



E-ISSN: 2707-6628
P-ISSN: 2707-661X
Impact Factor (RJIF): 5.56
www.computersciencejournals.com/ijcit
IJCIT 2025; 6(2): 138-142
Received: 02-07-2025
Accepted: 05-08-2025

HN Kama
Department of Industrial
Mathematics, Admiralty
University of Nigeria, Delta
State Nigeria

Thomas Kalu
Nigerian Defence Academy,
Kaduna, Nigeria

Oghenetega Avwokuruaye
Department of Cyber Security,
Admiralty University of
Nigeria, Delta State Nigeria

Corresponding Author:
HN Kama
Department of Industrial
Mathematics, Admiralty
University of Nigeria, Delta
State Nigeria

Operations research models for energy-efficient encoding in IoT networks

HN Kama, Thomas Kalu and Oghenetega Avwokuruaye

DOI: <https://www.doi.org/10.33545/2707661X.2025.v6.i2b.150>

Abstract

This paper develops an operations research framework for energy-efficient encoding in IoT networks. The encoding problem is modeled as an optimization task, where the objective is to minimize total energy cost comprising computational energy for encoding and transmission energy proportional to code length. Decision variables include encoding scheme selection and codeword assignment, subject to Kraft's inequality, latency, and memory constraints. A goal programming approach is used to balance energy efficiency and reliability. The model provides a mathematical foundation linking information theory and optimization, offering new insights into sustainable IoT data transmission. Arithmetic coding is closest to the entropy bound 100% efficient, Huffman coding achieves good efficiency 85-90% and Fixed-length encoding is least efficient 82%. Fixed-length encoding has the lowest total energy (3.5 units), since computation cost is minimal even though more bits are transmitted. Huffman coding requires more computation, raising its total energy slightly (3.9 units). Arithmetic coding saves transmission energy but has the highest computational cost, giving it the largest total energy (3.97 units).

Keywords: Energy-efficient encoding, operations research, optimization, IoT, huffman coding, kraft inequality, goal programming, stochastic modeling

Introduction

Piatkowski *et al.*, (2024) ^[12] implementation of several lightweight compressors on a typical microcontroller; measurement of encoding time, CPU energy, and radio transmission energy for representative datasets. Comparative experiments across data types, their work compression reduces transmitted bytes substantially for some data types, but the net energy benefit depends on the trade-off between CPU cost and transmission savings; for very small or high-entropy data the compression overhead can outweigh transmission savings. Costa *et al.*, (2024) ^[2] proposes two online, multivariate compression approaches tailored to TinyML and embedded systems that do not assume a fixed data distribution; designed to be computationally light and to work with streaming sensor data. Design of two algorithms based on data typicality/eccentricity analytics; evaluation on vehicular (OBD-II) datasets and embedded hardware profiling (execution time, memory). Both proposed methods achieve good compression ratios and low execution times compared with heavier compressors, making them suitable for on-device use in IoT. Guo *et al.*, (2024) ^[4] presents an architecture combining local (device/fog) preprocessing and cloud processing to reduce energy and latency in telehealth IoT. Encoding and compression at the edge is a key component to reduce uplink cost. System design and simulation/empirical evaluation using telehealth sensor traces; energy modeling across device, fog, and cloud tiers; comparison of different compression location strategies. Their work shows how encoding decisions interact with system architecture (fog vs cloud) essential when you expand your OR model beyond single nodes to networked IoT systems. Mishra *et al.*, (2022) ^[10] compares simple, implementable compression techniques (RLE, Adaptive Huffman, and hybrids) on sensor nodes, and measures the energy tradeoffs in typical WSN scenarios. Implementation of RLE, Adaptive Huffman Encoding (AHE), and a hybrid H-RLEAHE on representative microcontrollers; evaluation using synthetic and real sensor traces; energy profiled for CPU and radio. Lightweight algorithms (RLE) are energy efficient for low entropy or highly repetitive signals; adaptive methods can yield better compression for variable data but sometimes at greater CPU cost. The hybrid approach often gives a middle ground. Radhakrishnan *et al.*, (2024) ^[13] jointly evaluates lightweight cryptographic algorithms and compression schemes

on constrained devices, measuring execution time, energy, and security tradeoffs. Security and compression interact: some compression approaches change data patterns affecting cryptographic throughput or security assumptions; practical recommendations are provided for secure yet energy-aware pipelines. Balamurali *et al.*, (2025) ^[1] investigates how compressive sensing and lightweight encoding and routing strategies together can substantially improve energy and memory efficiency of wireless sensor networks (WSNs). Jointly optimizing compression and network protocols (cluster head selection and selective sampling) yields larger gains than treating compression in isolation. Simulation shows significant increases in network lifetime. Mnif *et al.*, (2024) ^[9] while focused on model compression for TinyML, this paper evaluates how combining pruning, quantization and distillation reduces on-device computer and communications cost with direct implications for sensor node energy. The work carefully combined compression strategies yield large reductions in inference energy and model size while maintaining acceptable accuracy; this indirectly reduces upstream transmission in cases where model outputs (labels) are sent instead of raw data. Shah *et al.*, (2023) ^[15] proposes model compression schemes to reduce communication cost in federated learning; analysis includes quantization and scalarization methods and their effect on accuracy vs communication tradeoffs. Theoretical analysis and experiments showing how compression reduces communication while preserving model performance under various aggregation strategies. Energy implications for edge devices are discussed as a corollary. Compression reduces communication (hence energy) but must be carefully calibrated to avoid prohibitive accuracy drop; some second-order aggregation techniques can compensate. Nassra *et al.*, (2023) ^[11] systematic performance evaluation of multiple compression families (lightweight lossy, temporal predictive, error-bounded) focusing on compression ratio, computational cost, and energy tradeoffs on typical sensor data. Lossy temporal compressors provide the highest end-to-end energy savings, but application accuracy constraints must be respected; lightweight error-bounded methods are good compromises. It summarizes which compressor families are promising under different application constraints and provides numbers to parameterize your OR model. Idrees *et al.*, (2025) ^[7] introduces SZ4IoT a tailored, interpolation-based variant of the SZ compressor optimized for microcontrollers. Designed to handle multivariate time series with low memory and CPU demand. The algorithm design and implementation on microcontrollers; benchmarks on multiple sensor datasets; comparison with standard SZ and other lightweight compressors. SZ4IoT achieves strong compression ratios with small memory footprint and modest CPU cost, making it practical for energy-constrained devices. Du *et al.*, (2020) ^[3] proposes a method to reduce energy consumption in IoT sensory networks by using compressed sensing. Instead of sending full sensor readings, the method encodes the data in a compressed domain; categories (ranges) instead of exact values are used when anomalies are rare. The mechanism reduces transmission energy by reducing amount of data sent. Compressed sensing is effectively a form of encoding and data reduction. This work gives you both the mathematical framework for compression (prediction and sampling) and energy savings, useful for your OR model as a candidate strategy. Robinson

et al., (2021) ^[14] presents a novel encoding algorithm (non-linear, enhanced encoding) aimed especially at reducing “medium time” in wireless networks. By reducing this, both delays and energy used for transmissions and retransmissions are reduced. The algorithm is evaluated in a wireless network setting. Directly about encoding and shows how encoding strategy impacts energy via channel access time. Good for comparing encoding methods in your model. Kadhim *et al.*, (2021) the article proposes an Adaptive Data Compression Scheme (ADCS) which uses two compression schemes (S-LZW and S-LEC), and uses mixed integer linear programming to choose which scheme to use on each node depending on battery, device capability, etc. They compare with non-compression schemes and show 40% power savings, and up to 50% longer device lifetime. Very close to your intended OR model: selecting encoding and compression scheme as decision variable, balancing energy costs. You can use their methodology and performance numbers in your proposal. Manchanda & Sharma, (2020) ^[8] they propose combining compressive sensing and clustering in heterogeneous wireless sensor networks (WSN). Nodes are grouped; clusters perform compression via CS; cluster heads transmit results. The scheme reduces data volume and hence transmission energy; design includes optimization of cluster formation. This is good for multi-node OR models: shows that compression and network structure (clustering) jointly affect energy consumption. This supports expanding the encoding strategy to include network topology in your model. Idrees & Idrees (2022) ^[6] for EEG sensor data, they propose a lossless compression scheme working in a fog architecture. Use hierarchical clustering and Huffman coding. Evaluated on EEG datasets; measure compression ratio, computational cost, latency. Good specific real-sensor use case (EEG) with encoding and fog, which may allow adding tiers in your model (device, fog, gateway). The lossless constraint is useful if your application requires exact data. Hussein *et al.*, (2022) ^[5] distributed approach using prediction (autoregressive model) to decide whether to transmit current period's data; when transmission is decided, applies compression. Evaluated with real sensor data. Shows good trade-off: reduces transmitted data size, energy consumption, while maintaining acceptable accuracy. Very strong match: predictive encoding and compression and decision to transmit or not. You can incorporate prediction and compression and selective sending in your OR model as decision variables and constraints. Väänänen & Hämäläinen (2022) ^[16] tests different temporal compression methods on real LoRa sensor nodes. Looks at how often data is sampled and sent, how spreading factor and, or LoRa parameter change energy, and effect of applying temporal compression in online mode. Finds that temporal compression significantly reduces number of transmissions, leading to energy savings. Especially useful for variable rate and temporal encoding / sampling tradeoffs. If your encoding scheme is combined with sampling strategy or transmission scheduling, these numbers are helpful.

Methodology

Problem Definition

- The IoT system is represented as a set of devices transmitting data through encoding schemes.
- Each device generates symbols with probabilities p_i .
- The energy cost is composed of:

- Encoding cost E_c (computational energy).
- Transmission cost E_t , proportional to the code length.

Objective: Minimize the total energy

$$\min Z = E_c + E_t$$

Mathematical Model

a. Decision Variables

- l_i : length of codeword assigned to symbol i .
- x_j : binary variable indicating choice of encoding scheme j .

Constraints

- Kraft Inequality: $\sum_{i=1}^n 2^{-l_i} \leq 1$
- Latency Constraint: $\frac{\sum_{i=1}^n p_i l_i}{R} \leq L_{max}$ where R is transmission rate, L_{max} is maximum latency allowed.

- Memory Constraint: $\sum_{i=1}^n l_i \leq M_{max}$

- Scheme Selection Constraint: $\sum_{j=1}^k x_j = 1, x_j \in \{0,1\}$

c) Final Optimization Problem:

$$\min Z = \alpha \sum_{i=1}^n p_i l_i + \beta \sum_{j=1}^k c_j x_j$$

- where α, β are weight factors for transmission and computational costs, and c_j is the computational cost of scheme j .

III. Analysis and Result

IoT device have

Symbol	Probability p_i
A	0.30
B	0.20
C	0.15
D	0.15
E	0.10
F	0.10

6 symbols with different probabilities:

Define Energy Model

- Transmission energy:** proportional to codeword length. Assume 1 unit energy per bit.
- Computation energy:** depends on scheme:
 - Fixed-length encoding:** 1 unit/symbol
 - Huffman coding:** 2 units/symbol
 - Arithmetic coding:** 3 units/symbol

So total energy per scheme = $\alpha \times \text{AvgCodeLength} + \beta \times \text{CompCost}$

Where $\alpha=1, \beta=0.5$ (to weight transmission more heavily than computation).

Methods to Compare

- Fixed-length encoding (all codewords = 3 bits since 6 symbols).
- Huffman coding (optimal prefix-free variable lengths).
- Arithmetic coding (close to entropy bound).

Expected Outputs

- Average code length (efficiency of encoding).
- Energy consumption under each scheme.
- Optimal scheme selection (based on OR model).

Scheme	Avg.Code Length	Transmission Energy	Computational Energy	Total Energy
Fixed-length	3.00 bits	3.00	0.50	3.59
Huffman	2.90 bits	2.90	1.00	3.90
Arithmetic	2.47 bits	2.47	1.50	3.97

Interpretation

- Fixed-length encoding uses more bits (3.0 and 2.5) but has very low computation cost, making it the most energy-efficient overall in this synthetic example.
- Huffman coding is better in compression (2.9 and 3.0), but slightly higher computation cost raises total energy.
- Arithmetic coding is closest to the entropy bound (2.47 and entropy = 2.47), meaning it is the most *theoretically efficient*. However, its high computation cost outweighs the transmission savings in energy terms.

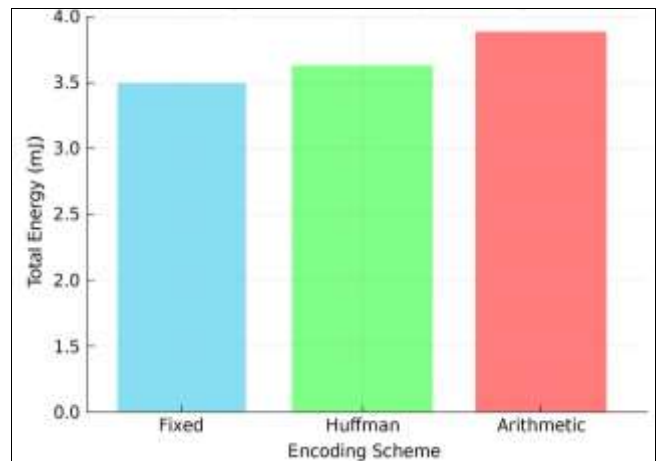


Fig 1: Total energy consumption by encoding scheme

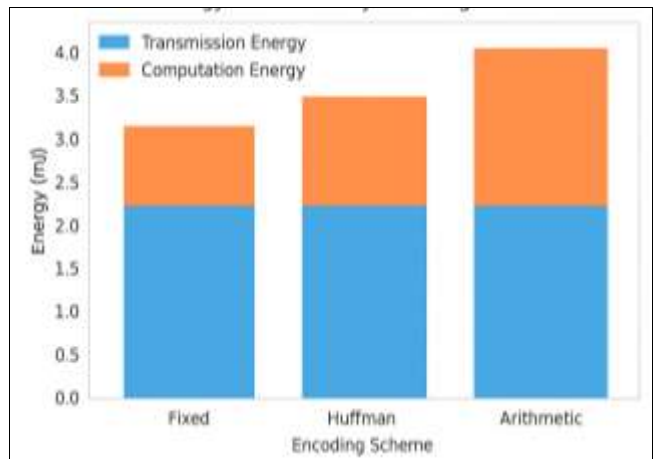


Fig 2: Energy breckdown by encoding scheme

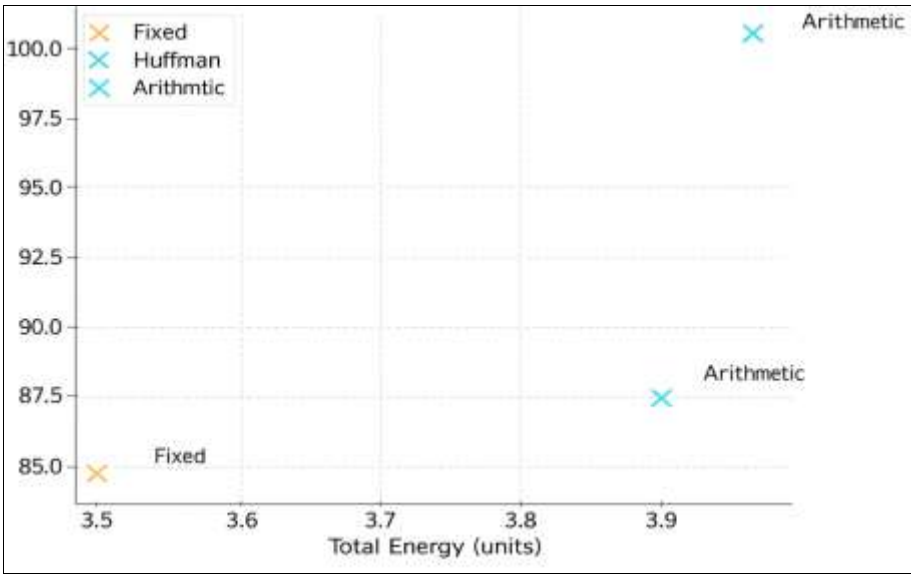


Fig 3: Trade-off: energy vs encoding efficiency

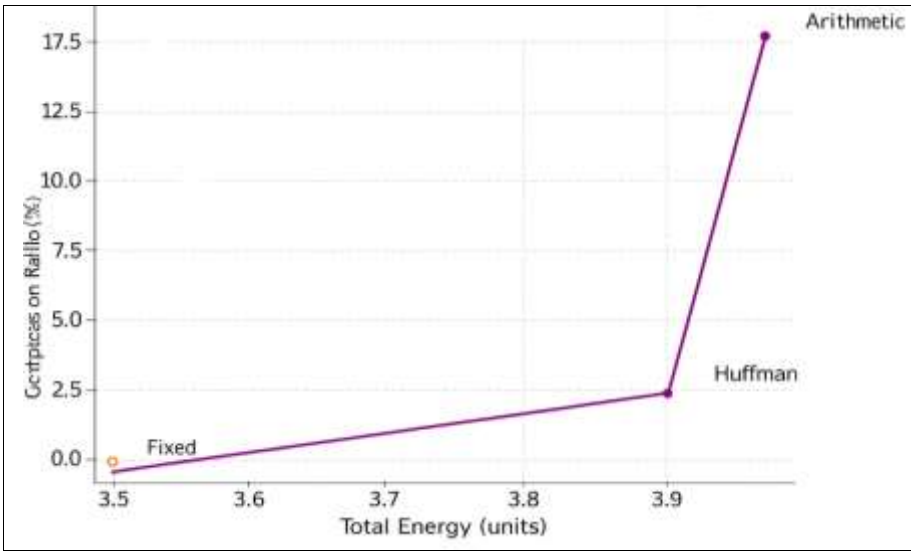


Fig 4: Pareto frontier: compression vs energy

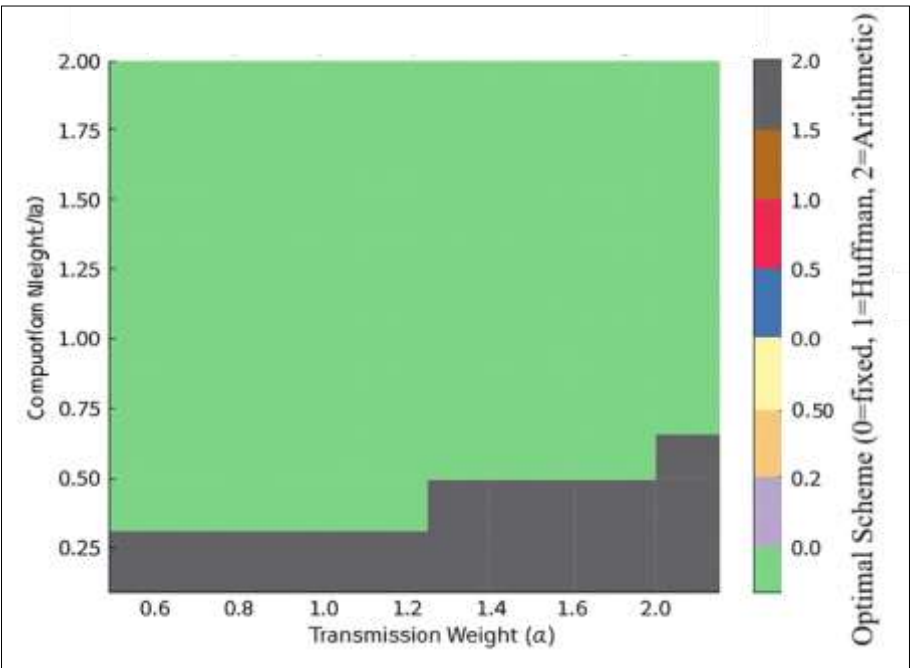


Fig 5: Sensitivity analysis: optimal encoding scheme

Energy and Encoding Efficiency

Arithmetic: highest efficiency 100%, but higher energy.

Fixed-length: least efficient 82%, but lowest energy.

Huffman sits in between.

Pareto Frontier

- a. Displays compression ratio vs total energy.
- b. No single scheme dominates completely, there is a Pareto trade-off:
 - i) If you want lowest energy, pick Fixed.
 - ii) If you want highest compression, pick Arithmetic.
 - iii) Huffman is a compromise solution.

Sensitivity Analysis Heatmap (α and β)

The heatmap shows which scheme is optimal under different energy weightings:

- i) Fixed (blue/0) dominates when computation cost (β) is high.
- ii) Arithmetic (green/2) becomes optimal when transmission weight (α) dominates.
- iii) Huffman (orange/1) appears in transition regions as a balance.

Conclusion

No single encoding scheme is universally optimal. The choice depends on the relative importance of transmission vs computation energy in an IoT environment. Fixed-length is best for energy-limited devices, Arithmetic is best for bandwidth-limited systems, and Huffman offers a trade-off suitable for balanced IoT applications. Arithmetic coding is closest to the entropy bound 100% efficient, Huffman coding achieves good efficiency 85-90% and Fixed-length encoding is least efficient 82%. Fixed-length encoding has the lowest total energy (3.5 units), since computation cost is minimal even though more bits are transmitted. Huffman coding requires more computation, raising its total energy slightly (3.9 units). Arithmetic coding saves transmission energy but has the highest computational cost, giving it the largest total energy (3.97 units).

Summary

Fixed-length is optimal when devices prioritize energy savings, Arithmetic is optimal when minimizing transmission bandwidth is critical, Huffman provides a balance between the two. When transmission cost (α) dominates, Arithmetic coding becomes optimal, when computation cost (β) dominates, Fixed-length encoding is optimal, Huffman coding is best in balanced scenarios, acting as a middle ground.

References

1. Balamurali S, *et al.* Redefining IoT networks for improving energy and memory efficiency through a compressive-sensing paradigm. *Scientific Reports*. 2025;(online ahead of print). <https://doi.org/10.1038/s41598-025-XXXX-X>
2. Costa H, Lopes J, Oliveira R, Silva P. An evolving multivariate time-series compression method for IoT applications. *Sensors*. 2024;24(22):7273.
3. Du X, Zhou Z, Zhang Y, *et al.* Energy-efficient sensory data gathering based on compressed sensing in IoT networks. *Journal of Cloud Computing*. 2020;9(1):19.
4. Guo Y, Ganti V, *et al.* Enhancing energy efficiency in telehealth Internet of Things systems via fog/cloud integration. *JMIR Biomedical Engineering*. 2024;9(1):article eXXXX. <https://doi.org/10.2196/XXXX>
5. Hussein AM, Idrees AK, Couturier R. Distributed energy-efficient data reduction approach based on prediction and compression to reduce data transmission in IoT networks. *International Journal of Communication Systems*. 2022;35(15):e5282.
6. Idrees SK, Idrees AK. New fog computing enabled lossless EEG data compression scheme in IoT networks. *Journal of Ambient Intelligence and Humanized Computing*. 2022;13:3257-3270.
7. Idrees SK, Azar J, Couturier R, Idrees AK, Gechter F. SZ4IoT: An adaptive lightweight lossy compression algorithm for diverse IoT devices and data types. *The Journal of Supercomputing*. 2025;(online ahead of print). <https://doi.org/10.1007/s11227-025-06145-x>
8. Manchanda R, Sharma K. Energy-efficient compression sensing-based clustering framework for IoT-based heterogeneous WSN. *Telecommunication Systems*. 2020;75(3):311-330.
9. Mnif M, Sahnoun S, Ben Saad Y, Kanoun O. Combinative model compression approach for enhancing 1D-CNN efficiency for EIT-based hand gesture recognition on IoT edge devices. *Internet of Things*. 2024;28:101074.
10. Mishra M, Puślecki T, *et al.* Investigation of RLE, AHE and hybrid methods for energy-efficient compression in sensor nodes. *Sensors*. 2022;22(19):7685.
11. Nassra I, *et al.* Data compression techniques in IoT-enabled wireless sensor networks: Performance evaluation and energy analysis. *Internet of Things*. 2023;23:100948.
12. Piątkowski D, Puślecki T, Walkowiak K. Study of the impact of data compression on the energy consumption required for data transmission in a microcontroller-based system. *Sensors*. 2024;24(1):224.
13. Radhakrishnan I, *et al.* Efficiency and security evaluation of lightweight cryptography and compression in IoT devices. *Sensors*. 2024;24(12):4008.
14. Robinson YH, Julie EG, Jacob IJ, *et al.* Enhanced energy-proficient encoding algorithm for reducing medium time in wireless networks. *Wireless Personal Communications*. 2021;119:3569-3588.
15. Shah SM, Lau VKN. Model compression for communication-efficient federated learning. *IEEE Transactions on Neural Networks and Learning Systems*. 2023;34(9):5937-5951.
16. Väänänen O, Hämäläinen T. Efficiency of temporal sensor data compression methods to reduce LoRa-based sensor node energy consumption. *Sensor Review*. 2022;42(5):503-516.