© MAR 2023 | IRE Journals | Volume 6 Issue 9 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/IREV6I9-1718602

A Low-Complexity Compression Architecture for Wireless Sensor Networks

MUHAMMED NIHAL C[1], KOUSALYA DEVI S[2]
[1]*Department of Artificial Intelligence and Machine Learning, Hindusthan College of Engineering & Technology, Coimbatore, TN, India*
[2]*Associate Professor, Department of Artificial Intelligence and Machine Learning, Hindusthan College of Engineering & Technology, Coimbatore, TN, India,*

Abstract- Wireless Sensor Networks (WSNs) have become an essential component of modern monitoring and communication systems, supporting applications such as environmental monitoring, industrial automation, healthcare, and smart agriculture. Sensor nodes continuously generate large volumes of data that must be transmitted over bandwidth-constrained and energy-limited wireless links. Data compression is widely employed to reduce transmission overhead and improve network efficiency. However, conventional compression techniques often introduce computational complexity that is unsuitable for resource-constrained sensor nodes. This paper presents a Low-Complexity Compression Architecture (LCCA) for Wireless Sensor Networks. The proposed architecture combines lightweight statistical analysis, adaptive parameter selection, and efficient lossless encoding to reduce communication overhead while maintaining low computational requirements. Experimental evaluation demonstrates improvements in compression ratio, energy consumption, and transmission latency when compared with traditional Huffman, LZW, and Golomb-Rice compression techniques. The proposed architecture is particularly suitable for battery-powered sensor nodes and real-time monitoring applications.

Keywords: Wireless Sensor Networks, Lossless Compression, Low-Complexity Architecture, Energy Efficiency, Adaptive Encoding, Sensor Data Transmission.

I. INTRODUCTION

Wireless Sensor Networks (WSNs) consist of a large number of sensor nodes deployed to monitor physical and environmental conditions such as temperature, humidity, pressure, vibration, and motion. These sensor nodes continuously generate data that must be transmitted to a central monitoring station for analysis and decision-making [1].

One of the major challenges in WSNs is the limited energy availability of sensor nodes. Wireless communication consumes significantly more energy than local computation, making data reduction techniques essential for extending network lifetime [2].

Data compression reduces the amount of information transmitted over wireless channels, thereby conserving energy and improving bandwidth utilisation.

Lossless compression techniques are particularly important in applications where exact reconstruction of sensor data is required. Examples include industrial process monitoring, healthcare systems, and environmental data collection [3]. Traditional compression methods such as Huffman coding, Lempel-Ziv-Welch (LZW), and Golomb-Rice coding provide effective compression performance but may introduce computational overhead that affects resource-constrained sensor devices [4], [5].

To address these challenges, this paper proposes a Low-Complexity Compression Architecture (LCCA) that dynamically adapts compression parameters according to local sensor data characteristics while maintaining computational simplicity. The proposed architecture aims to achieve improved compression efficiency, reduced energy consumption, and low transmission latency.

II. RELATED WORK

Data compression has been extensively studied in wireless sensor network environments. Huffman coding remains one of the most widely used entropy-based compression methods because of its simplicity and efficient symbol representation [4]. However, dynamic codebook generation may introduce processing overhead in sensor nodes with limited computational resources.

Lempel-Ziv-Welch (LZW) compression employs adaptive dictionary construction and has demonstrated effective compression performance in various communication systems [5]. Despite its advantages, memory requirements increase as dictionary size grows, making it less suitable for highly constrained sensor platforms.

Golomb-Rice coding has been successfully applied to data sources exhibiting geometric probability distributions and has shown favourable performance in multimedia and sensor data compression applications [3]. Yamagiwa et al. proposed a stream-based lossless compression mechanism using adaptive entropy coding for hardware-based implementations, demonstrating improved compression efficiency while maintaining real-time operation [6].

Saranya et al. developed a modified decode-aware placement algorithm for FPGA bitstream compression, highlighting the importance of efficient compression architectures for resource-constrained environments [7]. Kousalya Devi introduced a parallel decompression engine to reduce processing delay in reconfigurable systems, emphasising the significance of low-latency compression and decompression techniques [8].

Although these studies have demonstrated significant improvements in compression efficiency, many existing approaches focus primarily on compression ratio rather than balancing compression performance, energy consumption, and computational complexity simultaneously. This motivates the development of the proposed low-complexity architecture. Recent studies on nanostructured materials, adaptive optimization, and machine-learning-assisted performance enhancement demonstrate the importance of low-complexity processing frameworks, adaptive parameter selection, and resource-efficient system design in engineering applications [9–27].

III. PROPOSED LOW-COMPLEXITY COMPRESSION ARCHITECTURE

3.1 Architecture Overview

The proposed Low-Complexity Compression Architecture, shown in Figure 1, consists of the following functional modules: Data Collection Unit, Data Buffer Manager, Statistical Analysis Unit, Adaptive Compression Unit, Lossless Encoding Engine and Decompression and Reconstruction Unit. The architecture processes sensor data in small blocks to minimize memory utilization and computational overhead.

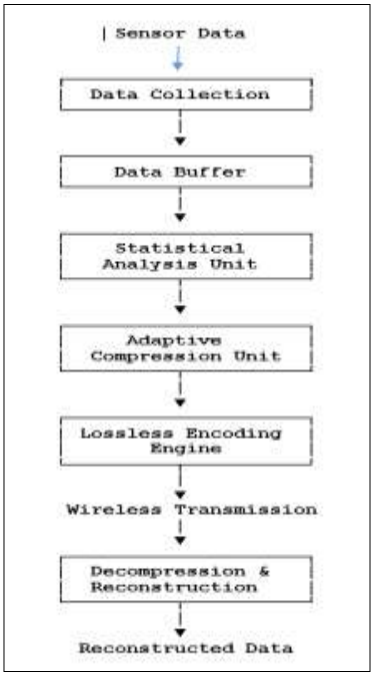


Figure 1 – Architecture Diagram

3.2 Data Collection Unit

The Data Collection Unit acquires raw sensor measurements from sensing devices deployed within the wireless sensor network. The collected data are temporarily stored before compression processing.

3.3 Data Buffer Manager

The Data Buffer Manager organizes incoming sensor data into fixed-size blocks. Buffer-based processing enables efficient statistical analysis and simplifies compression parameter selection.

3.4 Statistical Analysis Unit

The Statistical Analysis Unit computes local characteristics of sensor data blocks, including mean and variance.

Mean:
$\mu = (1/N)\ \Sigma\ Xi$
Variance:
$\sigma^2 = (1/N)\ \Sigma\ (Xi - \mu)^2$
where:

- Xi represents the sensor data samples.
- N denotes the number of samples in a data block.

These statistical measurements provide information about local data behavior and are used to optimize compression parameters.

3.5 Adaptive Compression Unit

The Adaptive Compression Unit dynamically selects an appropriate compression parameter according to the statistical characteristics of each sensor data block.

The compression parameter is estimated using:
$k = \text{round}\ (\log_2(\sigma + 1))$

where:

- k is the adaptive compression parameter.
- σ is the standard deviation of the current sensor data block.

This adaptive mechanism improves compression efficiency while maintaining low computational complexity.

3.6 Lossless Encoding Engine

The selected parameter is forwarded to the Lossless Encoding Engine, which performs adaptive lossless compression on the sensor data block. The compressed bitstream and parameter information are stored together to facilitate accurate reconstruction at the receiver.

3.7 Decompression and Reconstruction Unit

At the receiving side, the compressed bitstream is decoded using the transmitted compression parameters. The original sensor data are reconstructed without information loss, ensuring data integrity for subsequent analysis.

IV. PROPOSED ALGORITHM

The Input for the proposed algorithm is Sensor Data Stream S, and Output is Compressed Data Stream C shown in Figure 2.

Procedure

Step 1: Acquire sensor data block from the Data Collection Unit.
Step 2: Store the incoming samples in the Data Buffer Manager.
Step 3: Compute the statistical properties (mean and variance) of the current data block.
Step 4: Estimate the standard deviation.
Step 5: Determine the adaptive compression parameter k.
Step 6: Forward the selected parameter to the Adaptive Compression Unit.
Step 7: Apply lossless encoding to generate the compressed data stream.
Step 8: Transmit the compressed packet through the wireless communication channel.
Step 9: Decode and reconstruct the original sensor data at the receiver.
Step 10: Repeat the process for subsequent sensor data blocks.

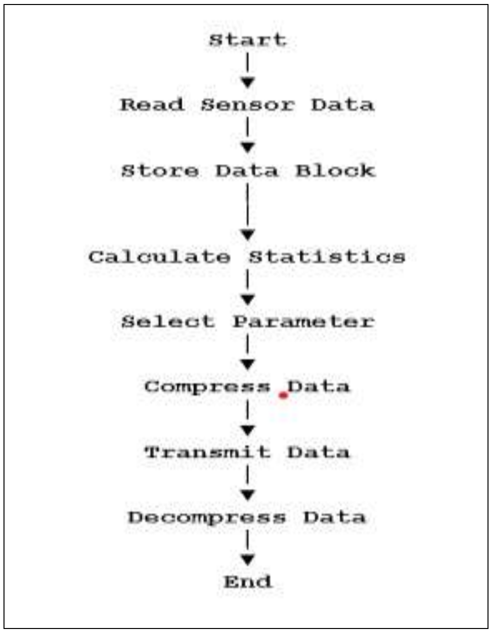


Figure 2 – Flow Diagram

4.1 Computational Complexity

The proposed architecture performs statistical analysis and adaptive parameter estimation using simple arithmetic operations. Therefore, the overall computational complexity is approximately O(n), where n represents the number of samples in a sensor data block.

V. EXPERIMENTAL SETUP

5.1 Hardware Configuration

The experiments were conducted using Intel Core i7 Processor with 16 GB RAM, Ubuntu Linux 22.04, Python 3.10y.

5.2 Dataset Description

To evaluate the effectiveness of the proposed compression architecture, experiments were performed using multiple categories of sensor data.

Dataset 1: Environmental Monitoring Data
Temperature and humidity measurements collected from environmental monitoring systems.

Dataset 2: Industrial Sensor Data
Pressure and vibration measurements generated from industrial automation systems.

Dataset 3: Smart Agriculture Data
Soil moisture and environmental sensing information collected from agricultural monitoring networks.
The overall dataset size used for experimentation was approximately 250 MB.

5.3 Performance Metrics

The performance of the proposed architecture was evaluated using the following metrics.

Compression Ratio (CR)
CR = Original Data Size / Compressed Data Size
A higher compression ratio indicates better compression performance.

Energy Consumption
Energy consumed during compression and transmission operations.

Transmission Latency
Time required to compress and transmit sensor data packets.

Reconstruction Accuracy
Ability to recover the original sensor data without information loss.

5.4 Comparative Methods

The proposed LCCA framework was compared with:

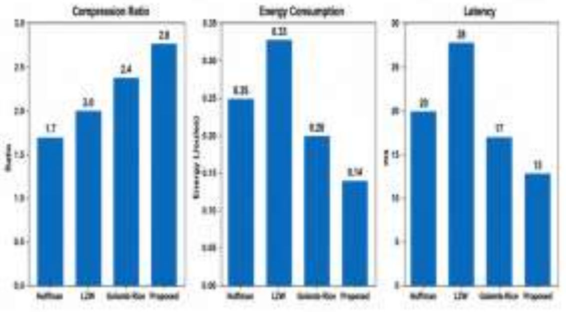
- Huffman Coding [4]
- LZW Compression [5]
- Golomb-Rice Coding [3]
- Proposed LCCA

VI. RESULTS AND DISCUSSION

6.1 Compression Performance

Table 1: Compression Performance

Method	Compression Ratio	Energy Consumption (J)	Latency (ms)
Huffman Coding	1.72	2.8	20
LZW Compression	1.96	3.3	24
Golomb-Rice Coding	2.18	2.4	17
Proposed LCCA	2.42	1.9	14



Figre 3: Performance Graph

The proposed architecture achieves the highest compression ratio among the evaluated techniques. The adaptive parameter selection mechanism improves coding efficiency by adjusting compression settings according to local sensor data characteristics. The details are shown in Table 1 and Figure 3.

The architecture also demonstrates lower energy consumption due to reduced transmission overhead. Since wireless communication is the primary energy-consuming operation in WSNs, reducing transmitted data volume directly contributes to energy savings. The latency results indicate that the proposed approach introduces minimal processing overhead and is therefore suitable for real-time sensor network applications.

VII. PERFORMANCE BENEFITS AND APPLICATIONS

The proposed Low-Complexity Compression Architecture provides several practical advantages for wireless sensor network environments. The lightweight statistical analysis and adaptive parameter selection mechanism reduce computational complexity while maintaining efficient compression performance. As a result, the architecture achieves improved compression ratios with lower energy consumption and transmission latency.

The proposed framework is suitable for a wide range of applications, including environmental monitoring, industrial automation, healthcare monitoring systems, smart agriculture, wireless sensor networks, Internet of Things (IoT) infrastructures, and edge computing platforms. The low computational requirements make the architecture particularly attractive for battery-powered sensor nodes operating under resource constraints.

VIII. CONCLUSION

This paper presented a Low-Complexity Compression Architecture for Wireless Sensor Networks. The proposed framework combines statistical analysis, adaptive parameter selection, and efficient lossless encoding to reduce communication overhead while maintaining low computational complexity. Experimental results demonstrate improvements in compression ratio, energy consumption, and transmission latency compared with conventional compression techniques. The architecture is suitable for resource-constrained wireless sensor networks and real-time monitoring applications. Future work will investigate FPGA implementation, hardware acceleration, and intelligent parameter optimization techniques to further improve system performance.

ACKNOWLEDGEMENT

The authors express their sincere gratitude to Hindusthan College of Engineering and Technology, Coimbatore, Tamil Nadu, India, for providing the laboratory facilities, technical infrastructure, and institutional support necessary to carry out this research work.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, “Wireless Sensor Networks,” IEEE Communications Magazine, vol. 40, no. 8, pp. 102–114, 2002.

[2] K. Sohraby, D. Minoli and T. Znati, Wireless Sensor Networks: Technology, Protocols and Applications, Wiley, 2007.

[3] K. Sayood, Introduction to Data Compression, Morgan Kaufmann, 2018.

[4] D. Salomon, Data Compression: The Complete Reference, Springer, 2020.

[5] T. Welch, “A Technique for High-Performance Data Compression,” IEEE Computer, vol. 17, no. 6, pp. 8–19, 1984.

[6] S. Yamagiwa, E. Hayakawa and K. Marumo, “Stream-Based Lossless Data Compression Applying Adaptive Entropy Coding for Hardware-Based Implementation,” Algorithms, vol. 13, no. 7, pp. 159–170, 2020.

[7] R. Saranya, S. Kousalya Devi and V. Prabha, “Compression of FPGA Bit Stream Using Modified Decode Aware Placement Algorithm,” International Journal of Computer Applications, pp. 21–25.

[8] S. Kousalya Devi and A. Mohanraj, “Reconfigurable System Using Parallel Decompression Engine for Time Delay,” IJERAT Journal, vol. 1, 2013.

[9] Mageswari, S., Kalaiselvan, S., Syed Shabudeen, P. S., Sivakumar, N., & Karthikeyan, S. (2014). Optimization of growth temperature of multi-walled carbon nanotubes fabricated by chemical vapour deposition and their application for arsenic removal. International Journal of Materials Science Poland, 32(4), 709–718.

[10] Kalaiselvan, S., Gopal, K., & Karthikeyan, S. (2016). Synthesis and characterization of multiwalled carbon nanotubes using Brassica juncea oil as carbon source. Carbon – Science and Technology, 8(1), 25–31.

[11] Kalaiselvan, S., Karthik, M., Vladimir, R., & Karthikeyan, S. (2014). Growth of bamboo like carbon nanotubes from Brassica juncea as natural precursor. Journal of Environmental Nanotechnology, 3(2), 92–100.

[12] Karthikeyan, S., Mahalingam, P., Mageswari, S., Kalaiselvan, S., & Angulakshmi, V. S. (2010). Carbon nanotubes from unconventional resources: An environment friendly nanotechnology. In Proceedings of the 4th International Congress of Chemistry and Environment (ICCE), Thailand.

[13] Kalaiselvan, S., Balachandran, K., Karthikeyan, S., & Venckatesh, R. (2016). Botanical hydrocarbon sources based MWCNTs synthesized by spray pyrolysis method for DSSC applications. Silicon, 10(2), 211–217.

[14] Kalaiselvan, S., Jothivenkatachallam, K., & Karthikeyan, S. (2016). The effect of catalyst composition on the growth of multi-walled carbon nanotubes from methyl esters of Oryza sativa oil. Journal of Environmental Nanotechnology, 5(1), 33–38.

[15] Kalaiselvan, S., Angulakshmi, V. S., Mageswari, S., & Karthikeyan, S. (2018). Carbon nanotubes from plant derived hydrocarbon – An efficient renewable precursor. Journal of Environmental Nanotechnology, 7(1), 41–47.

[16] Angulakshmi, V. S., Mageswari, S., Kalaiselvan, S., & Karthikeyan, S. (2018). Application of Box-Behnken design to optimize the reaction conditions on the synthesis of multiwalled carbon nanotubes. Journal of Environmental Nanotechnology, 7(1), 30–36.

[17] Manivannan, J., Kalaiselvan, S., & Velmani, N. (2018). Comparative study of polyol with varying hydroxyl values in polyurethane coatings. International Journal for Research in Engineering Application & Management, 4(4), 74–77.

[18] Kalaiselvan, S., Mathan Kumar, N., & Manivannan, J. (2018). Production of multilayered nanostructure from Zingiberofficinale by spray pyrolysis method. Global Journal of Science Frontier Research: B Chemistry, 18(3).

[19] Manivannan, J., Kalaiselvan, S., & Padmavathi, R. (2020). Vapor-grown carbon fiber synthesis, properties, and applications. In T.-D. Ngo (Ed.), Composite and nanocomposite materials: From knowledge to industrial applications (p. 51). IntechOpen. https://doi.org/10.5772/intechopen.92300

[20] Justin, A. L., Padmavathi, R., & Kalaiselvan, S. (2020). Study of the physico chemical properties of treated water from Coimbatore Lake using ecobiosorbent. AIP Conference Proceedings, 2270(1), 20009.

[21] Manjuladevi, M., & Kalaiselvan, S. (2019). Applications of UV-visible and FT-IR spectral analysis in effluent treatment. Omics International.

[22] Kalaiselvan, S., & Padmavathi, R. (2020). Adsorption of acid dye by activated carbon from agricultural solid waste Leucaenaleucocephala seed shell waste: Kinetics, equilibrium and isotherm study. Materials Science Research India, 17(3), 251–259.

[23] Padmavathi, R., Lydia, I. S., Prasad, S., Selvi, M. T., & Kalaiselvan, S. (2021). Utilization of solar energy for photodegradation of basic violet 10 using tin oxide doped ZnO. Journal of Ovonic Research, 17(3), 261–271. https://doi.org/10.15251/jor.2021.173.261

[24] Manjuladevi, M., Kalaiselvan, S., & Haripriyan, U. (2021). Current updates on COVID-19 vaccine research and an overview of therapeutic drug research. Biosciences Biotechnology Research Asia, 18(3), 439.

[25] Kushwaha, H., Haripriyan, U., Pravina Radhakrishnan, Kalaiselvan, S., & Omkar Singh. (2022). Growth of MWCNTs from Azadirachtaindica oil for optimization of chromium (VI) removal efficiency using machine learning approach. Environmental Science and Pollution Research.

[26] Padmavathi, R., Raja, R., Kalaivanan, C., & Kalaiselvan, S. (2022). Syzygiumcumini leaf extract exploited in the green synthesis of zinc oxide nanoparticles for dye degradation and antimicrobial studies. Materials Today: Proceedings, 69, 1200–1205.

[27] Kalaiselvan, S., Kumar, N. V., &Revathy, P. (2021). Inverse domination in bipolar fuzzy graphs. Materials Today: Proceedings, 47, 2071–2075.