Efficient and Error Free Missile Telemetric Data Compression Using LZMA2

Sudhakara Reddy Byreddy
Assistant Professor,Department of CSE
Vignan's Lara Institute of Technology & science
Andhra Pradesh, India
byreddysudh@gmail.com

Raja Rajeswari Nuthi
Department of ACSE
Vignan's Lara Institute of Technology & science
Andhra Pradesh, India
rajeswarinuthi209@gmail.com

Nitesh Bolleddu
Department of ACSE
Vignan's Lara Institute of Technology & science
Andhra Pradesh, India
bolleddunitesh77@gmail.com

Lohith Venkata Sai Charan Seeda
Department of ACSE
Vignan's Lara Institute of Technology & science
Andhra Pradesh, India
slvsaicharan@gmail.com

P. Lohith Reddy
Department of ACSE
Vignan's Lara Institute of Technology & science
Andhra Pradesh, India
p.lohithreddy2810@gmail.com

***Abstract*— Missile plays key role in present battle field to carry accurate attacks over different targets from aerial platforms such as airplanes, submarines, ground based launchers and vessels. Meanwhile on the same period, huge communication challenges exist for missiles. Vast communication difficulties exist for missiles that includes the plasma sheath and associated blackout regions. Moreover, the associated plasma, created the hot exhaust gases reacting to the atmospheric gases reacting to the atmospheric gases, that generates Electromagnetic Interferes(EMI). Furthermore, EMI interferes with the guidance and telemetry communication channels in the missiles. The blackout also makes it difficult for the enemy's radars to identify targets with the help of wave scattering to track the targets. Concerning the overcoming challenges that associated with data transmission in the time of blackout intervals. In this research the lossless compresses communication for compresses telemetry data. Exploiting the previous research on Lempel-Ziv-Markov Algorithm(LZMA), along with GZIP method, which creates high effectiveness on the compression or artificially produced sensor telemetry data. Moreover, this research Proposes the new approach of advanced LZMA. The generated telemetry data and new approach of LZMA2 outperformed previous approaches for high accuracy (100 %) and high compression ratio(5.85), outperforming previous method of LZMA(95.5 %) and 5.58 compression ratio, followed by GZIP approach(ratio 1.41).**

Keywords*— *Aerospace data transmission , EMI-resilient communication, Lossless compression, LZMA2 algorithm, Missile telemetry, Plasma-induced blackout,.

I. INTRODUCTION

Missile telemetry data captures a steady stream of details about how a missile performs and behaves as it flies along its path. This information is crucial for developing, testing, and analyzing missiles in action. Onboard sensors pick up key measurements like acceleration, temperature, pressure, voltage, speed direction, and control inputs all during the flight. Sensor outputs start as electrical signals either analog or digital, based on the setup. To make them reliable for sending and handling, engineers digitize them through sampling and analog to digital conversion. Then, they encode these digital signals with pulse code modulation to faithfully capture the measurements .[3].

These telemetry packets travel through radio frequency to a satellite or ground receiver. But in high-speed flight through the atmosphere, a plasma sheath forms around the missile, creating electromagnetic noise and weakening signals. This leads to blackout periods that disrupt reliable data transmission .

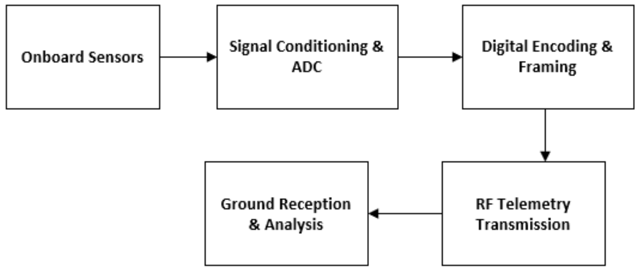

Fig 1: Raw Signal Compilation Engine

Performance Signal Readings involves the continuous evaluation of multiple parameters, resulting in the generation of massive data. The transmission of such extensive data becomes challenging and inefficient. Therefore, data compression techniques are essential to reduce data size, ensuring efficient transmission and storage while preserving the original data content.

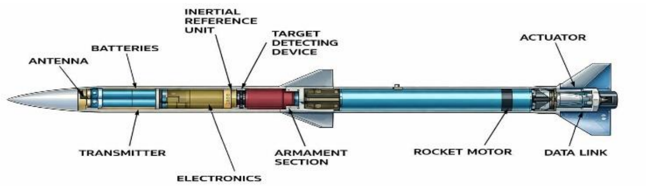

Fig. 2. Internal representation of a projectile highlighting electronic components.[2].

II. LITERATURE SURVEY

Information compaction lowers bits needed to represent digital signals for storage or transmission. In communication systems, employing data compression accelerates transfer rates effectively, reduces bandwidth requirements, and minimizes storage space. In telemetry based systems where large volumes of data are generated continuously compression becomes essential to ensure reliable transmission under limited communication resources.

The main job of data compression is dropping unnecessary repeats to save all vital content. By compressing telemetry data prior to transmission, more information can be sent within a limited communication window, which is particularly important in missile telemetry systems affected by communication blackout conditions. Data compression techniques are broadly classified into lossless and lossy compression methods.

Lossless compression techniques minimize data volume without any information loss, enabling complete restoration of the original data following decompression. These methods are required for telemetry use cases, where precise sensor measurements prove vital for mission evaluation and performance verification. Typical lossless approaches encompass Run Length Encoding, Huffman coding, Lempel–Ziv–Welch, GZIP, and Lempel–Ziv–Markov Algorithm [4,5].

Lossy data compression, by contrast, shrinks data volume by permanently removing less critical details. Although this approach finds common use in multimedia like images, audio, and video, it proves unsuitable for missile telemetry systems, where even slight data loss could lead to misinterpretation of sensor measurements and system performance.

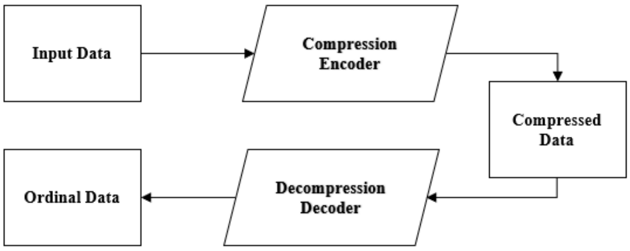

Fig. 3. Fundamental block layout for the compression procedure.

Telemetry essentially take live readings from sensors far away and sends them wirelessly to the control team. In missiles, these sensors keep records on key details like speed and acceleration, heat and pressure, power supply, the structure's condition, engine performance, and even the environment around it. Ground operators use this constant data feed to watch things unfold in real time, predict the missile's path, spot issues early, and review what happened afterward. But with data pouring in non-stop and in huge amounts, clever compression techniques are crucial to deliver it reliably and without delay.

When evaluating telemetry data compression methods, engineers weigh key factors like compression ratio, transmission range, data throughput, error rates, and overall efficiency. The best techniques deliver strong compression decreasing data size dramatically without sacrificing accuracy or reliability. This balance proves vital in critical scenarios, such as missile flights during signal blackouts, where every transmission window counts and even brief disruptions can compromise mission success .

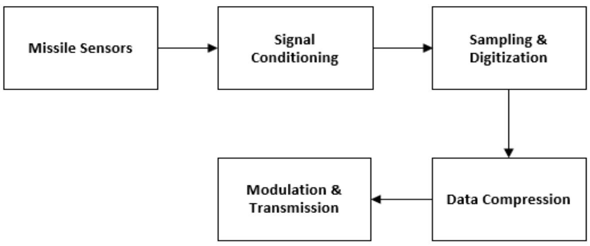

Fig. 4. Data Gathering and Uplink System

Fig. 4 describes The Data Gathering and Uplink system begins with onboard missile sensors that measure various flight and system parameters. The sensor outputs are conditioned to improve signal quality and then sampled and digitized using sampling conversion techniques. The digitized telemetry data is compressed to reduce data size before being modulated onto a carrier signal and transmitted through the missile antenna to the ground station during limited communication windows [6].

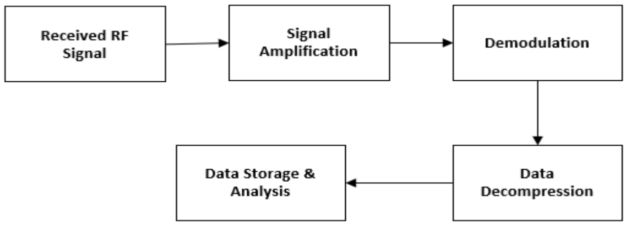

Fig. 5. Data Reception and Processing System

The telemetry receiving system is responsible for recovering and processing telemetry data transmitted from the missile. The received radio frequency signal is first amplified to improve signal strength and then demodulated to extract the compressed telemetry data. The compressed data is decompressed using the corresponding decompression algorithm to recover the original telemetry information. The recovered data is stored and analyzed for real-time monitoring and post-flight evaluation. Accurate decompression ensures that telemetry data integrity is preserved throughout the communication process [8].

Traditional reversible compression techniques such as Huffman Coding, Run Length Encoding, Lempel–Ziv–Welch (LZW), GZIP (DEFLATE), and also Lempel–Ziv–Markov Algorithm (LZMA) have been widely used in missile telemetry systems to reduce transmission bandwidth while ensuring accurate data recovery.

A. Run Length Encoding (RLE)

Run-Length Encoding (RLE) is a straightforward lossless technique that reduces data size by replacing sequences of identical values with a single instance and a count of its repetitions. In missile telemetry, it is effective for low-change data like status flags, on-off control signals, or startup routines where readings stay steady for brief intervals. Its simple design makes RLE ideal for real-time processing within missiles, since it uses very little computing power or memory. That said, missile telemetry typically involves fast-shifting sensor data like acceleration, pressure, and orientation which limits RLE's real-world punch during flights .[11,12].

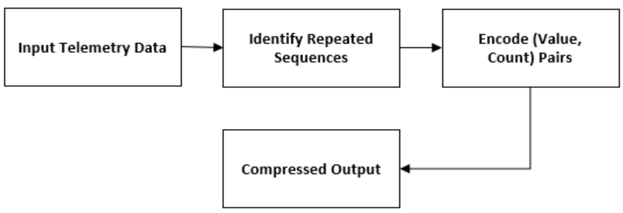

Fig. 6. RLE algorithm execution sequence.

Fig. 6 Presents the workflow for RLE shown below

- Data Gathering: Sensors grab real-time data from the fake dataset stream, feeding it straight into compression like outputs.
- Repetition Detection: RLE quickly scans for repeating values in stable parts, spotting redundancy without heavy computing.

- Encoding: It groups repeats as simple value-count pairs, minimizing data but keeping it fully accurate and lossless.
- Compressed Output: Pairs go out for transmission, saving bandwidth with fast, simple processing for live links.

B. Huffman Coding

Huffman coding reduce data without losing any details by giving short binary tags to common symbols and longer ones to rare ones, based on how often they appear. In missile telemetry, it works great on steady sensor readings or status updates from onboard systems, which follow reliable patterns.

Everyday symbols get tiny codes, so less data travels to the ground station saving precious bandwidth. This method achieves efficient, no-overlap coding. But it struggles if flight stages shift quickly, disrupting those patterns [6].

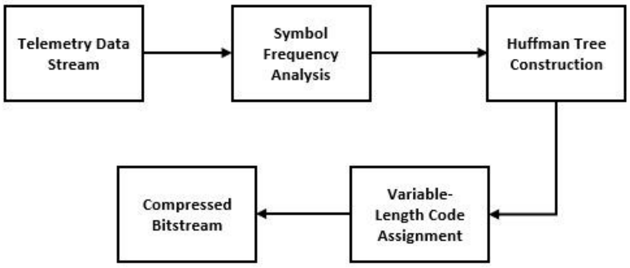

Fig. 7. Huffman Coding operational steps.

Fig. 7 Presents the workflow for Huffman coding shown below

- Input Data Stream: Uncompressed data streams from the sensors are consistently routed into our compression system
- Frequency Analysis of Symbols: We tally how often each data value shows up, marking the common ones for quick handling.
- Huffman Tree Construction: Using those counts, we grow a tree where busy symbols sit near the top, earning short paths.
- Variable-Length Code Assignment: Everyday values get brief binary labels; unusual values take longer ones, straight from the tree spots.
- Compressed Bitstream: Swap originals for these tags to spit out a tight bitstream less bandwidth, full unpack later.

C. Lempel–Ziv–Welch (LZW)

Lempel–Ziv–Welch (LZW) compression creates a phone book of repeating patterns in facts, swapping them for short numbers to save space without losing information. For missile telemetry, it shines on organized sensor packets and steady readings during calm flight stretches. It smartly grows this book from patterns it has seen, so no need to send the book along. But when flights get wild like sharp turns or re-entry the book's rigid growth slows down the space savings[4].

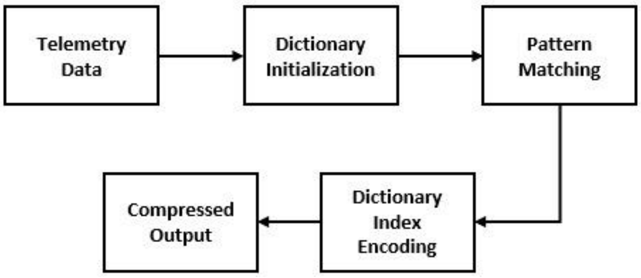

Fig. 8. LZW algorithm execution sequence

Fig. 8 Presents the workflow for LZW shown below

- Input Data: Extract the uncompressed flow from dataset sensors, full of potential repeats..
- Initialize Dictionary: Initiate with a basic list of single values, each with its own short code number.
- Pattern Matching: Slide through the stream, grabbing the longest chunk already in the book.
- Dictionary Index Encoding: Output the code number for matches; add fresh chunks to the book with new numbers.
- Compressed Output: End up with a short list of codes way less data to send, though wild data enlarges the book.

D. GZIP (DEFLATE Algorithm)

GZIP uses the DEFLATE method, integrating a sliding-window pattern matcher (LZ77) with smart variable-length codes (Huffman) to contract data without loss. In missile telemetry setups, teams often apply it at ground stations or for after flight reviews, thanks to its solid mix of size reduction and quick processing. The pattern matcher spots duplicate byte segments, while Huffman polishes the result with shorter tags for frequent items. It outperforms basic RLE, Huffman, or LZW alone, yet can fall short on missile data's tight structure and time-linked patterns.

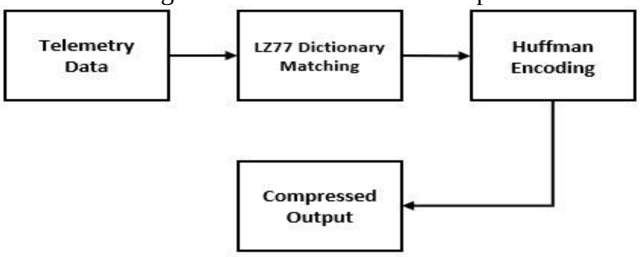

Fig. 9. GZIP operational steps

Fig. 9 Presents the workflow for GZIP shown below

- Collecting Data: Gather the full, unpacked dataset flow from synthetic sensors, mixing repeats and changes.
- LZ77 Dictionary Matching: Use a moving window to spot duplicate chunks nearby, swapping them for short distance and size pointers.
- Huffman Encoding: Turn those pointers and leftovers into variable-length bits short for commons, long for exceptional.
- Output: Produce the tight bitstream, exchanging some speed for good size cuts (less ideal for our structured data).

E. Lempel–Ziv–Markov Algorithm (LZMA)

LZMA is a smart lossless compression algorithm that builds on the classic LZ77 method. It adds clever probability predictions based on patterns (like Markov models) and efficient range encoding to pack data even tighter. In missile telemetry, it performs well at packing down huge streams of sensor information think inertial navigation status, guidance signals, and surroundings data. Those pattern-spotting tricks help it detect tricky links in the data, delivering much better compression than older approaches. That said, LZMA sticks to predetermined parameters and one data stream at a time. This can make it less flexible for managing mixed data types from various missile parts [14].

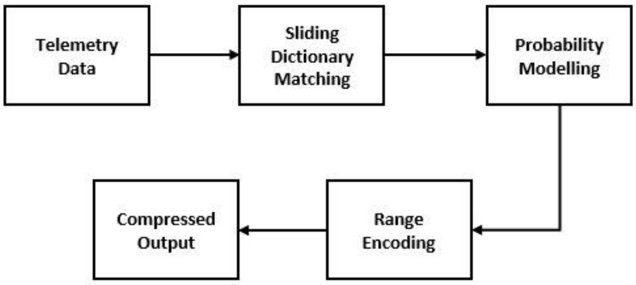

Fig. 10. LZMA operational steps

Fig. 10 Presents the workflow for LZMA shown below

- Input Raw Data: Take the raw dataset flow from replicated sensors, blending repeats and shifts..
- Sliding Dictionary Initialization: Start a big rolling window to hold recent patterns for quick repeat checks.
- Pattern Matching & Probability Modeling: Grab the longest matching chunks from the window; track odds of values based on what's come before.
- Range Encoding: Turn matches and singles into tiny fractional bits using smart probability adjustments.
- Compressed Output: Get a super-small bitstream beats basic methods but needs more power and space..

III. METHODOLOGY

This approach features a telemetry data processing pipeline that reduces transmission bandwidth while preserving full data accuracy. Raw signals from integrated sensors are first collected, then conditioned and sampled using analog to digital converters. The resulting digital values are packaged into a unified telemetry stream, organized into coordinated frames with explicit parameter marking for straight forward identification. Finally, the LZMA2 algorithm employing dictionary matching and interval coding provides superior lossless compression, yielding compact, reliable data suitable for bandwidth constrained channels.[15]

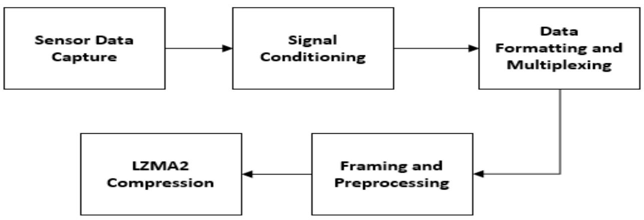

Fig. 11. General Telemetric Data Processing

Step 1: Sensors capture continuous real time parameters, building the foundational telemetry stream.

Step 2: Raw signals undergo filtering and amplification to eliminate noise, followed by sampling and digitizing for exact digital representation..

Step 3: Digital outputs convert to engineering units through adjustment, then multiplex into a single streamlined telemetry channel combining all sensor feeds.

Step 4: Data organizes into frames with synchronization bits, identifiers, timestamps, and contents. Preprocessing trims redundancy by eliminating, filling and condensing slowly varying values..

Step 5: Framed data splits into blocks for LZMA2 processing repeated sequences match against a dictionary, while unique literals store directly. Incompressible sections bypass compression, and range encoding yields the final lossless, compact stream.

LZMA2:

LZMA2 is an advanced lossless compression algorithm widely used for achieving high compression efficiency on structured data streams. LZMA2 mainly works by identifying repeated byte patterns using a dictionary based match searching technique and representing them in a compact form instead of storing the same data again.

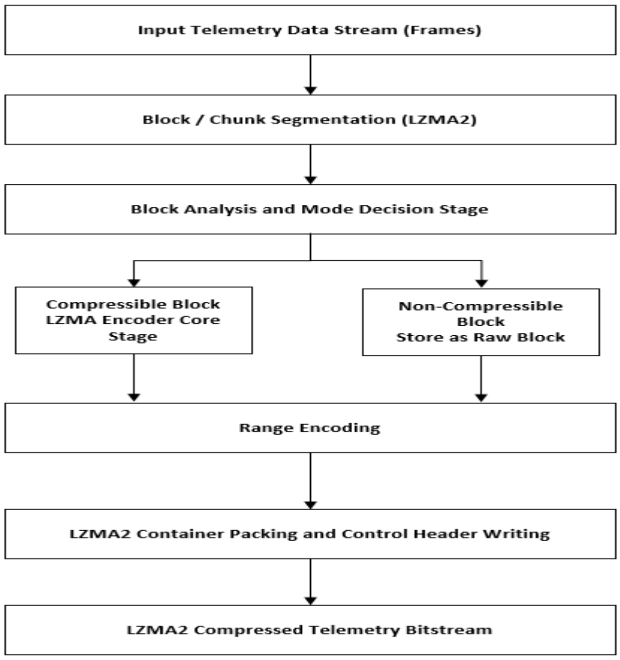

Fig. 12. Workflow of LZMA2

- Input Telemetry Data Stream: Telemetry frames from sensors form the initial byte stream input, complete with repeated headers, coordination bits, and payload values.
- Block / Chunk Segmentation: The stream breaks into manageable chunks. This division streamlines

processing and adapts to varying data patterns within the telemetry..

- Block Analysis and Mode Selection: Each chunk undergoes analysis to assess compression potential. Blocks showing clear repetition get processed further, while random or dense sections store in its current state to skip pointless overhead.
- LZMA Encoding (Dictionary Based Compression): For promising chunks, the encoder scans a dictionary buffer for duplicate sequences. Matches convert to compact length distance pairs unique bytes remain as direct literals.
- Range Encoding: Generated symbols from matches and literals feed into range encoding, minimising bit requirements and boosting overall density.
- LZMA2 Container Packing: Compressed and raw blocks pack into LZMA2 format, tagged with headers noting block types, sizes, and reset points for flawless unpacking..
- Final LZMA2 Output Stream: The assembled stream emerges as a dense, multi-block package primed for telemetry downlink channels.

A. *Dataset*

The synthetic dataset we created represents essential telemetry variables such as acceleration, temperature, pressure, voltage, and velocity components. It includes 90,000 minor frames organized into 9,000 major frames, where each major frame consists of 10 minor frames. Data within each minor frame gets sampled at 10-millisecond intervals, making one complete major frame cover 100 milliseconds..

IV. RESULTS AND DISCUSSION

For a thorough performance baseline, we evaluated both the standard GZIP compressor and our proposed LZMA2 on the 90,000-frame synthetic telemetric dataset. GZIP relies on the DEFLATE method, which blends LZ77 (Lempel-Ziv) dictionary matching with Huffman entropy coding. During testing, it processed the telemetry bitstream by hunting for repeated sequences inside a modest sliding window.

TABLE I. LZMA2 ALGORITHM OUTPUT ANALYSIS

Algorithm	**Archive size (kb)**	**Original size (kb)**
GZIP	10360	7340
LZMA2	10360	1768

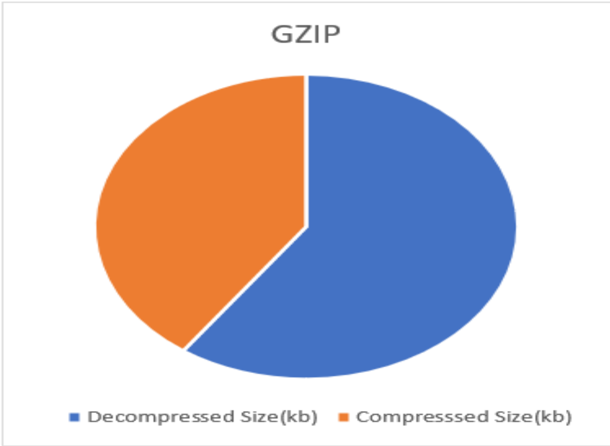

Fig. 13. Evaluation of GZIP Algorithm

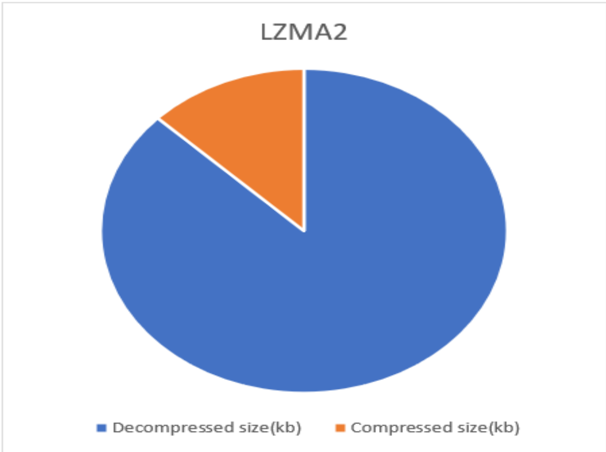

Fig. 14. Evaluation of LZMA2 Algorithm

TABLE II. Compression Efficiency Across Methods

Algorithm	**Compression Factor**
GZIP	1.41
LZMA2	5.58

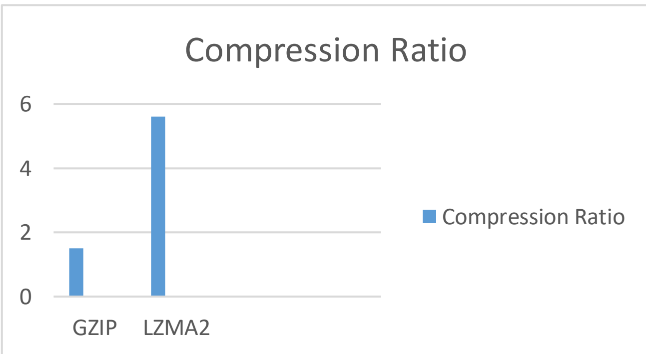

Fig. 15. Comparative study of algorithm compression efficiencies

As illustrated in Figures 11 to 13, LZMA achieves stronger compression results compared to GZIP, evidenced by its higher compression ratio. Tables I and II detail the LZMA algorithm's performance metrics, including compression ratios and data throughput, formatted for easy human readability.

Compression speed equals file size over processing duration shown in(1).

$$\text{Data rate } = \frac{size\ of\ File(MB)}{Time\ taken\ to\ compress(Sec)} \ (1)$$

TABLE III. RESULT EVALUATION OF DATA RATE

File Size	10 MB
Compression Time	190
Data Rate	0.052 MB/sec

Table III presents the calculated data rate of 0.052 MB/sec for the file.

Transmission Efficiency (TE) equals useful data bits divided by total transmitted bits, as indicated in equation (2).

$$TE = \frac{User\ data(bits)}{Total\ transmission\ size(bits)} \ (2)$$

TABLE IV. EVALUATION OF DATA RATE

User Information	10360
Total Transmission Size	10360
Transmission Efficiency	100.0%

Using data from Table IV, we eliminated the overhead bits from the total bits to isolate the pure user data payload. This adjustment revealed a transmission efficiency of 100% of the sent data reached its destination effectively.

CONCLUSION

During high-speed atmospheric flight, missile and spacecraft communication is severely affected due to the formation of a plasma sheath, which results in a communication blackout region. In this blackout phase, the telemetry link becomes unreliable and the continuous sensor information transmitted from the vehicle to the ground station may get disconnected, causing partial or complete data loss. This project implemented an efficient telemetry data compression framework that reduces the size of telemetry packets while maintaining data integrity, thereby enabling maximum data delivery even during blackout prone phases. Since telemetry contains highly critical mission parameters, it becomes necessary to decrease the size of transmitted information so that maximum information can be delivered with in the available link margin and limited bandwidth. The real missile telemetry data is confidential and difficult to obtain, a synthetic telemetry dataset was generated using key parameters such as acceleration, temperature, pressure, voltage, and velocity vector. The generated dataset consists of 90,000 minor frames, where each major frame contains 10 minor frames and represents a total duration of 100 milliseconds. The overall file size considered for experimentation was around 10 MB, which reflects a realistic telemetry stream. The literature survey evaluated the traditional lossless compression techniques, but LZMA2 was chosen for its superior balance of efficiency and adaptability. When we tested the algorithms on a telemetry dataset, LZMA2 exceeded GZIP performance, which reduced 10 MB of data to 1768 KB.

REFERENCES

[1] Accessed "https://www.indiatimes.com/trending/social relevance / brahmos - to- akash - here - is - indian -missile-list 588110.html" on 02.09.2023.

[2] Ryu, S. G., Jeong, J. J., & Shim, D. H. (2022). Sensor Data Prediction in Missile Flight Tests. Sensors, 22(23), 9410.

[3] Asokan, A., Anitha, J., Ciobanu, M., Gabor, A., Naaji, A., & Hemanth, D. J. (2020). Image processing techniques for analysis of satellite images for historical maps classification—An overview. Applied Sciences, 10(12), 4207

[4] Ramalingam, P., Mehbodniya, A., Webber, J. L., Shabaz, M., & Gopalakrishnan, L. (2022). Telemetry data compression algorithm using balanced recurrent neural network and deep learning. Computational Intelligence and Neuroscience, 2022.

[5] Ma, Z., Zhu, H., He, Z., Lu, Y., & Song, F. (2022). Deep Lossless Compression Algorithm Based on Arithmetic Coding for Power Data. Sensors, 22(14), 5331.

[6] Yılmaz, D., & Sağlam, A. (2017). The effects of data compression in missile telemetry systems. Journal of Aeronautics and Space Technologies, 10(2), 37-47.

[7] Sidhu, A. S., & Garg, M. (2014). Research Paper on Text Data Compression Algorithm using Hybrid Approach. International Journal of Computer Science and Mobile Computing, 3(12), 01-10.

[8] Stecuła, B., Stecuła, K., & Kapczyński, A. (2022). Compression of Text in Selected Languages—Efficiency, Volume, and Time Comparison. Sensors, 22(17), 6393.

[9] Oswald, C., Ghosh, A. I., & Sivaselvan, B. (2015, December). Knowledge engineering perspective of text compression. In 2015 Annual IEEE India Conference (INDICON) (pp. 1-6). IEEE.

[10] Maluf, D. A., Tran, P. B., & Tran, D. (2008, March). Effective data representation and compression in ground data systems. In 2008 IEEE Aerospace Conference (pp. 1-7). IEEE.

[11] Kingma, F., Abbeel, P., & Ho, J. (2019, May). Bit-swap: Recursive bits-back coding for lossless compression with hierarchical latent variables. In International Conference on Machine Learning (pp. 3408 3417). PMLR.

[12] Saidani, A., Jianwen, X., & Mansouri, D. (2020). A lossless compression approach based on delta encoding and T-RLE in WSNs. Wireless Communications and Mobile Computing, 2020, 1-10.

[13] Ullah, F., & Yahya, K. M. (2012). A new data compression technique using an evolutionary programming approach. In Emerging Trends and Applications in Information Communication Technologies: Second International Multi Topic Conference, IMTIC 2012, Jamshoro, Pakistan, March 28-30, 2012. Proceedings 2 (pp. 524-531). Springer Berlin Heidelberg.

[14] Nasif, A., Othman, Z. A., & Sani, N. S. (2021). The deep learning solutions on lossless compression methods for alleviating data load on IoT nodes in smart cities. Sensors, 21(12), 4223

[15] M. Zeidlitz, "Improving Compression Ratio in Backup," Linkoping ¨ University, Sweden, 2012.