\bar{1}} \mid 1, \forall e_1 \mid 1\} \\ &= X_1 X_2, \end{aligned} \tag{19}$$

where $X_1 = P\{q = 1 \mid 1\}$ and $X_2 = P\{e_1 > e_{\bar{1}} \mid 1, \forall e_1 \mid 1\}$. Expanding the expression for X_1 we get,

$$\begin{aligned} X_1 &= P\{q = 1 \mid 1\} \\ &= \int_0^{\infty} \Big[(1 - P\{q_1 = 0 \mid 1\} P\{q_{\bar{1}} = 0 \mid 1\}) p_1(e_1) \Big] de_1 \end{aligned} \tag{20}$$

In Equation (20), $P\{q_1 = 0 \mid 1\}$ follows a *Rician distribution*, whereas $P\{q_{\bar{1}} = 0 \mid 1\}$ follows a *Rayleigh distribution*.

$$P\{q_1 = 0 \mid 1\} = \int_0^{e_{th}} \left\{ \left(\frac{e_1}{\sigma^2}\right) \cdot e^{\left(-\frac{e_1^2 + s_1^2}{2\sigma^2}\right)} \cdot I_0\left(\frac{e_1 . s_1}{\sigma^2}\right) \right\} de_1 \tag{21}$$

and

$$\begin{aligned} P\{q_{\bar{1}} = 0 \mid 1\} &= \int_0^{e_{th}} \left(\frac{e_{\bar{1}}}{\sigma^2}\right) \cdot e^{\left(-\frac{e_{\bar{1}}^2}{2\sigma^2}\right)} de_{\bar{1}} \\ &= 1 - e^{\left(-\frac{e_{th}^2}{2\sigma^2}\right)} \end{aligned} \tag{22}$$

Substituting the values of $P\{q_1 = 0 \mid 1\}$ and $P\{q_{\bar{1}} = 0 \mid 1\}$ from eqns. (21) and (22) in eq. (20), we can then compute X_1. To compute the value of X_2 we note that,

$$X_2 = \int_0^{\infty} \Big[P\{e_1 > e_{\bar{1}} \ \mid 1\} \Big] p_1(e_1) de_1 \tag{23}$$

Thus, we have,

$$X_2 = \int_0^{\infty} \left[\left(1 - e^{\left(-\frac{e_1}{\sigma^2}\right)}\right) \left(\frac{e_1}{\sigma^2}\right) e^{\left(-\frac{e_1^2 + s_1^2}{2\sigma^2}\right)} I_0\left(\frac{e_1 s_1}{\sigma^2}\right) \right] de_1 \tag{24}$$

Let P_0, P_1, and $P_{\bar{1}}$ represent the probabilities of occurrences of symbols 0, 1, and $\bar{1}$, respectively, in the transmitted frame. Therefore, the probability that the transmitted *Dual Run Distribution based Encoding* (DRDE) frame with $d' = d+1$ digits will be received correctly by our proposed receiver scheme can be expressed as the *Frame Correct Rate* (*FCR*), given by:

$$FCR = \alpha^{d'P_0} \times \beta^{d'P_1} \times \delta^{d'P_{\bar{1}}} \tag{25}$$

Since $\beta = \delta$, we can rewrite eq.(25) as

$$FCR = \alpha^{d'P_0} \times \beta^{d'(P_1 + P_{\bar{1}})} \tag{26}$$

It may be noted that the frame error rate (FER) for the transmitted message is represented by $FER = (1 - FCR)$.

Now consider the situation that we transmit the original binary message of d bits with a BER of ϱ. Then the probability of receiving the whole message frame containing d bits without error is given $(1 - \varrho)^d$. Hence, the frame error rate (FER) for binary FSK can be represented as $FER = \{1 - (1 - \varrho)\}^d$, which can be roughly estimated as $d\varrho$, since ϱ is usually very small. Therefore, $BER \approx FER/d$. To compute BER in our proposed hybrid modulation scheme, we will set signal amplitude s_A to 1, i.e., signal power will be $P_{s_A} = \frac{1}{2}s_A^2 = 1/2$ and hence the signal-to-noise ratio (SNR) will be equal to $\frac{1}{2\sigma^2}$. We compute the optimum value of threshold voltage e_{th} using Mathematica that provides the minimum possible BER for a given *SNR* value. This optimum e_{th} is then used in eq.(26) to find the required scaled average transmitter power with our proposed scheme to meet a given value of BER. On the other hand, it may be noted that the BER and *SNR* values for BFSK modulation are related as $\varrho = \frac{1}{2}e^{-\varphi/2}$, where *SNR* in dB is represented as $SNR_{dB} = 10\log_{10}\varphi$.

C. EFFECT OF DATA COMPRESSION ON ENERGY SAVINGS

Our proposed encoding scheme results in a substantial reduction in the number of transmitted bits due to the compressed biased exponent field used in each of the encoding algorithms. This compression leads to a decrease in the overall time required for transmitting and receiving the data bits, resulting in additional energy savings for the transmitter and receiver. The estimation of these energy savings is provided below:

Let $\mathcal{T}_B$ and $\mathcal{E}_B$ be the transmission time and transmitter energy for the conventional BFSK technique to transmit the individual raw data set, and let $\mathcal{T}_E$ and $\mathcal{E}_E$ denote the corresponding transmission time and transmitter energy required for the proposed FSK-ASK technique to transmit the encoded correlated data set. Let $\mathcal{T}_E/\mathcal{T}_B = \zeta$. Noting that $\mathcal{E}_B = P_B \times \mathcal{T}_B$ and $\mathcal{E}_E = P_E \times \mathcal{T}_E$, we get,

$$\frac{\mathcal{E}_E}{\mathcal{E}_B} = \frac{P_E}{P_B} \times \frac{\mathcal{T}_E}{\mathcal{T}_B} = \frac{P_E}{P_B} \times \zeta \tag{27}$$

Thus, for $\zeta < 1$, there will be saving in transmitter energy compared to the conventional transmission of raw data using BFSK.

Example 5: As an illustrative example, we consider the temperature dataset 1 taken from the SENSEnuts motes [45]. Temperature data shows high temporal correlation with 91.5% *of 0's in the encoded data with our proposed scheme. That indicates,* $P_0 = 91.5\%$ *for this data set. Table 2 shows the SNR values using the proposed hybrid FSK-ASK modulation scheme with encoded data and also using the conventional BFSK with the raw sensor data for different BER values in the range from* 10^{-4} *to* 10^{-8}*. The optimum threshold values* e_{th} *found for the individual cases are also shown. These SNR values given in the table 2 from which we find that on an average, our proposed hybrid FSK-ASK scheme requires a peak SNR value nearly* 2.562 *dB more than that with BFSK for the same BER value. However, considering the fact that the transmitter is silent for* 91.5% *time during the period of 0's, i.e., it is ON only for* 8.5% *time, and noting that* $10\log_{10} 0.085 = -10.70$ *dB, it turns out that the average SNR value with the encoded data using FSK-ASK is less than that with raw data using BFSK by* $10.70 - 2.562 = 8.143$ *dB, which corresponds to* $\frac{P_E}{P_B} = 10^{-8.143/10} = 0.1533$*, where* P_B *and* P_E *are the average transmitter powers with BFSK applied on individual data*

TABLE 2. Peak *SNR* values for FSK-ASK with DRDE encoding and BFSK with raw data from Dataset1.

SNR in *dB* with FSK-ASK using Encoded Data	BER Value	Corresponding Optimum Value of (e_{th})	*SNR* in *dB* with BFSK Using Raw Data
14.00	4.3012×10^{-4}	0.60	11.67
15.00	1.4169×10^{-4}	0.59	12.46
16.00	3.0026×10^{-5}	0.57	13.41
17.00	8.8588×10^{-7}	0.55	14.36
18.00	7.8263×10^{-8}	0.54	15.29

elements and the proposed FSK-ASK technique applied on block encoded correlated data, respectively.

Example 6: Continuing with the encoded values of temperature dataset used in Example 5, we find that the average size of packets containing the encoded data is 78.8% *of the original packet size which corresponds to* $\frac{\mathcal{T}_E}{\mathcal{T}_B} = 0.788$. *Hence, from eq.(27) we have,*

$$\frac{\mathcal{E}_E}{\mathcal{E}_B} = \frac{P_E}{P_B} \times \frac{\mathcal{T}_E}{\mathcal{T}_B} = 0.1533 \times 0.78 = 0.1198$$

So, $\mathcal{E}_E = 0.1198 \times \mathcal{E}_B$ *which implies* 88.01% *savings of transmitter energy savings over the conventional technique using BFSK.*

Analyzing the above simulation result, the following inference can be drawn:

Theorem 1: In an AWGN environment, using proposed encoding scheme DRDE, that produces very high frequent silent symbol 0 *other than other two non-silent encoding symbols i.e., 1 and* $\bar{1}$. *Thus, the deep-sleep period (%) of transmitter is greatly enhanced. Using the silent symbol based communication and hybrid ASK-FSK modulation technique, our encoding scheme consumes approximately* 11.98% *transmitter energy as compared to binary FSK, resulting* 88.01% *of energy savings at the transmitter side. Due to average compression rate of 0.22, energy savings is achieved at the receiver side is about 12% compared to BFSK.*

VII. SIMULATION ON REAL-LIFE SENSOR DATA

In this section, we present the simulation results of our proposed technique using three different real-life sensor datasets. We compare the performance of our proposed technique with existing techniques in terms of transmitter and receiver energy savings as well as transmission time.

A. SIMULATION WITH DATASET 1

The experiment involved the use of 13 SENSEnuts motes [45], equipped with Atmel SAM4S16C ARM Cortex-M4 MC processor, wherein one mote functioned as the PAN (Personal Area Network) coordinator while the remaining 12 acted as coordinators. The PAN coordinator was connected to the computer through a USB cable, and the SENSEnuts motes operated on the Zigbee protocol based on the IEEE 802.15.4 standard. These motes use the license-free 2.4 GHz ISM frequency band, offering up to 250 Kbps data rate. The coding for the experiment was done using C and Matlab in the SENSEnuts GUI. The PAN coordinator, acting as a sink node, received data from the coordinators and processed it to make necessary decisions, operating in PULL and PUSH modes. In this study, it was assumed that the signal transmission pattern of the sensors was omni-directional. Over a period of 10 days and observation frequency of 1 minute, 14360 temperature data was collected using each HTP Sensor Module.

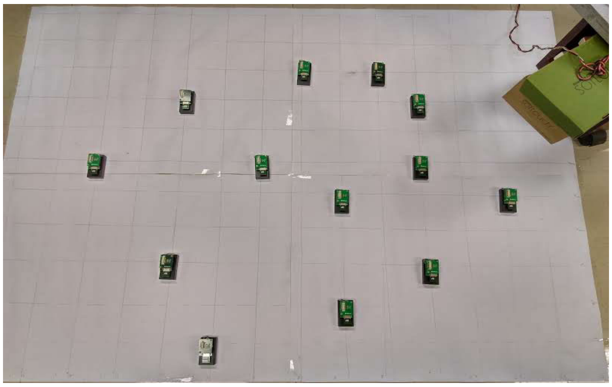


FIGURE 6. Sensor Real Testbed with SENSEnuts Motes.

B. SIMULATION WITH DATASET 2

This dataset is obtained from the Activity 2.3 CityPulse EU FP7 project [46], which focuses on real-time IoT stream processing and large-scale data analytics for smart city applications. The dataset consists of air pollution data collected by various types of sensors deployed over a specific region, and it includes the following details:

- *Five types of sensors measuring air pollution:* The dataset includes measurements from sensors for sulphur dioxide (SO_2), carbon monoxide (CO), nitrogen dioxide (NO_2), particulate matter (PM), and ozone (O_3).
- *Geographical location:* The data is collected from Hinnerup, Denmark. The latitude ranges from 56.1257N to 56.2349N, and the longitude ranges from 10.1050E to 10.2462E.
- *Number of sensors:* There are 449 sensors of each type deployed in the area, and the observations were made within the following time frame:
    - *Duration:* From 01-08-2014, 00:05 hrs to 01-10-2014, 00:00 hrs.
    - *Observation Frequency:* Data was recorded every 5 minutes.
    - *Total number of data per node:* Each sensor collected a total of 17,568 data points.

Table 3 shows the overall energy E_N expenditure by the wsn in mWh range, percentage energy savings and equivalent CO_2 footprint (in mg/day) with our proposed encoding scheme on the dataset 2 using CC1100 [33] (operating at 10 dBm with I_{HIGH} = 30.3 mA) and CC2420 [34] (operating at 0 dBm with I_{HIGH} = 17.4 mA), Maxim 2820 [35], TR 1000 [36] transceivers. The detail radio specification is given in Table 1. The overall energy consumed by a sensor node is a combination of *base energy*, *transmission energy* and *switching energy* multiplied by total number of sensors deployed in a region, duty cycle and packet transmitter rate of the transceiver. It is observed that the overall CO_2 footprint generation lies in the range between 1.48-0.041 mg/day which is 78% lesser than conventional BFSK modulation scheme.

1) PERFORMANCE COMPARISON BASED ON CO_2 FOOTPRINT GENERATION

We now compare the effectiveness of our proposed *Dual Run Distribution based Encoding* (DRDE) scheme with two efficient data reduction and aggregation technique that is CNS [37] and PCA [38]. Comprehensive simulation results with City Pulse dataset [46] based on CO_2 emission rate is shown Table 7 using two low-cost and low-data rate transceiver CC1100 [33] and CC2420 [34] chips operating at 10 dBm and 0 dBm, respectively. Table 7 depicts the generation of CO_2 footprint with DRDE scheme which is around 47% and 70% lesser with respect to CNS [37] and PCA [38] schemes, respectively.

2) TEST SUITE ANALYSIS

The encoding scheme possesses significant potential for reducing energy consumption during both transmission and reception of data in wireless sensor networks (WSNs). Analysis of the encoded data reveals that the frequency of the Null symbol (0) is typically between 75-80% of the total encoded data. This high proportion of Null symbols enables the transceiver to remain in a ‘‘silent’’ state for the duration of Null symbol transmission, which is considerably longer than the transmission period of the other two non-silent symbols (1 and $\bar{1}$). Table 6 gives the energy savings profile for multiple datasets with respect to BFSK modulation.

C. EFFECT OF ENCODING

Proposed *Dual Run Distribution based Encoding* (DRDE) scheme is highly effective in reducing transmission energy by compressing encoded data and increasing silent symbols in the message. This results in an increase in silent communication, which is a low-power deep sleep mode of operation for a sensor node. For the purpose of our experiments, we prepared a test suite consisting of more than 200 files [47], categorized in different sensor applications and comprising multiple representative files, which we usually encounter in our day-to-day life in different real-life applications. Table 4 gives the detail description of various files used in test suite. The relative frequencies corresponding to each $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$ for a specific file type is obtained by dividing the total occurrences of exactly i_{k_1} number of runs of 1's of length k_1, $1 \leq k_1 \leq n$ and i_{k_2} number of runs of 10's of length k_2, $1 \leq k_2 \leq n$ by the total number of frames required to read the entire set of files. The probability of occurrences of i_{k_1} number of runs of 1's of length k_1, $1 \leq k_1 \leq n$ and i_{k_2} number of runs of 10's of length k_2, $1 \leq k_2 \leq n$ in a given binary string of equally likelihood of all possible combinations of length n is given by $\frac{N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}}{2^n}$. Table 5 presents the result of our experiment on Actuaries Climate Index Data for n = 8 as percentage value. All simulations were done using MATLAB. The *root mean square* (RMS) deviation in percentage values of $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$ from theoretically calculated values are derived for n = 8 and it appears that, the deviation in percentage values of $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$ for different values of $(i_{k_1},k_1)(i_{k_2},k_2)$ always lies in within ±4.5% for all types of files encountered in test suite. Figure 7 depicts the distribution comparison between theoretical and encoded version of Actuaries Climate Index for n = 8 as percentage value. The fact that the DRDE encoding scheme maintains an error margin of only ±4.5% for all types of files in the test suite confirms its lossless encoding capability.

D. PERFORMANCE MEASURES OF DATA COMPRESSION

The assessment of data compression techniques encompasses the evaluation of the subsequent metrics:

- Compression Ratio ($\mathcal{C}_{\text{ratio}}$): This metric quantifies the efficiency of compression algorithms. It is defined as the ratio of the size of the uncompressed data in bits (B_{uc}) to the size of the compressed data in bits (B_{c}), expressed as $\mathcal{C}_{\text{ratio}} = \frac{B_{\text{uc}}}{B_{\text{c}}}$.
- Compression Time (T_{comp}) Requirement: This metric measures the duration required to compress the original data.
- CPU Power Utilization (in mJ): This involves examining the impact of data compression on CPU resource usage or power consumption, with a focus on energy efficiency.
- Overall Transmission Cost (in mJ): This metric quantifies the energy expended for transmitting compressed data over a network.

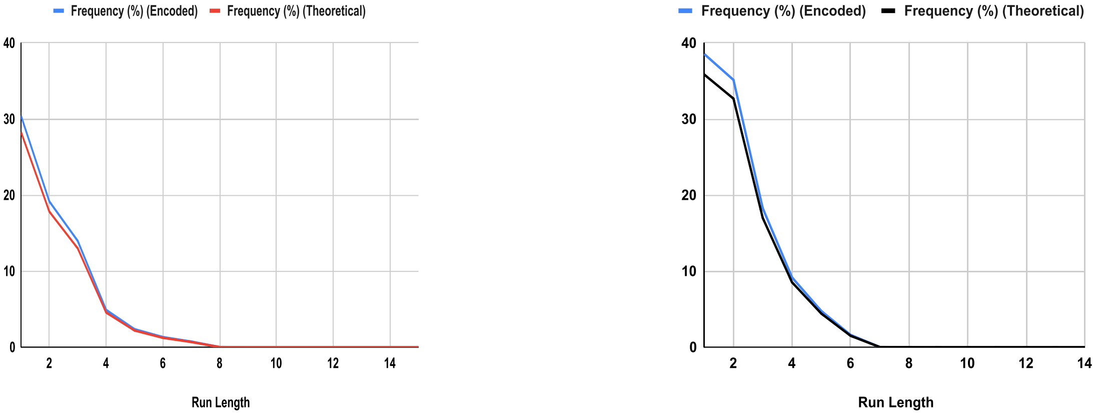
In this section, famous real-world data [48] obtained from 54 sensors deployed within the Intel Berkeley Research lab (illustrated in Fig. 9) has been employed to assess the compression ratios of different existing algorithms. The dataset contains 2.3 million readings of various physical parameters e.g., humidity, temperature, light, and voltage, gathered at intervals of 31 seconds. This identical dataset has been utilized to quantify the complete energy consumption of the transceiver for a single packet transmission of variable sizes, ranging between 128 kB to 8 Mb. The algorithms were executed on an LPC2148 microcontroller that is built upon a 16-bit/32-bit ARM7TDMI chipset, depicted in Fig. 8. The system accommodates on-chip static RAM ranging from 8 kB to 40 kB and on-chip flash memory spanning from

TABLE 3. Overall energy savings and equivalent CO_2 emission footprint (per day) for the Dataset 2.

Performance Comparison					
with CC1100 [33]			with CC2420 [34]		
E_{wsn} (mWh)	(%) Savings	CO_2 footprint (mg/day)	E_{wsn} (mWh)	% Savings	CO_2 footprint (mg/day)
0.092	76.45	0.082	0.0502	79.48	0.0451
with Maxim 2820 [35]			with TR 1000 [36]		
E_{wsn} (mWh)	% Savings	CO_2 footprint (mg/day)	E_{wsn} (mWh)	% Savings	CO_2 footprint (mg/day)
1.64	75.32	1.48	0.0458	77.65	0.041

TABLE 4. Description of test suite.

SL. No.	Dataset	Source	Dataset Description			
			No. of Data Instances	Dataset Characteristics	Attributes Characteristics	Spatial Attributes
1	Actuaries Climate Index	Canada & USA	78097	Time-Series	Real	6
2	Motion Sensor	Queen Mary University, London, UK	10293	Time-Series	Real	9
3	3D Road Network	Aarhus University, Denmark	434874	Sequential	Real	4
4	Air Quality	ENEA, Italy	9358	Multivariate, Time-Series	Real	15
5	Online News Popularity	Universidade do Porto, Portugal	39797	Multivariate	Integer, Real	61
6	Gas Sensor	UC, San Diego, USA	180	Multivariate, Time-Series	Real	15
7	Parkinson Disease	Istanbul University, Turkey	77	Multivariate	Integer	7
8	Ozone Level Detection	University of Houston, USA	2536	Multivariate, Time-Series	Real	73
9	Wireless Indoor Localization	IIT-D, India	2000	Multivariate	Real	7
10	Crowdsourced Mapping	IGES, Japan	10546	Multivariate	N/A	29
11	Airfoil Self-Noise	Intelnics, NASA	1503	Multivariate	Real	6
12	Maisbich Temperature Sensor	Luxemburg	10005	Sequential, Time-Series	Real	2



k_1, k_2 = **4, 4**

k_1, k_2 = **3, 3**

FIGURE 7. $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$ for $n = 8$ distribution comparison between Theoretical vs Encoded actuaries climate index data.

32 kB to 512 kB. The CPU operating voltage (V_D) spans from 3.0 V to 3.6 V, and the clock frequency (f_{clk}) operates between 1 to 25 MHz. Notably, the active mode (I_A) and transmission current (I_{tx}) values are 30 mA and 100 mA, respectively. The computation power consumption, known as active state power consumption per clock cycle within the LPC2148 microcontroller system can be computed as $E_{active} = \frac{V_D \times I_A}{f_{clk}}$. Similarly, the energy consumption due to packet transmission can be computed as $E_{tx} = \frac{V_D \times I_{tx}}{R_{data}}$, considering a data transmission rate (R_{data}) of 2.5 Kbps. We have computed E_{active} and E_{tx} for LPC2148 microcontroller with ARM7TDMI chipset using Intel lab dataset [48], i.e., $E_{active} = 9nJ$ and $E_{tx} = 43nJ$, respectively.

Table 8 provides an illustration of how fundamental CPU activity cycles can be harnessed to signify the energy expenditure linked to data transportation. We have enumerated the cumulative essential CPU cycles linked to each data compression approach, such as NCS [22], DSC [23], [24], ALDC [25], and RMS-DSC [26], and subsequently compared them against our proposed DRDE technique.

- Our proposed encoding method requires a total of 12 addition (ADD) operations, 7 move (MOV) operations, 13 load (LDR) operations, 4 store (STR) operations, 10 subtraction (SUB) operations, and 2 subtraction with flags (SUBS) operations. It also involves 30 comparison (CMP) operations, 10 branching/conditional

TABLE 5. Relative frequency distribution of $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$ in actuaries climate index data (%) with $n = 8$.

Runlength Size		Percentage Values of $N_n^{(i_{k_1},k_1)(i_{k_2},k_2)}$			
		$k_1 = 1$	$k_1 = 2$	$k_1 = 3$	$k_1 = 4$
		$k_2 = 1$	$k_2 = 2$	$k_2 = 3$	$k_2 = 4$
1	k_1	1.209	8.0298	26.3862	38.5628
	k_2	-	4.5281	12.2593	30.4521
2	k_1	-	2.5534	6.7535	35.1591
	k_2	-	1.4398	3.0778	19.1753
3	k_1	-	0.8119	2.4806	18.2753
	k_2	-	0.5304	1.4023	13.9838
4	k_1	-	0.0305	0.8616	9.1807
	k_2	-	-	0.2991	4.9027
5	k_1	-	-	0.0563	4.7630
	k_2	-	-	0.0186	2.3701
6	k_1	-	-	-	1.6212
	k_2	-	-	-	1.3365
7	k_1	-	-	-	0.0223
	k_2	-	-	-	0.7418
8	k_1	-	-	-	0.0044
	k_2	-	-	-	-

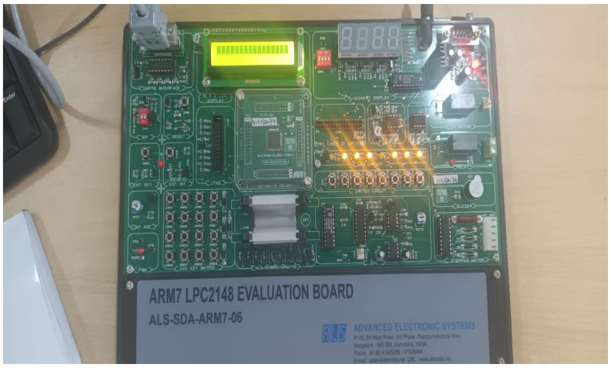


FIGURE 8. LPC2148 evaluation board.

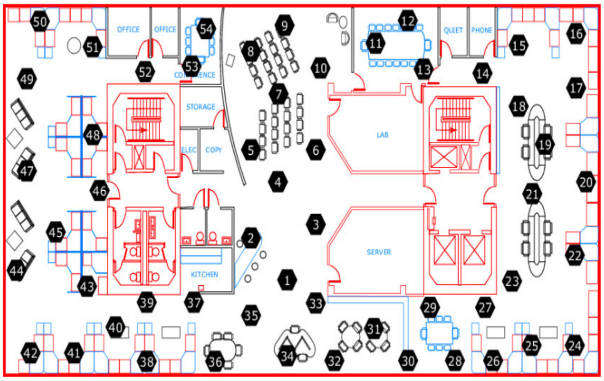


FIGURE 9. Sensor deployment setup at Intel Lab.

operations (e.g., BNE), and 14 bit shifting (LSL) operations, as outlined in Algorithm 1. Consequently, the DRDE method employs a total of 143 CPU cycles, which is equivalent to $143 \times 9 \times 10^{-9} = 1.29\mu J$. The transmission energy cost per packet of size 128 kB is calculated as $128 \times 10^3 \times \mathcal{C}_{\text{ratio}} \times 43 \times 10^{-9} = 1.21\ mJ$.

- The NCS [22] method requires 20 addition operations, 10 subtraction operations, 12 division operations, 7 load/store operations, 2 multiplication operations, 5 shifting operations, 5 Ex-OR operations, 20 branching operations, and 12 comparison operations for encoding data in ARM7 assembly language. This corresponds to 725 CPU cycles.
- The DSC [23], [24] method necessitates 28 addition operations, 7 division operations, 15 subtraction operations, 25 load/store operations, 8 multiplication operations, 16 branching operations, and 25 comparison operations for encoding data in ARM7 assembly language. This translates to 749 CPU cycles.
- The ALDC [25] method involves 35 addition operations, 17 subtraction operations, 5 division operations, 12 load/store operations, 2 multiplication operations, 2 Ex-OR operations, 15 shifting operations, 12 branching operations, and 21 comparison operations for encoding data in ARM7 assembly language. This results in 464 CPU cycles.
- The RMS-DSC [26] method requires 5 division operations, 18 addition operations, 28 subtraction operations, 8 load/store operations, 30 multiplication operations, 16 branching operations, and 55 comparison operations for encoding data in ARM7 assembly language. This corresponds to 392 CPU cycles.

The comprehensive comparative analysis, focusing on both CPU power consumption and transmission cost, is presented in Table 9. It is evident from the results in Table 9 that the DRDE method outperforms other data compression techniques in terms of CPU power and transmission cost. The simulation results are based on Intel sensor data [48] with a packet size of 128 Kb. In Table 9, the first column, labeled 1-$\mathcal{C}_{\text{ratio}}$, illustrates the reduction in transmitted data length achieved by applying the data compression algorithm. Conversely, the compression time is computed by summing up the CPU cycles of all ARM7 assembly instructions required for the implementation of specific encoding algorithms. The term "CPU power" refers to the power consumption of the device resulting from the processing of the corresponding CPU cycles in the active mode. Lastly, the "transmission cost" (in mJ) is the expense associated with transmitting a compressed bit within a single packet of size 128 kB at a data rate of 2.5 Kbps. The observations clearly indicate that our proposed DRDE encoding scheme demands 77%, 64.4%, 67.6%, and 48.8% fewer CPU cycles compared to NCS [22], DSC [23], [24], ALDC [25], and RMS-DSC [26] methods, respectively. Consequently, DRDE scheme consumes very less computation power which is about 1.29μJ per packet of size 128kb, which is 49.2%, 22.1%, 28.2%, 14% less compared with NCS [22], DSC [23], [24], ALDC [25], RMS-DSC [26] schemes, respectively. Moreover, the DRDE scheme outperforms in terms of transmission energy cost with respect to other data encoding schemes.

TABLE 6. Energy savings profile of test suite.

Dataset	DRDE			BINARY		Energy Savings (%)	
	0 (%)	1 (%)	$\bar{1}$ (%)	0 (%)	1 (%)	Transmitter	Receiver
Actuaries Climate Index	68.28	20.63	11.09	56.32	43.68	75.54	16.12
Motion Sensor	71.68	17.42	11.19	46.20	53.79	78.03	18.78
3D Road Network	76.05	17.62	6.31	56.21	43.76	77.44	18.84
Air Quality	69.31	18.56	12.08	63.76	36.23	73.73	13.03
Online News Popularity	68.02	19.93	12.05	48.27	51.72	83.73	13.04
Gas Sensor	69.27	20.48	8.87	53.45	46.55	78.4	21.94
Parkinson Disease	84.63	10.57	4.78	77.38	22.61	90.8	19.48
Ozone Level Detection	61.95	23.26	14.78	55.16	44.83	69.56	13.94
Wireless Indoor Localization	87.25	7.63	5.10	80.99	19.01	78.91	14.9
Crowdsourced Mapping	61.49	26.79	11.71	54.50	45.94	70.28	6.65
Airfoil Self-Noise	68.67	18.27	13.06	54.50	45.94	80.59	22.05
Maisbich Temperature Sensor	72.32	17.21	10.45	56.54	43.45	81.10	24.11

TABLE 7. CO_2 emission from coal.

Dataset Type	Energy Reduction Technique	CO_2 Footprint (kg/day)	
		using CC1100	using CC2420
Dataset 2	DRDE	0.082	0.054
	CNS	0.156	0.097
	PCA	0.271	0.166

TABLE 8. CPU cycles for multiple operations in LPC2148 Microcontroller using ARM7TDMI Chipset.

Operations	No. of CPU Cycles
ADD	2
CMP	1
EOR	2
MOV	1
MUL	32
SUBS	2
SUB	1
DIV	45
LDR	2
BNE	2
LSL	1

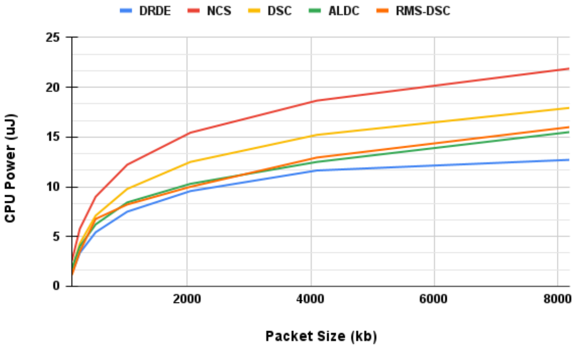


FIGURE 10. Variation of CPU Power (μJ) with Packet Size (kb).

We have conducted further investigations to assess the efficacy of our proposed scheme across three distinct categories:

1) *CPU Power vs Packet Size*: Figure 10 illustrates the relationship between the packet size and the overall CPU computational power. The experiment considers a maximum packet size of 8.1 Mb. For processing this data on the LPC2148 microcontroller, a slight delay of 40 ms is introduced after each data stream segmentation, which subsequently elongates the overall computation time. The figure demonstrates that the computational power consumption of our proposed scheme remains superior to existing schemes even for larger packet sizes.
2) *Transmission Cost vs Packet Size*: The performance of the DRDE scheme in terms of transmission power across varying packet sizes is showcased in Figure 11.
3) *Compression Ratio vs Block Size*: The compression ratio plays a pivotal role in conserving energy, impacting both computational and transmission consumption. Consequently, we have evaluated the fluctuation of the compression ratio with changing block size (r),

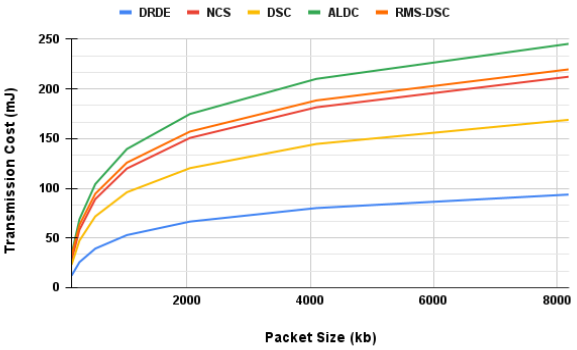


FIGURE 11. Variation of transmission cost (mJ) with Packet Size (kb).

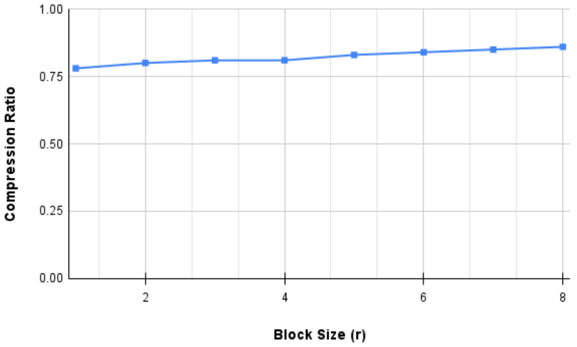


FIGURE 12. Variation of compression ratio with block size.

TABLE 9. Computational complexity analysis in terms of all performance parameters between various schemes.

Data Compression Schemes	1-$\mathcal{C}_{\text{ratio}}$	Compression Time (CPU Cycles)	CPU Power	Transmission Cost
DRDE	0.22	65	1.29 μJ	1.21 mJ
NCS [22]	0.52	285	2.56 μJ	2.75 mJ
DSC [23], [24]	0.42	183	1.67 μJ	2.31 mJ
ALDC [25]	0.61	201	1.809 μJ	33.6 mJ
RMS-DSC [26]	0.58	127	1.143 μJ	3.19 mJ

considering the values of k_1 and k_2 as $(4, 4)$, as depicted in Figure 12.

VIII. CONCLUSION

Attaining sustainability in low-power, low-cost monitoring applications in smart IoT-based WSNs requires reducing the redundant information transmission in wireless communication. The transmission energy can be minimized by increasing the use of silent symbols in the transmitted string, while retaining lower *root mean square* (RMS) deviation. Our proposed two-fold novel encoding scheme *Dual Run Distribution based Encoding* (DRDE) produces a biased distribution of encoded symbols with a significant proportion of 0s in each encoded message. These encoded data blocks are then utilized to generate data packets, which are subsequently sent by using hybrid FSK-ASK modulation/demodulation techniques to keep the transmitter silent during 0s. These simulation results show about 88% (theoretical) and 79-82% (practical) savings in transmitter and 12% (theoretical) and 23.5% (practical) energy savings at receiver over conventional BFSK with highly correlated real-life sensor dataset. Practical comparative analysis on energy savings are performed using low-cost low data rate radios like the CC1100, CC2420, Maxim 2820 and RFM TR 1000 with existing popular communication schemes like CNS and PCA. As a result, our suggested technique appears to be successful and novel in terms of energy-efficient communication and, as a corollary, reduced CO_2 emission footprint with generation of $1.48 - 0.041$ mg/day in WSN applications, which is 78% lesser than conventional BFSK modulation scheme. Furthermore, we investigate the influence of various data compression algorithms on computation time, CPU power consumption, and transmission cost on an LPC2148 microcontroller built upon a 16-bit/32-bit ARM7TDMI chipset. It makes the scheme environmentally sustainable and suitable for data-critical IoT applications, where data-dropping is not preferable, e.g., personal healthcare monitoring, video surveillance.

REFERENCES

[1] T. Barnett, S. Jain, U. Andra, and T. Khurana. (2023). *Cisco Visual Networking Index: Forecast and Trends, 2017–2023*. [Online]. Available: https://www.gsma.com/spectrum/wp-content/uploads/2013/03/Cisco_VNI-global-mobile-data-traffic-forecast-update.pdf

[2] T. Wu, J.-M. Redoute, and M. Yuce, "A wearable, low-power, real-time ECG monitor for smart T-shirt and IoT healthcare applications," in *Advances in Body Area Networks I*. Cham, Switzerland: Springer, 2019, pp. 165–173.

[3] X. Chen, J. Zhang, B. Lin, Z. Chen, K. Wolter, and G. Min, "Energy-efficient offloading for DNN-based smart IoT systems in cloud-edge environments," *IEEE Trans. Parallel Distrib. Syst.*, vol. 33, no. 3, pp. 683–697, Mar. 2022.

[4] A. S. Albahri, J. K. Alwan, Z. K. Taha, S. F. Ismail, R. A. Hamid, A. A. Zaidan, O. S. Albahri, B. B. Zaidan, A. H. Alamoodi, and M. A. Alsalem, "IoT-based telemedicine for disease prevention and health promotion: State-of-the-art," *J. Netw. Comput. Appl.*, vol. 173, Jan. 2021, Art. no. 102873.

[5] S. Adapa, S. M. Fazal-e-Hasan, S. B. Makam, M. M. Azeem, and G. Mortimer, "Examining the antecedents and consequences of perceived shopping value through smart retail technology," *J. Retailing Consum. Services*, vol. 52, Jan. 2020, Art. no. 101901.

[6] Q. Zhu, S. W. Loke, R. Trujillo-Rasua, F. Jiang, and Y. Xiang, "Applications of distributed ledger technologies to the Internet of Things: A survey," *ACM Comput. Surv.*, vol. 52, no. 6, pp. 1–34, Nov. 2020.

[7] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of Things," *IEEE Trans. Cloud Comput.*, vol. 6, no. 1, pp. 46–59, Jan. 2018.

[8] S. Misra and N. Saha, "Detour: Dynamic task offloading in software-defined fog for IoT applications," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 5, pp. 1159–1166, May 2019.

[9] S. Misra, A. Singh, S. Chatterjee, and M. S. Obaidat, "Mils-cloud: A sensor-cloud-based architecture for the integration of military tri-services operations and decision making," *IEEE Syst. J.*, vol. 10, no. 2, pp. 628–636, Jun. 2016.

[10] T. Wang, Y. Li, G. Wang, J. Cao, M. Z. A. Bhuiyan, and W. Jia, "Sustainable and efficient data collection from WSNs to cloud," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 2, pp. 252–262, Apr. 2019.

[11] A. Mukherjee, S. Misra, V. S. P. Chandra, and N. S. Raghuwanshi, "ECoR: Energy-aware collaborative routing for task offload in sustainable UAV swarms," *IEEE Trans. Sustain. Comput.*, vol. 5, no. 4, pp. 514–525, Oct. 2020.

[12] A. Boukerche, S. Guan, and R. E. De Grande, "A task-centric mobile cloud-based system to enable energy-aware efficient offloading," *IEEE Trans. Sustain. Comput.*, vol. 3, no. 4, pp. 248–261, Oct. 2018.

[13] G. Xie, Y. Chen, X. Xiao, C. Xu, R. Li, and K. Li, "Energy-efficient fault-tolerant scheduling of reliable parallel applications on heterogeneous distributed embedded systems," *IEEE Trans. Sustain. Comput.*, vol. 3, no. 3, pp. 167–181, Jul. 2018.

[14] V. Gupta and S. De, "Energy-efficient temporal sensing: An age-of-sample-based approach," *IEEE Internet Things J.*, vol. 9, no. 3, pp. 1806–1817, Feb. 2022.

[15] Y. P. Chen, D. Wang, and J. Zhang, "Variable-base tacit communication: A new energy efficient communication scheme for sensor networks," in *Proc. 1st Int. Conf. Integr. Internet Adhoc Sensor Netw.*, ACM, 2006, doi: 10.1145/1142680.1142715.

[16] Z. Wang, E. Bulut, and B. K. Szymanski, "Energy efficient collision aware multipath routing for wireless sensor networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2009, pp. 1–5.

[17] M. Čagalj, J.-P. Hubaux, and C. C. Enz, "Energy-efficient broadcasting in all-wireless networks," *Wireless Netw.*, vol. 11, nos. 1–2, pp. 177–188, Jan. 2005.

[18] J. Polastre, R. Szewczyk, and D. Culler, "Telos: Enabling ultra-low power wireless research," in *Proc. IPSN 4th Int. Symp. Inf. Process. Sensor Netw.*, Apr. 2005, pp. 364–369.

[19] M.-T. Hoang, N. Sugii, and K. Ishibashi, "A 1.36 μW 312–315 MHz synchronized-OOK receiver for wireless sensor networks using 65 nm SOTB CMOS technology," *Solid-State Electron.*, vol. 117, pp. 161–169, Mar. 2016.

[20] A. Sharma, A. Banerjee, and P. Sircar, "Performance analysis of energy-efficient modulation techniques for wireless sensor networks," in *Proc. Annu. IEEE India Conf.*, Dec. 2008, pp. 327–332.

[21] Y. Zhan and H. Dai, "Energy-based transmission strategy selection for wireless sensor networks," in *Proc. GLOBECOM IEEE Global Telecommun. Conf.*, Nov. 2005, pp. 1–5.

[22] J. Uthayakumar, M. Elhoseny, and K. Shankar, "Highly reliable and low-complexity image compression scheme using neighborhood correlation sequence algorithm in WSN," *IEEE Trans. Rel.*, vol. 69, no. 4, pp. 1398–1423, Dec. 2020.

[23] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. IT-19, no. 4, pp. 471–480, Jul. 1973, doi: 10.1109/TIT.1973.1055037.

[24] I. Csiszar, "Linear codes for sources and source networks: Error exponents, universal coding," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 4, pp. 585–592, Jul. 1982.

[25] K. L. Ketshabetswe, A. M. Zungeru, B. Mtengi, C. K. Lebekwe, and S. R. S. Prabaharan, "Data compression algorithms for wireless sensor networks: A review and comparison," *IEEE Access*, vol. 9, pp. 136872–136891, 2021.

[26] B. A. Lungisani, C. K. Lebekwe, A. M. Zungeru, and A. Yahya, "Image compression techniques in wireless sensor networks: A survey and comparison," *IEEE Access*, vol. 10, pp. 82511–82530, 2022.

[27] Y. Zhu and R. Sivakumar, "Challenges: Communication through silence in wireless sensor networks," in *Proc. 11th Annu. Int. Conf. Mobile Comput. Netw.*, Aug. 2005, pp. 140–147.

[28] Y. Zhou, L. Yang, L. Yang, and M. Ni, "Novel energy-efficient data gathering scheme exploiting spatial–temporal correlation for wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 2019, pp. 1–10, May 2019, doi: 10.1155/2019/4182563.

[29] K. Sinha, B. P. Sinha, and D. Datta, "An energy-efficient communication scheme for wireless networks: A redundant radix-based approach," *IEEE Trans. Wireless Commun.*, vol. 10, no. 2, pp. 550–559, Feb. 2011.

[30] A. Bhattacharya, P. Majumder, K. Sinha, B. P. Sinha, and K. Kavitha, "An energy-efficient wireless communication scheme using quint Fibonacci number system," *Int. J. Commun. Netw. Distrib. Syst.*, vol. 16, no. 2, pp. 140–161, 2016.

[31] P. Majumder, P. Chatterjee, and K. Sinha, "Run length distribution based block coding scheme for sustainable IoT applications," in *Proc. 2nd PhD Colloq. Ethically Driven Innov. Technol. Soc. (PhD EDITS)*, Nov. 2020, pp. 1–2.

[32] K. Sinha and B. P. Sinha, "On the distribution of runs of ones in binary strings," *Comput. Math. With Appl.*, vol. 58, no. 9, pp. 1816–1829, Nov. 2009.

[33] *CC1100 Datasheet*. Accessed: Mar. 10, 2023. [Online]. Available: https://www.ti.com/product/CC1100

[34] *CC2420 Datasheet*. Accessed: Apr. 25, 2023. [Online]. Available:https://www.ti.com/lit/ds/symlink/cc2420.pdf

[35] *Maxim2820 Datasheet*. Accessed: Mar. 10, 2023. [Online]. Available: https://www.maximintegrated.com/en/products/comms/wireless-RF/MAX2820.html

[36] *RFM TR1000 Datasheet*. Accessed: Apr. 8, 2023. [Online]. Available:https://pdf1.alldatasheet.com/datasheet-pdf/view/106155/RFM/TR1000.html

[37] K. Sinha, B. P. Sinha, and D. Datta, "CNS: A new energy efficient transmission scheme for wireless sensor networks," *Wireless Netw.*, vol. 16, no. 8, pp. 2087–2104, Nov. 2010.

[38] X. Zhang, H. Wu, Q. Li, and B. Pan, "An event-based data aggregation scheme using PCA and SVR for WSNs," in *Proc. IEEE 85th Veh. Technol. Conf. (VTC Spring)*, Jun. 2017, pp. 1–5.

[39] H. S. Wilf, *Generating Functionology*. Boca Raton, FL, USA: CRC Press, 2005.

[40] I. Buchanan, "Basic techniques of combinatorial theory," *J. Oper. Res. Soc.*, vol. 31, no. 1, p. 85, Jan. 1980.

[41] P. Majumder, S. R. Das, K. Sinha, and B. P. Sinha, "Sustainable operations in IoT by combining spatiotemporal data correlation with silent symbol based communication strategy," *IEEE Trans. Sustain. Comput.*, vol. 8, no. 1, pp. 133–152, Jan. 2023, doi: 10.1109/TSUSC.2022.3194506.

[42] C. K. Gasch, D. J. Brown, C. S. Campbell, D. R. Cobos, E. S. Brooks, M. Chahal, and M. Poggio, "A field-scale sensor network data set for monitoring and modeling the spatial and temporal variation of soil water content in a dryland agricultural field," *Water Resour. Res.*, vol. 53, no. 12, pp. 10878–10887, Dec. 2017.

[43] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in *Noise Reduction in Speech Processin*. Berlin, Germany: Springer, 2009, pp. 1–4.

[44] *Carbon Emission Calculator*. Accessed: Apr. 12, 2023. [Online]. Available: https://www.rensmart.com/Calculators/KWH-to-CO2

[45] *SENSEnuts*. Accessed: Jan. 25, 2023. [Online]. Available: https://www.eigen.in/pdf/sensenutsuniversity.pdf

[46] *Pollution Dataset*. Accessed: Feb. 10, 2023. [Online]. Available: https://mobcom.ecs.hs-osnabrueck.de/svn/CityPulse/Proposal/CityPulse_Proposal_final.pdf

[47] *UCI Machine Learning Repository*. Accessed: Mar. 15, 2023. [Online]. Available: https://archive.ics.uci.edu/ml/index.php

[48] *Intel Berkeley Research Lab*. Accessed: Feb. 5, 2023. [Online]. Available: http://db.csail.mit.edu/labdata/labdata.html

PRATHAM MAJUMDER (Member, IEEE) is currently pursuing the Ph.D. degree with the University of Calcutta, India. His Ph.D. work were carried out with the Advanced Computing and Microelectronics Unit, Indian Statistical Institute, Kolkata, India. His research interests include energy efficient wireless networks and wireless security.

PUNYASHA CHATTERJEE (Senior Member, IEEE) received the B.Tech., M.Tech., and Ph.D. degrees in information technology from the University of Calcutta, Kolkata, India, in 2003, 2005, and 2018, respectively. She has been an Associate Professor with the School of Mobile Computing and Communication, Jadavpur University, Kolkata, since 2012. Her research interests include ad-hoc networks, wireless sensor networks, the Internet of Things, and pervasive computing. She is a member of ACM.

SAURAV MALLIK (Member, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Jadavpur University, Kolkata, India, in 2017. His Ph.D. works were conducted with the Machine Intelligence Unit, Indian Statistical Institute, Kolkata. He is currently a Postdoctoral Fellow of environmental epigenetics with the Harvard T. H. Chan School of Public Health, Boston, MA, USA. Previously, he was a Postdoctoral Fellow with the School of Biomedical Informatics, The University of Texas Health Science Center, Houston, TX, USA; and the Division of Biostatistics, Department of Public Health Sciences, University of Miami Miller School of Medicine, Miami, FL, USA. He was a Research Associate with the Council of Scientific and Industrial Research, MHRD, Government of India, in 2017. He has coauthored more than 75 research articles with a Google H-index of 17. His research interests include computational biology, bioinformatics, data mining, bio-statistics, and pattern recognition. He was a recipient of the Emerging Researcher in Bioinformatics Award from the Bioclues BIRD Award Steering Committee, India, in 2020. He is an editor of many journals.

AMAL AL-RASHEED received the Ph.D. degree in information systems from King Saud University, in 2017. She is currently an Associate Professor with the Department of Information Systems, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University (PNU), Riyadh, Saudi Arabia. She has been involved in many projects related to learning technologies, cyber security, and virtual reality. Her contributions in research projects of academia led to the publication of papers in many journals and conferences. Her research interests include education, knowledge management, data mining, data analytics, cyber security, and natural language processing. In 2017, she received the Research Excellence Award by PNU for her publications during performing the Ph.D. degree.

MOHAMED ABBAS received the B.Sc. degree in electronics engineering from the Faculty of Engineering, Mansoura University, Egypt, in 1998, and the M.Sc. and Ph.D. degrees in computer engineering from Mansoura University, in 2002 and 2008, respectively. Since 2008, he has been an Assistant Professor with the Department of Communications and Computer Engineering, College of Engineering, Delta University. He is currently an Associate Professor with the Department of Electrical Engineering, King Khalid University, Abha, Saudi Arabia. His research interests include intelligent systems, medical informatics, nanotechnology, and bioinformatics.

MALAK SAEED M. ALQAHTANI received the B.Sc. degree in software engineering from the Faculty of Computing, Engineering and Media, De Montfort University, Leicester, U.K., in 2019. She is currently pursuing the master's degree with the College of Computer Science, King Khalid University, Abha, Saudi Arabia. Her research interest includes technology-related topics for mass communication module projects. Furthermore, she researches computer virus threats and the Y2K bug for the news feature writing module and for the social research module.

BEN OTHMAN SOUFIENE received the M.S. degree from the University of Monastir, in 2012, and the Ph.D. degree in computer science from Manouba University, in 2016, for his dissertation on secure data aggregation in wireless sensor networks. He was an Assistant Professor of computer science with the University of Gabes, Tunisia, from 2016 to 2023. His research interests include the Internet of Medical Things, wireless body sensor networks, wireless networks, artificial intelligence, machine learning, and big data.