Adaptive Entropy Prediction-based Lossless Compression for Efficient Data Transmission in Terrestrial and Underwater IoT Systems

Ankur Sisodia and Swati Vishnoi

Noida Institute of Engineering and Technology, Greater Noida, India

https://doi.org/10.26636/jtit.2026.3.2711

Submitted: 8 Jul. 2026, Reviewed: 17 Jul. 2026, Accepted: 13 Aug. 2026

Abstract — Demand for quick lossless data compression systems has become more important recently due to the growing deployment of Internet of Things (IoT) devices in terrestrial and underwater environments, under restricted bandwidth, latency and energy consumption conditions. Current dictionary-based and entropy-driven compression methods depend on reactive adaptation and fixed block processing, which makes them less useful in situations where sensing is constantly changing. This paper presents a self-optimizing entropy prediction-assisted lossless compression framework (EP-SLZW), where the compression technique is based on the expected data redundancy and current channel condition. To achieve optimal results, lightweight entropy prediction is employed in combination with adaptive block segmentation and dual dictionary learning between processing and compression. The optimization component of the proposed method enhances the compression strategy by taking care of all important environmental factors to provide the best solutions for radio frequency and acoustic communication channels. The method is evaluated with the help of the MQTT and CoAP protocols through NetSim simulations and hardware experiments. The results show that the proposed approach performs better in comparison to conventional LZW as well as Huffman and run-length encoding techniques, providing better throughput, compression ratio, and less end-to-end delay when dealing with resource-constrained IoT devices.

Keywords — adaptive dictionary, IoT networks, lossless data compression, prediction of entropy

1. Introduction

The Internet of Things (IoT) has become a crucial technology enabling various intelligent systems, including smart city infrastructure, precision agriculture, industrial control, and environmental monitoring. Systems of this type depend on extensive sensor networks which gather information to support automatic management and intelligent decision processes. IoT networks support multiple communication modes, including terrestrial RF systems and underwater acoustic communication. Networks must process sensor data on a continuous basis, facing restrictions regarding their bandwidth, energy usage, and response time. Low channel capacity, high propagation delays, signal attenuation, and rapidly changing ambient noise levels create additional difficulties for underwater networks. Efficient data processing becomes a critical requirement that operators need to take into consideration in order to maintain integrity of their networks [1], [2].

Lempel-Ziv-Welch (LZW), Huffman data coding, and run-length encoding are traditional compression techniques users widely implement due to their straightforward nature and good performance [3]. These methods need fixed block structures, or require real-time data statistics for their adaptive mechanisms, which makes it difficult to handle the rapid changes in data patterns occurring during sensing operations in IoT. Sensor readings show two main patterns, as they experience periods of strong repeated measurements and unexpected new values that result from environmental changes and various other events. The use of static strategies decreases compression efficiency and leads to higher cost of communication.

Researchers have studied adaptive compression systems that adjust their block size and dictionary formats according to the detected entropy and temporary redundancy patterns. Most existing methods continue to use fixed settings because they perform better than established methods, but they still function as reactive systems changing their parameters after major data transformations [4]. Moreover, systems use identical compression methods on all types of communication channels because they are not capable of categorizing the operational characteristics of terrestrial and underwater communication. A system's inability to anticipate its working environment leads to excessive data retransmissions that extend total response times and waste energy resources, particularly in underwater IoT networks, where retransmission expenses are extremely high [5].

This work presents a novel lossless compression method which uses a self-optimizing system with entropy prediction to solve connection problems between different terrestrial and underwater IoT networks. The proposed solution uses lightweight entropy prediction to facilitate predictive adaptation, which enables the compression engine to adapt to changes in the redundancy of the incoming data. The system achieves an equilibrium between three competing demands, i.e. compression efficiency, computational resources, and energy usage, by implementing predictive adaptation together with dynamic block segmentation and dual dictionary learning. It uses an environment-aware optimization strategy to modify its compression operations based on the distinct radio frequency and acoustic channel characteristics.

The proposed method was evaluated through performance testing that relied on NetSim simulations and hardware tests with multiple IoT sensor nodes using the MQTT and CoAP communication protocols. The suggested approach is compared with existing lossless compression techniques using parameters such as compression ratio, throughput, end-to-end delay, and computational complexity. The experimental results show that the proposed EP-SLZW method performs better than the existing approaches while being suitable for resource-constrained IoT devices. The research work presents the following main contributions, which are described below:

1) A self-adaptive design with lightweight entropy prediction is implemented to forecast data redundancy changes during different sensing conditions. This capability is used to adjust the compression method before the data stream experiences significant statistical changes.
2) A predictive block size control method is proposed that regulates data compression by monitoring changes in entropy patterns. This approach is used to make intelligent decisions about block size selection, which leads to better compression results while keeping extra processing costs at a minimum.
3) Local and global dictionaries are implemented to manage memory consumption and speed of convergence while achieving the appropriate compression ratio for unevenly distributed data.
4) The framework uses its control system to switch between different data compression methods, depending on the terrestrial radio frequency and underwater acoustic channels.
5) The proposed framework is evaluated through NetSim simulations and testing on actual multi-sensor IoT hardware which uses both MQTT and CoAP protocols. The system demonstrates improved compression ratios, throughput and end-to-end delay performance compared to traditional LZW and Huffman coding and run-length encoding methods.

2. Related Works

Data compression is a key method to make communication in IoT networks more efficient, especially when bandwidth and energy are limited. The early data processing relied on classical lossless algorithms, including Huffman coding, run-length encoding and Lempel-Ziv-Welch (LZW), as these algorithms offer straightforward implementation and produce consistent decoding results [6]. Such methods are used in low-power sensing platforms, but their effectiveness depends on the data characteristics and the specific configuration of the sensor blocks. Sensors in multiple IoT applications demonstrate statistical patterns that change over time. The typical applications of fixed block sizes and static dictionary structures face reduced effectiveness in this concept [7].

Researchers have proposed several adaptive compression techniques that adjust block size and encoding settings based on actual entropy measurements and symbol distribution patterns. The compression ratio for image and sensor data streams has improved through entropy-based block segmentation, which works best in regions that exhibit repetitive patterning [8]. Compressive sensing was also explored together with block-based adaptive coding methods for transmitting underwater images and enhancing the performance of acoustic sensor networks under restricted bandwidth conditions. Other methods adjust compression parameters when substantial alterations occur in data patterns, leading to enhanced efficiency in data reduction [9]. This leads to temporary performance issues because it takes time to react, which affects applications that depend on instant detection of quick changes in data.

Researchers have shown great interest in finding better ways to improve dictionary-based compression methods. The development of LZW and LZ77 compression algorithms which use dynamic dictionary pruning, context modelling and memory-aware management, helps achieve faster convergence and improved compression [10]. Sliding-window dictionaries are researched as a solution to control memory usage, while other work examined hierarchical or clustered dictionaries for IoT systems that operate in distributed environments. Most dictionary optimization methods still need to use local data, since they cannot handle long-term data patterns that occur between data blocks or forecast upcoming data trends. The system becomes less adaptable, because it cannot handle multiple data types that exist together in different environments [11].

An underwater IoT (U-IoT) network has emerged recently as a research domain due to the fact that these networks face particular communication challenges, including extended propagation delays, high signal loss, and restricted transmission capacity. Underwater environments rely on compression methods that prioritize energy efficiency and latency reduction over maximum compression ratio [12]. In the literature, three different approaches are presented, including acoustic channel-aware encoding schemes, predictive data aggregation, and hybrid compression-routing methods to decrease retransmissions and packet loss. Most of these methods depend on specific physical layer assumptions and routing protocols that restrict their application to mixed terrestrial and underwater operations [13].

Current research investigates data compression methods that employ basic forecasting models and neural predictors to determine future data behavior [14]. Most existing predictive systems currently work to estimate traffic patterns or develop optimal routing solutions that do not integrate with the data compression process. This research fails to investigate dictionary adaptation techniques and compression policies tailored to specific environments, creating opportunities for better optimization across all system layers and different operational environments [15], [16].

Tab. 1. Comparison of existing IoT compression approaches.

Method	Adaptive block size	Entropy prediction	Dictionary learning	Environment aware (terrestrial, underwater)	Energy and delay optimization	Key limitations
Run-length encoding (RLE)	No	No	No	No	No	Poor performance for non-repetitive data
Huffman coding	No	No	No	No	Partial	Fixed coding tree, sensitive to data distribution
Classical LZW	No	No	Static	No	Partial	The dictionary grows blindly, no anticipation
Entropy-based adaptive compression	Yes	No	Limited	No	Partial	Reactive adaptation only
Compressive sensing (IoUT)	Yes	No	No	Partial	Yes	High reconstruction complexity
Predictive data aggregation	No	Yes	No	Partial	Yes	Not integrated with compression stage
Dynamic dictionary LZ variants	Partial	No	Yes	No	Partial	No environment awareness
Adaptive LZW	Partial	No	Adaptive	No	Partial	Limited adaptation to communication environments
Trie-based shared dictionary compression	No	No	Shared trie dictionary	No	Partial	Higher memory requirement and synchronization overhead
Proposed	Yes	Yes	Dual (local+global)	Yes	Yes	—

While recent studies propose enhanced compression techniques such as dynamic dictionary learning, dictionary compression, predictive aggregation of data, and advanced lightweight entropy-based encoding developed specifically for the edge and IoT architectures, there is still an inherent necessity for additional memory or computing power, or for pre-learning which is problematic in the case when the microcontroller capabilities are limited. Table 1 shows a comparison of existing IoT compression approaches.

3. System Model and Problem Formulation

The study investigates IoT networks containing various types of sensor nodes that operate in both terrestrial and water environments. The terrestrial segment uses radio frequency (RF) communication, while the underwater segment uses an acoustic or short-range optical link for sending data. Sensor nodes which form group $\mathcal{N} = \{n_1, n_2, \ldots, n_N\}$ operate continuously to monitor environmental conditions by measuring temperature, humidity, pressure, light intensity, and distance. The nodes transmit their detected data to a sink or gateway node by using multi-hop or single-hop communication through MQTT and CoAP.

The network function during time intervals $t = 1, 2, \ldots$, which each node uses to generate its data stream is:

$$D_i(t) = \{x_1, x_2, \ldots, x_{L_t}\} , \tag{1}$$

where L_t is the number of samples taken at time t.

3.1. Data and Channel Model

The data stream being detected operates like a separate system that depends on memory functions, while its statistical characteristics develop over time due to the fact that environmental factors and event-based behavioral patterns evolve. The instantaneous entropy of a data block B_t at time t is defined as:

$$H(B_t) = -\sum_{k=1}^{K} p_k \log_2 p_k , \tag{2}$$

where p_k represents the probability of symbol k appearing in B_t and K defines the total number of symbols in the alphabet.

The communication channel exhibits major variations which depend on different environmental conditions. The terrestrial IoT system operates with a data transmission capacity as follows:

$$R_t = B_t \log_2(1 + \mathrm{SNR}_T) , \tag{3}$$

where B_T is the channel bandwidth and SNR_T is the signal-to-noise ratio. Underwater data transmission capabilities operating at lower speeds are defined as:

$$R_U = B_U \log_2(1 + \mathrm{SNR}_U) , \tag{4}$$

with conditions $B_U \ll B_T$ and $\mathrm{SNR}_U < \mathrm{SNR}_T$.

3.2. Compression Model

Every node applies a lossless compression function $\mathcal{C}(\cdot)$ before sending data. The original data size is represented by S_o bits while the compressed data size is represented by S_c. The compression ratio is calculated as:

$$C_R = \frac{S_o}{S_c} .$$

The duration of compression processing is represented by T_c and the transmission time is calculated as:

$$T_{tx} = \frac{S_c}{R} .$$

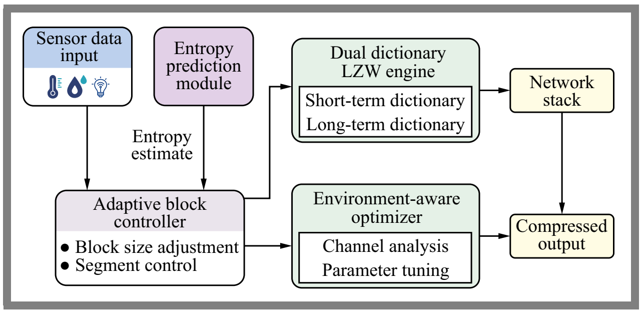

Fig. 1. EP-SLZW framework architecture.

The setting of R determines which value from $R \in \{R_T, R_U\}$ to use. The complete end-to-end delay calculation establishes $T_{e2e} = T_c + T_{tx} + T_{prop}$, where T_{prop} represents the duration required for data to traverse through intermediate nodes.

3.3. Energy Consumption Model

The total energy consumption for each data block is as follows:

$$E_{total} = E_{comp} + E_{tx} \ , \tag{5}$$

where compression energy is $E_{comp} = P_{cpu} \cdot T_c$, and transmission energy is $E_{tx} = P_{tx} \cdot T_{tx}$.

The system uses P_{cpu} as the CPU power consumption and P_{tx} as the transmission power consumption. Underwater networks use higher P_{tx} to operate due to specific environment conditions.

3.4. Problem Formulation

The compression framework manages three components which determine its performance. The optimization problem is stated as max C_r min T_{e2e}, E_{total} and

$$\mathcal{D}\big(\mathcal{C}(D_i(t))\big) = D_i(t), \ \ \forall i, t \ , \tag{6}$$

where:

- $B_{min} \leqslant B_t \leqslant B_{max}$,
- $T_c \leqslant T_{max}$,
- $\mathcal{D}(\cdot)$ is the function that decompresses,
- B_t is the adaptive block size,
- B_{min} and B_{max} are the limits on the size of blocks that can be used,
- T_{max} is the longest time that a real-time application can wait for compression.

4. Proposed Framework

The proposed entropy prediction-assisted self-optimizing LZW (EP-SLZW) framework is a data processing layer for sequential block raw sensor streams. It combines entropy prediction, adaptive block segmentation, dual-dictionary compression, and environment-aware optimization into a single system, as shown in Fig. 1.

Initially, sensor readings are buffered and put into preliminary data blocks. The system determines present entropy H_t value for every block at time t to predict future data redundancy changes. EP-SLZW uses more than just reactive entropy assessments, as it implements a basic forecasting system relying on an exponential weighted moving average (EWMA) model to predict subsequent block entropy in the following manner:

$$\hat{H}_{t+1} = \alpha H_t + (1 - \alpha)\, \hat{H}_t \ , \tag{7}$$

where H_t is determined entropy and $\hat{H}_t$ is previous predicted entropy.

The range of $0 < \alpha < 1$ determines the degree to which the predictor reacts to new information. The predicted entropy helps the framework prepare for changes in data randomness by varying the compression settings to keep things running smoothly until performance drops excessively. The block segmentation strategy uses the predicted entropy value $\hat{H}_{t+1}$, to determine the size of the upcoming data block. The current block size is represented by B_t. The update rule is as follows:

$$B_{t+1} = \begin{cases} \min(B_t + \Delta, B_{max}), & \text{if } \hat{H}_{t+1} < \tau \\ \min(B_t - \Delta, B_{min}), & \text{if } \hat{H}_{t+1} \geqslant \tau \end{cases} \ , \tag{8}$$

The operational constraints of the proposed framework are determined by the relationship between entropy threshold τ and block adjustment factor Δ, which define the range of the block size bounded by B_{min} and B_{max}. The data stream becomes extremely redundant when entropy values drop below the threshold, which leads to selecting larger block sizes that maximize dictionary reuse. The framework selects smaller block sizes, because rising predicted entropy values require dictionary size control, which accelerates compression to prevent performance drops during times of high data randomness.

The proposed framework achieves its primary extension through the implementation of dual-dictionary LZW compression, which provides exceptional flexibility to handle diverse data patterns. Two dictionaries are used – as a local dictionary and a global dictionary. The local dictionary reset function at block start enables quick adaptation to unexpected changes that occur in sensor data patterns. On the other hand, the global dictionary is retained over several blocks, which enables the detection of slowly varying or periodic sensor data patterns over long time periods. The dictionary selection process uses a normalized entropy factor that defines the process as:

$$\eta_t = \frac{H_t}{H_{max}} \ , \tag{9}$$

where H_{max} denotes the maximum possible entropy of the symbol alphabet.

Based on this metric, the dictionary selection rule is given by:

$$\text{Dictionary} = \begin{cases} \text{global}, & \eta_t < \gamma \\ \text{local}, & \eta_t \geqslant \gamma \end{cases} \ ,$$

The two operational modes for EP-SLZW are activated through γ which acts as the boundary between these two modes. The algorithm uses two types of redundancy patterns to store data. The dictionary size increases only during periods in which system activity becomes less predictable,

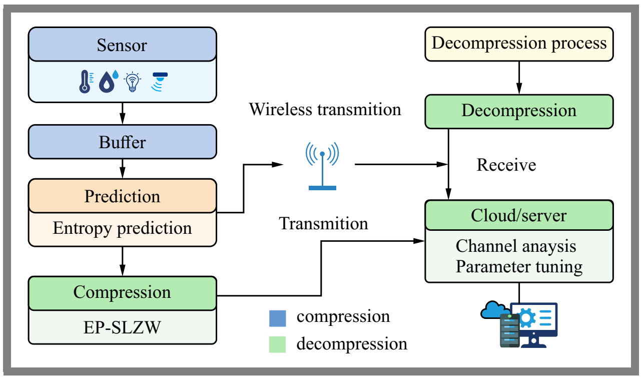

Fig. 2. Data flow of the proposed compression-decompression process.

because this method enables to store both temporary and permanent data. The method uses an environment indicator that helps it understand the different IoT communication environments. The numerical values of all hyperparameters (α, τ, Δ, and γ) and the methodology used for their selection are presented in Sec. 6.

Terrestrial communication systems use RF signals which assign an indicator value of $\lambda = 0$. Underwater communication systems which use acoustic or optical links assign an indicator value of $\lambda = 1$. The approach achieves its target goal through high processing throughput combined with moderate compression latency. Aggressive data compression techniques are used for underwater communication to save energy and bandwidth. The system uses environmental conditions to dynamically adjust its values τ, Δ, and γ.

The receiver uses standard LZW decompression to extract data from compressed blocks, which includes synchronous updates of their dictionaries. The proposed compression framework maintains lossless data recovery, which ensures accurate retrieval of original data. The compressed metadata inform the decoder which dictionaries to use, which lets it do the same thing with two dictionaries. Block decompression enables the restoration of original data streams $D' = \mathcal{D}(\mathcal{C}(D)) = D$.

The data processing scheme of the proposed compression-decompression process is presented in Fig. 2.

Algorithm 1 Proposed compression framework

Input: Sensor data stream D, initial block size B_0, block size bounds B_{min}, B_{max}, entropy threshold τ, dictionary switching threshold γ, prediction factor α, environment indicator $\lambda \in \{0, 1\}$
Output: Compressed data stream C
Initialize:

Set $t \leftarrow 0$
Set $B_t \leftarrow B_0$
Global dictionary D_g
Set predicted entropy $\hat{H}_0 \leftarrow H_0$
Compressed stream $C \leftarrow \emptyset$
while $D \neq \emptyset$ **do**
  Extract block B_t from D of size B_t
  Compute the block entropy as
  $H(t) = -\sum_{k=1}^{K} p_k \log_2 p_k$
  Predict the next entropy using EWMA by Eq. (7)
  Compute the normalized entropy by Eq. (10)
Dictionary selection
  if $\eta_t < \gamma$ **then**
    select global dictionary D_g
  else
    initialize local dictionary D_l
  end if
  Block size adaptation by Eq. (8)
Tunning of environment-aware parameter
  if $\gamma = 1$ (underwater) **then**
    increase τ, decrease Δ
  else
    maintain default values
  end if
  Compress block B_t using selected dictionary with LZW procedure to obtain C_t
  Append C_t to C
  Update predicted entropy $\hat{H}_t \leftarrow \hat{H}_{t+1}$
  Increment $t \leftarrow t + 1$
end while
Return compressed stream C

5. Computational Complexity and Energy Analysis

This section evaluates the proposed EP-SLZW framework by assessing its memory requirements, processing demands and energy consumption in order to determine its suitability for IoT devices with restricted resources operating in both terrestrial and underwater environments. The size of the data block is defined by its symbol count n and its dictionary size M. The first block needs to be scanned to determine the symbol frequency because this information is necessary to calculate the entropy for each block. The process requires $O(n)$ time to complete. The prediction of entropy is based on the exponential weighted moving average model.

The dual-dictionary LZW compression process is the most important part of the framework. The dictionary operations use hash-based indexing enable dictionaries to process every symbol in their contents during dictionary lookup and insertion functions at a fixed time interval. The compression process requires $O(n)$ time, as it handles all blocks of data. The LZW decoding process requires standard decompression to complete in $O(n)$ time, which operates at linear speed based on the block size. The framework processes each block with a total computational load of $O(n)$ which matches the standard LZW and Huffman methods, because they both show the same asymptotic performance. The additional prediction and control systems add minimal workload to the system while maintaining its ability to scale efficiently.

The memory needed for EP-SLZW includes space for the input block, entropy statistics, and two dictionaries. Let M_g and M_l be the sizes of the global and local dictionaries, respectively. The total amount of used memory is $O(n + M_g + M_l)$. The local dictionary is cleared after each block operation.

Algorithm 2 Proposed decompression framework

Input: Compressed data stream C, initial block size B_0, block size bounds B_{min}, B_{max}, dictionary switching threshold γ, entropy prediction factor α, environment indicator $\lambda \in \{0, 1\}$
Output: Sensor data stream D' is reconstructed
Initialize:

```
Set t ← 0
Set B_t ← B_0
Initialize global dictionary D_g
Set predicted entropy Ĥ_0 ← H_0
Initialize reconstructed stream D' ← ∅
while compressed stream C is not empty do
    Extract compressed block C_t from C
    Retrieve the dictionary selection flag
     from metadata of C_t
  Dictionary initialization
    if flag indicates global dictionary then
        use D_g
    else
        initialize local dictionary D_l
    end if
    Apply the standard LZW decoding on C_t
     using selected dictionary to obtain block B_t
    Append B_t to reconstructed stream D'
    Compute the block entropy using:
    H(t) = - Σ_{k=1}^{K} p_k log_2 p_k
    Predict the next entropy using EWMA by Eq. (7)
    Block size adaptation by Eq. (8)
    Update predicted entropy Ĥ_t ← Ĥ_{t+1}
    Increment t ← t + 1
end while
Return compressed stream D'
```

The system maintains memory balance by periodically removing elements from the global dictionary. The memory consumption remains controlled since the system limits its usage to microcontroller platforms which operate with minimal power in common IoT devices. The total energy used per block is modeled in the following way:

$$E_{total} = E_{comp} + E_{tx} \;, \tag{10}$$

where E_{comp} represents the energy required for compression work and E_{tx} denotes the transmission energy used for wireless data transmission.

The energy required for compression work is:

$$E_{comp} = P_{cpu} \cdot T_c \;. \tag{11}$$

The energy used for transmission is determined using P_{cpu} being the average amount of power that a processor uses and T_c – the time it takes to compress each block. The energy used for transmission is derived as follows:

$$E_{tx} = P_{tx} \cdot \frac{S_c}{R} \;. \tag{12}$$

The formula describes transmission power P_{tx} as the product of S_c (representing compressed data size) and R (denoting the maximum achievable channel data rate).

The processing power requirements of EP-SLZW increase, because the scheme needs additional resources for entropy prediction and dictionary management, yet its superior compression capabilities result in S_c being reduced. The underwater IoT environments demonstrate this effect with increased strength

$$P_{tx}^{(U)} \gg P_{tx}^{(T)} \;, \;\; R_U \ll R_T \;.$$

The system uses U and T as superscripts to identify underwater and terrestrial environments. Higher energy savings through are achieved data volume reduction, as EP-SLZW transmits less data than it needs to send at E_{tx} while E_{comp} experiences only a minimal rise:

$$E_{total}^{EP-SLZW} < E_{total}^{baseline} \;.$$

The adaptive block size enables precise management of dictionary size changes that occur during times of high data stream entropy, since it prevents dictionary expansion during random sensing intervals. The system maintains minimal processor usage, while it solves energy stability problems which emerge when data patterns occur abruptly.

6. Experimental Setup

To evaluate the performance of the EP-SLZW framework, we used two testing methods, i.e. simulation testing and hardware testing, across both terrestrial and underwater IoT environments. The NetSim simulator executes network simulations to create large-scale IoT network models that assess system performance under various operational conditions. Simulation tests use a 1000 × 1000 m sensing area while testing multiple sensor node configurations (ranging from 50 to 500) to determine how network density affects its performance.

Sensor nodes are deployed randomly inside the sensing area and send data using a fixed rate of five packets (carrying 1024 bytes) per second (Figs. 3, 4). The study evaluates how application layer communication protocols impact compression performance through the use of MQTT and CoAP protocols. The terrestrial network maintains a constant wireless bandwidth of 2 Mbps throughout the entire simulation period which extends from 250 s to 750 s, as shown in Tab. 2. Underwater simulations test different physical layer settings to create an acoustic channel simulating reduced bandwidth, extended propagation delay, and increased packet loss rates. Identical routing and medium access conditions throughout their experiments are maintained to enable unbiased assessment of compression algorithm performance.

The proposed framework functions at the sensor node layer, and its effectiveness gets tested through a comparison with standard lossless compression techniques, i.e. LZW and Huffman coding and run-length encoding (RLE). The NetSim protocol stack includes all algorithms that operate with identical configuration settings for packet creation, network routing and data storage.

An additional evaluation was performed using a physical IoT testbed comprising a low-power microcontroller platform, an Arduino Uno board with multiple sensors (including

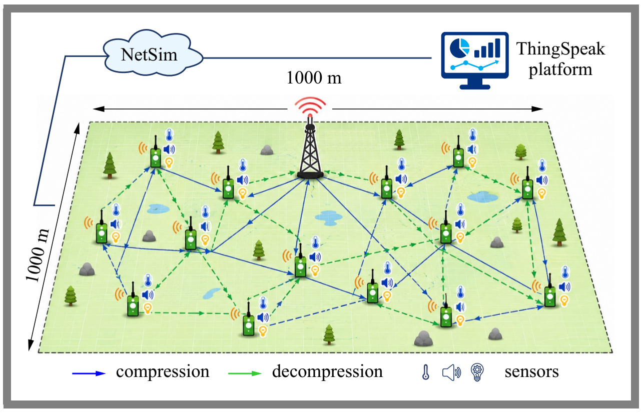

Fig. 3. Topology of terrestrial IoT simulation (NetSim).

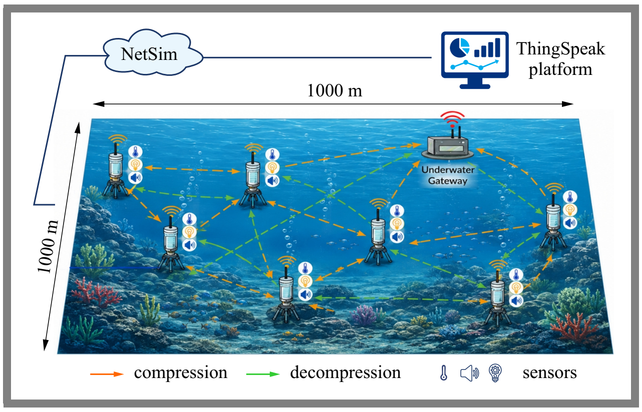

Fig. 4. Underwater IoT simulation topology (NetSim).

a temperature-humidity, an ultrasonic distance, and a light intensity sensor) (Tab. 3). The framework operates through periodic sensor reading that is processed at the current location before forwarding the data. The research simulates restricted communication environments through an infrared transceiver module which enables data to be transferred wirelessly at short distances, since RF signals face challenges in underwater transmission. The receiver system transmits uncompressed data to an ESP8266 Wi-Fi module that forwards the information to a cloud server, where it may be accessed by the ThingSpeak platform. Such an approach enables real-time data logging and visualization.

The same hardware setup is used for terrestrial and underwater tests. Underwater conditions sensor modules and IR transceivers are positioned inside waterproof cases that partially submerge to simulate underwater noise and attenuation effects.

Although the infrared (IR) transceiver employed in the hardware prototype in the underwater communication experiment can be attributed to the use of the channel model, it is not aimed at simulating all the properties of the underwater acoustic channel. Instead, it is supposed to serve as an illustration of the application of the compression framework in the underwater communication problem with limited transmission distance, attenuation, and bandwidth. Therefore, features such as delay, attenuation, and multipath propagation are modeled separately in NetSim. Consequently, the outcomes of the experiment can be interpreted as the demonstration of

Tab. 2. Simulation parameters for terrestrial and underwater IoT networks.

Parameter	Value
Simulation platform	NetSim
Simulation area	1000 × 1000
Number of sensor nodes	50 – 500
Node deployment	Random
Application protocols	MQTT, CoAP
Packet size	1024 bytes
Packet generation rate	5 packets/s
Simulation time	250 – 750 s
Terrestrial bandwidth	2 Mbps
Underwater bandwidth	Low (acoustic channel model)
Routing configuration	Fixed for all experiments
Compression algorithms	EP-SLZW, LZW, Huffman, RLE
Performance metrics	Compression ratio, throughput, end-to-end delay, compression time
Channel type	RF (terrestrial), acoustic (simulation), IR-based proof-of-concept (hardware)

Tab. 3. Hardware testbed configuration.

Component	Model / Specification	Purpose
Microcontroller	Arduino Uno (ATmega328P)	Sensor interfacing and compression processing
Temperature–humidity sensor	DHt11	Environmental data acquisition
Ultrasonic sensor	HC-SR04	Distance measurement
Light sensor	LDR module	Light intensity detection
IR transmitter	940 nm IR LED module	Short-range wireless data transmission
IR receiver	TSOP38238 / equivalent	Short-range wireless data reception
Display module	LCD 16 × 2	Local monitoring of sensor readings
Wi-Fi module	ESP8266	Cloud connectivity (ThingSpeak)
Power supply	5 V regulated + battery backup	Stable system operation
Cloud platform	ThingSpeak	Data storage and visualization
Communication protocols	MQTT, CoAP	Application-layer data transmission

the application of the compression framework, and not the complete underwater acoustic communication system.

Four metrics are used to evaluate performance: compression ratio, throughput, end-to-end delay, and compression time. The experiments were carried out 30 times, in an independent manner, with each iteration relying on different seeds. Nevertheless, the same network configurations and traffic conditions were applied. The data shows the average standard deviation (SD) values of the results of 30 independent experiments. A confidence interval (CI) of 95% was also estimated for all metrics. Finally, the paired Student t-test was performed on the results of repeated experiments to assess the significance of the results obtained with respect to the baseline methods.

Tab. 4. Hyperparameter configuration of EP-SLZW.

Parameter	Value	Description	Selection method
α	0.35	EWMA smoothing coefficient	Grid search on validation dataset
τ	0.55	Entropy threshold for adaptive block sizing	Validation experiments
Δ	64 bytes	Block size adjustment step	Sensitivity analysis
γ	0.60	Threshold for local/ global dictionary switching	Empirical tuning
B_{min}	256 bytes	Minimum allowable block size	Memory constraint of sensor nodes
B_{max}	4096 bytes	Maximum allowable block size	Compression efficiency and latency trade-off

6.1. Hyperparameter Configuration

To ensure repeatability of the results obtained during the simulations and to achieve an optimal balance between compression ratio, processing time, and delivery delay, the parameter tuning has been performed. Several parameter values have been tested on the validation set, and the final combination has been chosen according to the maximum compression rate along with minimum processing time and delay. Table 4 shows the hyperparameter values utilized during all simulations and hardware tests.

7. Results and Discussion

The proposed system achieves higher compression ratios and better throughput efficiency while reducing end-to-end latency compared to all baseline algorithms tested. Both terrestrial RF networks and underwater communication systems with restricted bandwidth prove effective with the developed improvements which work in simulation environments and physical testing facilities. The EP-SLZW system achieved an average compression ratio of 7.68 during terrestrial simulation experiments, as shown in Tab. 5. The average compression ratios for Huffman coding, LZW and RLE equaled 4.87, 3.92, and 2.41, respectively. Throughput exceeds standard LZW performance by 93% and RLE performance by nearly 200%. The end-to-end delay time is 21% shorter when using Huffman coding and 26% shorter when using LZW, in comparison to other methods. The compression process requires additional time because of the need to execute operations that predict entropy while handling dictionary management tasks.

The underwater simulation results, shown in Tab. 6, illustrate how performance differences between systems become more evident. EP-SLZW achieves a compression ratio of 9.41, which exceeds (two-fold) the compression ratio of Huffman coding, yet reaches almost three times the compression ratio of RLE. The system provides throughput performance improved by more than 115%, exceeding the throughput of LZW, and it delivers end-to-end delay which exceeds that of RLE and LZW by 55 and 44%, respectively. The results demonstrate that

Tab. 5. Performance comparison in terrestrial simulation (average values).

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	2.41	0.98	1820	940
Huffman	4.87	1.52	1340	1710
LZW	3.92	1.36	1495	1540
EP-SLZW	7.68 ±0.12	2.94 ±0.08	1185 ±28	1825 ±42

Tab. 6. Performance comparison in underwater simulation (average values).

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	2.05	0.62	8120	960
Huffman	4.12	1.05	5280	1830
LZW	3.46	0.89	6125	1615
EP-SLZW	9.41 ±0.14	1.91 ±0.07	2715 ±25	3085 ±45

Tab. 7. Performance comparison using a terrestrial hardware testbed.

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	2.36	0.83	2150	1015
Huffman	4.54	1.27	1610	1935
LZW	3.71	1.12	1785	1680
EP-SLZW	6.92 ±0.11	1.88 ±0.09	1325 ±24	2150 ±41

Tab. 8. Performance comparison using an underwater hardware testbed.

Algorithm	Compression ratio	Throughput [kbps]	End-to-end delay [ms]	Compression time [ms]
RLE	1.98	0.59	4980	1120
Huffman	3.87	0.94	3510	2040
LZW	3.25	0.86	3820	1815
EP-SLZW	6.47 ±0.16	1.54 ±0.06	2630 ±23	2890 ±43

underwater environments experiencing low channel capacity and high retransmission costs benefit the most from aggressive data size reductions.

The proposed framework shows increased stability under actual conditions, and hardware tests demonstrate its performance during sensor noise processing limits and unstable wireless links. The method maintains a compression ratio of 6.92 in land tests (Tab. 7), while achieving a latency that is 26% lower than that of LZW. 79% higher throughput is achieved than in the case of LZW in underwater hardware tests (Tab. 8), with delay reduced by 31%.

7.1. Interpretation of performance gains

The designed system improves performance, as it enables better handling of smaller data payloads. Smaller data packets use less bandwidth, which means that there are fewer network collisions and a lower chance of having to resend data. The experimental results demonstrate that underwater IoT networks maintain consistent performance across different packet sizes, which shows their ability to handle various traffic patterns. The experimental results demonstrate that the

Tab. 9. Ablation study results.

Configuration	Compression ratio	Throughput [kbps]	End-to-end delay [ms]
Fixed block + single dictionary	3.85	1.34	1510
Adaptive block only	5.21	1.87	1375
Adaptive block + dual dictionary	6.43	2.42	1260
Without entropy prediction	6.71	2.63	1235
Without environment awareness	7.12	2.72	1274
EP-SLZW	7.68	2.94	1185

proposed framework achieves energy savings that exceed the energy expenses that EP-SLZW introduces. Underwater IoT networks benefit from this because wireless data transmission requires more energy than local computation.

7.2. Ablation Study

To assess the impact of each of the key components in the proposed framework, an ablation study was conducted using the terrestrial simulation setup (Tab. 9).

The system demonstrates decreased processing efficiency along with increased execution latency when the compression algorithm lacks entropy prediction, which shows that dynamic data environments need predictive adaptation. To continue with the analysis of the environment-adaptive optimization system, the environment-adaptive setting was turned off while the remaining elements, such as entropy value estimation, the blocks of adaptive splitting, and dual dictionary compression, remain working. In this case, it should be noted that the compression parameters of both terrestrial and underwater communication channels remained the same, regardless of their specifications. One may notice from Tab. 9 that the algorithm with the disabled environment-adaptive module managed to compress the data achieving a compression ratio of 7.68, instead of 7.12, which also allowed achieving the throughput of 2.94 kbps compared with 2.72 kbps when the environment-adaptive module was in use. The average end-to-end delay has increased from 1185 ms in the case of the disabled environment-adaptive module to 1274 ms when the algorithm with enabled environment-adaptive optimization technique was used. Therefore, using the adaptive optimization system improved overall system performance. The contribution of the adaptive optimization system to the increase in overall throughput equaled 8%, while the change of the average end-to-end delay amounted to 7%. The proactive adaptation methods of the EP-SLZW framework combat the performance decline that typically occurs during high-rate data transmissions.

To assess if the improvements in performance obtained did not occur due to the randomness of the experiments, all simulations and experiments were carried out 30 times under the same conditions, but with different random seeds. The mean standard deviation, and 95% confidence interval for all the metrics were computed. The paired t-test was then carried out for the proposed EP-SLZW method and the old LZW algorithm, which served as a benchmark. The p-values for all metrics remained below the level of 0.05, indicating that the improvement obtained with EP-SLZW is statistically significant, as seen in Tab. 10.

Tab. 10. Statistical significance analysis.

Performance metric	Mean improvement	95% CI	p-value
Compression ratio	95.9%	93.8 – 98.0	<0.001
Throughput	116.2%	112.4 – 119.7	<0.001
End-to-end delay	20.7%	18.9 – 22.5	<0.001
Compression time	+18.5%	16.8 – 20.1	0.003

The low values of standard deviation and the narrow confidence intervals prove that the proposed method is functionally stable. Furthermore, the results of the statistical tests indicate that the advantages in the compression rates, speed of operation, and end-to-end lag are statistically significant at a 95% level of confidence.

7.3. Comparison with Recent Adaptive Compression Methods

Besides the traditional lossless encoding methodologies that have been investigated through experimental examination, new compression approaches with adaptive dictionaries have emerged lately. These include adaptive LZW encoding methods and common dictionaries with tries for edge computing. However, while these methods utilize adaptive memory allocation or common dictionaries, they do not adopt a predictive technique for evaluating entropy and communication environment awareness. Additionally, their high memory requirements and elevated synchronization overhead render them unsuitable for use in IoT sensor nodes due to resource constraints. The proposed method employs optimization methods for entropy, adaptive segmentation, and environmental awareness while ensuring computational complexity of $O(n)$.

8. Conclusion and Future Work

The proposed method uses multiple techniques that include entropy prediction and adaptive block segmentation together with dual-dictionary learning and environment-aware optimization to create a system that automatically adapts to different data and communication scenarios. The system operates by relying on mechanisms differing from those of standard reactive compression approaches. Three essential components are established, including a system model and an algorithmic design, together with a method for measuring analytical complexity and energy consumption, to construct the foundational framework.

The EP-SLZW method demonstrates better results than traditional LZW and Huffman coding and run-length encoding, as it achieves better compression ratios, higher throughput and shorter end-to-end delays according to multiple simulation studies and actual hardware tests. The results demonstrated that the framework performs best in underwater IoT environments, since these situations require urgent data transmission to manage their limited bandwidth and expensive energy consumption. Ablation analysis demonstrated that all architectural components contribute to a significant performance improvement of the entire system. The dynamic sensing environment requires active dictionary management, for it directly impacts the performance of the system.

The algorithm requires additional computational resources, due to the need to perform entropy estimation tasks and implement predictive control functions. The processing demands of this system remain acceptable for most IoT microcontroller platforms. However, upcoming research will focus on achieving greater cost reductions through the implementation of lightweight machine learning predictors and hardware-assisted dictionary management systems.

The current work suffered from specific constraints, as it has used the infrared communication module to conduct the hardware testing of the system in underwater conditions. Although this approach validates the applicability of this compression method, the nature of communication in the underwater environment cannot be fully simulated. Future considerations will focus on testing the performance of the EP-SLZW method, i.e. on conducting the tests under real underwater circumstances, using acoustic modems. Furthermore, the comparison made in the current project was limited to traditional lightweight lossless compression algorithms, because the aim was for the approach to be adopted in low-power microcontroller systems in the first place.

References

[1] K. Sharma and S.K. Shivandu, "Integrating Artificial Intelligence and Internet of Things (IoT) for Enhanced Crop Monitoring and Management in Precision Agriculture", *Sensors International*, vol. 5, art. no. 100292, 2024 (https://doi.org/10.1016/j.sintl.2024.100292).

[2] Z. Li *et al.*, "Recent Progress on Underwater Wireless Communication Methods and Applications", *Journal of Marine Science and Engineering*, vol. 13, art. no. 1505, 2025 (https://doi.org/10.3390/jmse13081505).

[3] A. Hassan *et al.*, "Arithmetic N-gram: an Efficient Data Compression Technique", *Discover Computing*, vol. 27, art. no. 1, 2024 (https://doi.org/10.1007/s10791-024-09431-y).

[4] B.A. Lungisani, A.M. Zungeru, C. Lebekwe, and A. Yahya, "Autoencoder-based Image Compression for Wireless Sensor Networks", *Scientific African*, vol. 24, art. no. e02159, 2024 (https://doi.org/10.1016/j.sciaf.2024.e02159).

[5] A. Pal *et al.*, "Communication for Underwater Sensor Networks: A Comprehensive Summary", *ACM Transactions on Sensor Networks*, vol. 19, pp. 1–44, 2022 (https://doi.org/10.1145/3546827).

[6] V. Alevizos *et al.*, "A Logarithmic Compression Method for Magnitude-rich Data: The LPPIE Approach", *Technologies*, vol. 13, art.no. 278, 2025 (https://doi.org/10.3390/technologies13070278).

[7] J.C. Gambiroža,T. Mastelić, I.K. Nižetić, and M. Čagalj, "Lost in Data: Recognizing Type of Time Series Sensor Data Using Signal Pattern Classification", *International Journal of Data Science and Analytics*, vol. 20, pp. 397–408, 2025 (https://doi.org/10.1007/s41060-023-00413-9).

[8] V. Kovtun, "Adaptive Information-constrained Mapping for Feature Compression in Edge AI and Federated Systems", *Scientific Reports*, vol. 15, art. no. 30915, 2025 (https://doi.org/10.1038/s41598-025-16604-2).

[9] H. Hu *et al.*, "Full-process Adaptive Encoding and Decoding Framework for Remote Sensing Images Based on Compression Sensing", *Remote Sensing*, vol. 16, art. no. 1529, 2024 (https://doi.org/10.3390/rs16091529).

[10] S.A. Alwahab, "New Technology to Improve an Image Compression Using the LZW Algorithm", *Al-Iraqia Journal for Scientific Engineering Research*,vol. 2, pp. 41–46, 2023 (https://doi.org/10.58564/IJSER.2.3.2023.86).

[11] R. Adam *et al.*, "Lossless Compression with Trie-based Shared Dictionary for Omics Data in Edge-cloud Frameworks", *Journal of Sensor and Actuator Networks*, vol. 14, art. no. 41, 2025 (https://doi.org/10.3390/jsan14020041).

[12] S. Chen *et al.*, "Application of Optical Communication Technology for UAV Swarm", *Electronics*, vol. 14, art. no. 994, 2025 (https://doi.org/10.3390/electronics14050994).

[13] W. Fang *et al.*, "Energy Efficient Unified Computing Framework for Smart Grids with AI-driven Communication, Supercomputing, and Energy Perception Orchestration", *Sustainable Computing: Informatics and Systems*, vol. 49, art. no. 101289, 2026 (https://doi.org/10.1016/j.suscom.2025.101289).

[14] A. Fathalla, K. Li, A. Salah, and M.F. Mohamed, "An LSTM-based Distributed Scheme for Data Transmission Reduction of IoT Systems", *Neurocomputing*, vol. 485, pp. 166–180, 2022 (https://doi.org/10.1016/j.neucom.2021.02.105).

[15] M. Fira, H.N. Costin,and L. Goras, "A Study on Dictionary Selection in Compressive Sensing for ECG Signals Compression and Classification", *Biosensors*, vol. 12, art. no. 146, 2022 (https://doi.org/10.3390/bios12030146).

[16] R. Bender-Salazar, "Design Thinking as an Effective Method for Problem-setting and Needfinding for Entrepreneurial Teams Addressing Wicked Problems", *Journal of Innovation and Entrepreneurship*, vol. 12, art. no. 24, 2023 (https://doi.org/10.1186/s13731-023-00291-2).

Ankur Sisodia, Ph.D., Assistant Professor
Department of CSE (IOT)
https://orcid.org/0000-0002-2253-0382
E-mail: ankur22887@gmail.com
Noida Institute of Engineering and Technology, Greater Noida, India
https://www.niet.co.in

Swati Vishnoi, Assistant Professor
Department of CSE
https://orcid.org/0000-0001-6846-108X
E-mail: swativishnoi1@gmail.com
Noida Institute of Engineering and Technology, Greater Noida, India
https://www.niet.co.in