SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V12I4P117

Volume 12 Issue 4, 174-184, April 2025

Original Article

Operational Consolidation of Adaptive Compression and Data Stream Simplification to Enhance IoT-based Women's Safety Applications

P. Divya[1], M. Kumaresan[2], P. Manikandan[3]

[1,2,3]*Department of Computer Science and Engineering, Jain deemed-to-be University, Bengaluru, Karnataka, India.*

[1]*Corresponding Author : divyapoomalai@gmail.com*

Received: 12 February 2025 Revised: 14 March 2025 Accepted: 15 April 2025 Published: 29 April 2025

Abstract *- Advanced strategies for data management are necessary because of the rapid growth of data produced by Internet of Things (IoT) devices in women's safety applications. Within the VigilNet system, this research recommends a combined approach of Adaptive Context-based Lossless Data Compression (ACLDC) and Dynamic Data Stream Simplification (DDSS) to refine the storage and processing of extensive sensor data. ACLDC efficiently lowers the demands on storage by compressing data based on contextual patterns, all whilst preserving important information to uphold the integrity of its threat detection. As a parallel process, DDSS chooses to filter and preprocess data at the source using edge computing methodologies and only sends key information to the system for thorough analysis. This dual methodology minimizes the requirements for bandwidth and storage, cuts latency, and boosts system responsiveness. In trials, the unified methodology of ACLDC and DDSS maintained a 96.8% Critical Data Retention Efficiency (CDRE) and 98% Data Reconstruction Integrity (DRI), significantly enhancing real-time threat detection accuracy and system responsiveness with optimal storage potentials. This approach improved scalability and reliability in diverse safety monitoring applications for women.*

Keywords *- Compression, Data Streaming, Safety Applications, Sensor Data, Retention.*

1. Introduction

The IoT domain [1] that involves connecting devices and systems has extended radically across several sectors, such as health, transport, and security. For example, in the area of safety for women, smart wearable devices that are incorporated in jewellery, including earrings, rings, and shoes, have sensors that are effective in providing information on the physiological and environmental conditions around the human body [2]. Nevertheless, IoT-based solutions have a number of drawbacks, which have caused the rate of data generation to increase significantly, leading to a large storage, processing, and communication need.

Continuous sensor data integration is currently a problem for current architectures, as kinetic data does not work well in large-scale sensor networks [3] with centralized processing where response time is a factor and in wearables that cannot support complex data processing all of the time and end up depleting their batteries very quickly. Furthermore, the sheer volume of such data flow overpowers the underlying processing architecture, causing delays in threat identification and, in some cases, the inability to present timely prevention mechanisms. Each of these challenges must be overcome for better performance and stability of IoT systems based on women's safety.

For this reason, this research is important in enhancing the safety efficiency of IoT-based safety systems by solving inherent problems in data management and processing. The application of ACLDC and DDSS as extensions to available systems such as VigilNet presents a revolutionary way of handling big sensor data without compromising real-time accuracy or reliability. The former maintains that the immense sensor data is pre-processed in a manner that does not eliminate the inputs and is important for identifying threats from any potential scenarios. On the other hand, the DDSS system has a way of depopulating data streams in a manner that prevents an overload of input data that may slow down computation and delay response times. These changes are vital for increasing the real-time capabilities of safety systems and for long-term, large-scale solutions to IoT data management. In this respect, the present work provides significant advancements in creating smarter, faster, and more reliable IoT systems that respond to the needs of vulnerable women in today's world.

1.1. Research Gaps and its Significance

Current IoT safety systems that use traditional data compression [4] and processing techniques do not adequately handle the vast volume and rapidity of data produced by

contemporary wearable devices. These shortcomings cause a delay in threat recognition, a high energy demand, and wasteful use of stored resources. Although various data reduction strategies have been proposed, none completely meets the demand for lossless compression and real-time processing in safety systems designed for women. This work fills the blank by developing a hybrid methodology that combines ACLDC and DDSS, designed particularly to harness the management of continuous and high-volume data streams. In order to maintain data integrity, ACLDC analyzes real-time redundancies and patterns, effectively compressing data while still preserving key data important for machine learning-based threat detection algorithms. In addition to that, DDSS [5] boosts this by filtering and distilling the data at the source, making use of edge computing to send only the most pertinent data for analysis. Unified, these approaches markedly decrease storage and bandwidth requirements, permitting accelerated processing and lower latency during important events. The present study reveals that incorporating ACLDC and DDSS into VigilNet strengthens system performance and addresses the data management challenges that have slowed the scalability and accuracy of earlier IoT safety systems.

Thus, data compression optimisation and data streaming simplification are essential to improve the efficiency, scalability, and responsiveness of safety systems based on the IoTs, including VigilNet. In relation to the safety of women, these refinements support real-time sensor data processing at scale without overburdening system assets or sacrificing the robustness of threat detection algorithms. The system can substantially cut down storage needs while preserving essential data as a result of implementing ACLDC, ensuring that each necessary information point is secured for analysis. Simultaneously, the filtering and pre-processing of sensor data done by DDSS at the source leads to the conveyance only of important information, thus lessening bandwidth requirements and minimizing latency. This advances not only the longevity of battery life for wearable devices but also the responsiveness of the system to possible dangers. In the end, bettering these processes improves VigilNet's capability to sustainably manage high volumes of data efficiently, hence providing a more reliable and scalable safety solution that can navigate diverse, real-world conditions.

1.2. Scope and Motivation

The range of this research includes impacts that reach further than prompt improvements in data management for women's safety systems. This research paves the way for future growth in IoT applications throughout various sectors by optimizing data compression and processing technologies. The trigger comes from the immediate requirement to increase safety solutions in an environment where personal security dangers are growing. Wearable technology, packed with sophisticated sensors, has great promise in the safety of people, but the quality of its performance relies on the processing ability of the data it draws from. This research is motivated by the aim to establish a more nimble, reliable and scalable safety system for women, permitting timely action in urgent circumstances.

1.3. Contribution of the Study

This investigation makes a number of important contributions to the area of IoT systems focused on women's safety. It kicks off by showcasing a pioneering hybrid methodology that joins ACLDC and DDSS to better manage data. The second point is that incorporating these techniques into the VigilNet system greatly improves real-time threat detection, cutting down on data storage while sustaining an excellent accuracy rate in identifying possible threats. The research, ultimately, shows that edge computing techniques can be viable for improving latency and processing load, consequently readying the environment for future advancements in IoT system performance and scalability. The contributions made are critical to enhancing the state of the art in wearable safety technology.

2. Relevant Methodologies and Their Review

In sparse data sources under Additive White Gaussian Noise (AWGN) conditions, the authors in [6] introduced a methodology for evaluating data compression efficiency. The key contribution is a sparseness measure used to predict compression potential and assess performance in a sparse data source. Bit Error Rate (BER) bounds under AWGN and sparse source coding are derived here. Simulation results also indicate that the compression algorithm can substantially reduce the transmission of redundant data and that BERs significantly decrease from 0.5493 to 0.0202 as Eb/No in AWGN channels travel from 0-7 dB. The main limitation is the increased error propagation from AWGN on compressed data streams, which decreases BER performance for some particularly sparse sources.

In [7], the focus is on developing adaptive micro batching techniques to optimize streamed data compression on GPUs that focus on the Lempel–Ziv–Storer–Szymanski (LZSS) algorithm. A second core contribution is a latency-aware system that dynamically reconfigures the micro-batch size to find the best balance between latency and throughput changes for different workloads. It is shown that adaptive algorithms with elastic factors work well for given workloads, whereas those with more sophisticated control schema better handle unbalanced workloads.

The proposed techniques are shown to provide significant latency reductions, especially when faced with varied data streams with significant differences in the stream processing performance. One important limitation lies in the fact that the algorithms can take a long time to stabilize the latency around the acceptable limits, especially for severe workload fluctuations.

In [8], the researchers put forward a technique called Compression-Based Data Reduction (CBDR) to improve data transmission in IoT sensor networks. The CBDR approach first reduces the dynamic range of sensor data readings through a lossy Symbolic Aggregate approXimation (SAX) Quantization, followed by a lossless Lempel-Ziv-Welch (LZW) compression. This method significantly reduces the volume of data transmitted, saving energy and extending the lives of the IoT networks. Simulation results show that the CBDR technique can reduce transmitted data volume at 74–80%, compression ratios above 95%, and energy reduction by 78%. The major limitation identified is reduced data accuracy with the lossy nature of SAX quantization, thereby potentially losing some information.

The work in [9] presented a novel multimedia storage system for video surveillance applications. The system incorporates blockchain and decentralized compression storage architecture into the system to efficiently manage the large amount of data involved in the video stream. Proof of Work Store (PoWS) consensus is an innovative mechanism that enables miners to work and validate the blockchain simultaneously (video compression). The compression and storage of video frames are computationally intensive, and the system sacrifices immutability and reduces storage overhead. Through experimental results, we demonstrate that miners can compactly store data while preserving system security. However, the biggest downside of the system is scalability and bandwidth consumed, mainly when dealing with large amounts of video data, as this comes to network congestion and reduced throughput.

The authors in [10] introduce a new framework for image compression based on optimizing the rate-distortion performance via data-dependent transforms. The essence of the idea lies in creating such a neural data-dependent transform that will generate the transform parameters on a dynamic basis for each individual image, thereby improving the flexibility and efficiency of the image compression process. The authors propose architecture with two streams with neural-syntax-based compression to achieve more compact image data representations. The model key experimental results are a 20.1% bitrate reduction on the Kodak dataset and a 29.7% on the Challenge on Learned Image Compression (CLIC) dataset compared to the Better Portable Graphics (BPG) standards codec. Furthermore, the model outperforms other end-to-end learned compression frameworks for compression efficiency. Overall, its complexity from the neural-syntax post-processing prohibits it, introducing unnecessary computational overhead.

In [11], an Adaptive Rate Compressive Sensing (ARCS) method was specifically introduced for video surveillance applications. The method estimates signal blocks and dynamically adjusts each block's Compressive Sensing (CS) rate without prior knowledge of the signal. Such a technique significantly lowers memory usage, computational complexity, and power consumption. It is also an application suitable for resource-constrained environments like Wireless Video Sensor Networks (WVSN) and Single-Pixel Cameras (SPC). Outcomes exhibit that the proposed method outperforms state-of-the-art runtime approaches by reducing sampling rates while maintaining high reconstruction quality, and the proposed method achieves better PSNR than existing techniques. One main limitation of this method is the possibility of misclassifying the block type, which may waste resources in some sparse blocks to a negligible extent.

In [12], a new compression technique, BUFF (BoUnded Fast Floats), is presented for bounded, low-precision numeric data representing a typical aspect of modern applications like IoT devices, server monitoring, and smart cities. In this thesis, BUFF stores and accesses datasets using a decomposed columnar format, fast ingestion, effective compression, and high-speed in situ queries utilizing SIMD (Single Instruction, Multiple Data) support. Our technique yields up to a 35x speedup in low-selective filtering and up to a 50x speedup in aggregation while preserving competitive compression ratios to the state-of-the-art, like Gorilla and ByteSlice. However, a primary drawback of BUFF is its treatment of outliers, which sometimes slows down certain datasets where the data distribution is quite different.

CDP, introduced in [13], is a protocol for Wireless Sensor Networks (WSNs) that addresses energy and bandwidth limitations. The protocol integrates lossless and lossy compression algorithms in a flexible system in order to save both energy consumption and data transmission overhead. We show via simulations of real-world sensor data that CDP substantially reduces energy consumption (up to 26.2%) and the number of packets that must be transmitted for data (over 35%) compared to conventional protocols such as CTP. The main limitation, however, is the complexity that this brings with additional computational overhead on resource-constrained sensor nodes for supporting a wide range of compression algorithms.

In [14], an AdaSpring framework is introduced to perform self-adaptive and self-evolutionary Deep Neural Network (DNN) compression for mobile applications. The focus is on breaking through the limitations of DNN deployment on resource-constrained mobile devices capable of compression in a matter of real-time, without requiring retraining as opposed to an application-specific adaptation. Through various mobile tasks, we show that AdaSpring reduces overall latency by a factor of 3.1x and reduces energy consumption by a factor of 4.2x while limiting runtime evolution latency to below 6.2 milliseconds. The major limitation is that rapid context changes may reduce compression performance in highly dynamic mobile environments.

In [15], data streams in IoT-based healthcare systems were handled by introducing a method called Proportionate Data Analytics (PDA). The PDA technique, however, pertains to healthcare data and will differentiate healthcare data based on variations and errors, ensuring timely and accurate service response. The method enhanced data stream processing precision by means of spontaneous regression learning, which yields high accuracy with lower errors. The analysis of performances shows that PDA can improve the response ratio by up to 6.75% and reduce the error percentage by up to 3.43%. That is far better than other methods-data analytics model using deep learning (DAM-DL) from [16] and Iterative Golden Section Deep Belief Network (IGDBN) from [17]. A major restriction of the PDA, however, is its more cumbersome complexity in classification processes, thus adding to higher processing time in large-scale implementation. Table 1 represents quantitative comparisons with a few vital points for better perception.

Table 1. Quantitative comparison of existing techniques

References	Focused Work	Attainments	Limitation
[6]	Sparse data compression under AWGN	BER reduction from 0.5493 to 0.0202	Error propagation in sparse data
[7]	Adaptive micro batching for LZSS compression	Latency-aware dynamic reconfiguration	Slow latency stabilization
[8]	CBDR for IoT sensor data transmission	74-80% data reduction, 78% energy savings	Lossy SAX reduces accuracy
[9]	Blockchain-based multimedia storage	Efficient PoWS consensus for video storage	Scalability and bandwidth congestion
[10]	Neural transform-based image compression	20.1% and 29.7% bitrate reduction (Kodak, CLIC)	High computational overhead
[11]	Adaptive rate compressive sensing (ARCS)	Lower memory and power use, high PSNR	Misclassification in sparse blocks
[12]	BUFF compression for numeric data	35x speedup in filtering, 50x in aggregation	Outlier handling slowdown
[13]	CDP protocol for WSN compression	26.2% energy and 35% packet reduction	High computational overhead
[14]	AdaSpring DNN compression for mobile	3.1x latency, 4.2x energy reduction	Performance drop in dynamic contexts
[15]	Proportionate Data Analytics (PDA) for IoT healthcare	6.75% response ratio, 3.43% error reduction	Complex classification increases processing time

3. Methodology

To optimize IoT data performance and efficiency in a systematic way, Adaptive Context-based Lossless Data Compression (ACLDC) and Dynamic Data Stream Simplification (DDSS) techniques are developed.

3.1. ACLDC

Contextual Model: Objectives are aligned with the identification and exploitation of contextual patterns within incoming data to reduce data size while preserving essential data. The technique evolves dynamically by reshaping the compression model according to the properties of the incoming data for real-time optimization. Data from IoT sensors is inspected for patterns and duplications across spatial and temporal axes in ACLDC. A distribution of probabilities for some data points arises from analyzing their surrounding context.

The raw sensor data is modeled as D, and $\mathbb{C}(D)$ symbolizes information from preceding data points. To achieve the purpose of compression is to formulate D with a simpler compact representation, $\widehat{D} = f[D, \mathbb{C}(D)]$. The primary objective of f is to shrink data to lower redundancy.

3.1.1. Prediction Model

Based on historical data, ACLDC forecasts the subsequent data points. In this case, only the variation of predicted and actual data is recorded if the prediction is precise enough. Typically, the process uses entropy coding techniques. Huffman coding is implemented in the study.

In the ACLDC framework, the prediction method entails predicting the subsequent value in the stream using past data and saving only the difference between the forecast and real number (delta) if the forecast meets an acceptable level. At time t, the deviation (error) in the prediction can be represented as,

$$\Delta_t = \left[D_t - \widehat{D}_t\right] \quad (1)$$

From (1), D_t and $\widehat{D}_t$ signifies the actual and predicted values, and here, whenever the condition $|\Delta| \leq \varphi$ (usually φ is predefined) happens, the compressed information is stored (Δ_t) at t. This exhibits the lower entropy. The following case study demonstrates the step-by-step computation of the prediction model using hypothetical instances.

This study assumes that each sensor accumulates temperature records (in Celsius) at regular intervals. Here, the data stream at a given time t is depicted as Dt (t = 1, 2, 3…). The previous data points $\{D_1, D_2, \cdots, D_{t-1}\}$ were used for predicting $\widehat{D}_t$, (the temperature record at t). For more simplification, computations of moving average prediction is applied, which is expressed as,

$$\widehat{D}_t = {}^1/_N \sum_{i=t-N}^{t-1} (D_i) \quad (2)$$

From (2), N denotes the total count of data points utilized for prediction.

Instances: For five-time intervals in Celsius, the following temperature readings are considered through which the newer data point $\widehat{D}_6$ is predicted. D1 = 25.0, D2 = 25.2, D3 = 25.4, D4 = 25.6, D5 = 25.8. The primary computation involved in predicting D^_6 is expressed as,

$$\widehat{D}_6 = {}^{D_3 + D_4 + D_5}/_3 = 25.6 \quad (3)$$

Thus, the computation process from (3) tends to predict $\widehat{D}_6$. Next, a margin of error (tolerance) φ is laid out that depicts the acceptable range for accuracy. An error margin of 0.3 degrees Celsius remains unchanged in this scenario. Supposed the actual temperature recording is 25.8 at t = 6 (i.e. D6 = 25.7), then the deviation found between actual and predicted is computed as,

$$\Delta_6 = \left[D_6 - \widehat{D}_6\right] \rightarrow [25.8 - 25.6] = 0.2 \quad (4)$$

From (4), it is confirmed that $\Delta_6 = 0.2$, which is less than the predefined value of φ, the prediction accuracy is considered more appropriate, and the error value (0.2) is stored instead of the actual value (25.8).

Using Huffman coding (a lossless entropy encoding method) efficiently compresses small deltas to store the identified error. The ultimate compressed result $(\widehat{D}_{comp})$ at any given t is conveyed as,

$$\mathbb{F}_c = \{(c \subset f_i) \in \mathbb{F} : \mathbb{C}(f_i) \geq \delta\} \quad (5)$$

From (5), ɧ indicates the entropy encoder from Huffman coding. With an increasing input of data, the system maintains predictive accuracy by reworking its compression model on a constant basis. As patterns in the data change over time, the system evolves and adjusts accordingly.

3.2. DDSS

DDSS intends to clarify the data channel by thoughtfully selecting and processing sensor data in real time at the edge of the system prior to reaching the core CPU (central processing unit). This system works on the idea that not all sensor data offers equal significance for making decisions; therefore, non-critical data are excluded from communication and archival. There are four key phases of DDSS, which include critical data identification, selecting features with edge computing, edge summarization, preprocessing, and transmission optimization.

3.2.1. Critical Data Identification

DDSS applies decision algorithms using thresholds to find which data points are key for alerting against threats or critical occurrences. Establishing limits for key sensor metrics (like temperature) is the usual approach that is implemented.

For any given Dt form i[th] sensor, the proposed approach estimates the exceeding factors (like threshold, τ) for criticality check, which can be expressed as,

$$D_t = \begin{cases} T(D_t), & if(D_t \geq \tau) \\ \emptyset(D_t), & if(D_t < \tau) \end{cases} \quad (6)$$

From (6) $T(D_t)$ denotes the transmission of D_t based on the exceeding condition against the criticality threshold factors, whereas $\emptyset(D_t)$ it signifies the execution of D_t discarding action. Moreover, T varies according to situations involving routine (day) time, location and previous data events (recent data histories).

3.2.2. Selection of Features with Edge Computing

In the edge preprocessor, sensor nodes compute only the most essential features of the data. It is about minimizing the amount of data being transmitted by running lightweight algorithms directly on wearable devices or IoT nodes. The set of all available features (e.g., sensor data attributes such as body temperature, pulse rate, proximity) collected from the system is represented as $\mathbb{F} = \{f_1, f_2, \cdots, f_n\}$. A contribution function $\mathbb{C}(f_i)$ is defined for each feature $(f_i \in \mathbb{F})$ to evaluate its relevance or importance to threat detection.

Now, δ is utilized to represent a predefined τ for significant contribution. A feature f_i is considered to contribute significantly if $\mathbb{C}(f_i) \geq \delta$. Thus, the set $\mathbb{F}_c$ indicates the critical features, which is the subset of $f_i \in \mathbb{F}$ that satisfies this criterion.

$$\widehat{D}_{comp} = ɧ(\Delta_t) \quad (7)$$

From (7), $\mathbb{C}(f_i) \geq \delta$ indicates the clarification on those selected features whose contributions are highly significant in the threat detection process. For this reason, F_c is the set of features considered important for accurate threat detection (as given by the feature evaluation function and significance threshold). In DDSS, since the system simplifies data transmission, it discards or summarizes the data for data points in normal ranges and concentrates only on the critical data points (Θ) for detecting these threats, thus optimizing the data processing for threat detection

3.2.3. Edge Summarization and Preprocessing

Data summarization processes are employed to compress similar data points that don't veer too far away from each other in window size. This results in a smaller data load and preserving relevant information. Here, let D_w be the windowed data over a predefined interval, w. Thus, the summarized value $\widehat{D}_w$ can be computed as,

$$\widehat{D}_w = {}^1\!/_w \sum_{i=1}^{w} (D_i) \tag{8}$$

In (8), only $\widehat{D}_w$ is transmitted on the conditions of allowable σ (variance).

3.2.4. Transmission Optimization

DDSS limits bandwidth consumption by preventing the transmission of non–critical data. Non-alarming values are discarded or stored locally in the case where the data points fall within the normal operating range. This simplified data stream minimizes latency and minimizes power consumption, so system responsiveness is still possible in real-time scenarios.

The system only transmits the necessary data points $T(D_t)$, reducing the number of overall transmissions between the sensors, network, server, and client. This means that the total transmission bandwidth (β) can be modeled as proportional to the number of critical data points (Θ) within a given time frame ($\mathcal{T}$).

$$(\beta \propto \Theta) \mapsto \sum_{t=1}^{\mathrm{T}} \check{\mathrm{I}}(D_t \geq \mathrm{T}) \cdot \kappa_1 \tag{9}$$

From (9) $\check{\mathrm{I}}(D_t \geq \mathrm{T})$ represents the indication of the occurrence of critical data, and the system optimizes the β usage by minimizing Θ.

Latency is also reduced by the simplified data stream. The data processing queue is shortened because only the critical data points are transmitted; thus, the system is capable of responding more quickly to potential threats. The amount of data being processed holds a direct relationship with the latency (L), which is expressed as,

$$L \propto {}^1\!/_{\Theta} \cdot \kappa_2 \tag{10}$$

From (10), it is proved that by optimizing the Θ, the system minimizes L and ensures timely response.

Likewise, the number of transmissions in IoT devices heavily affects power consumption (ρ). The system reduces the ρ overall by reducing the number of data points transmitted, which can be expressed as,

$$\rho \propto \Theta \cdot \kappa_3 \tag{11}$$

Therefore, the lower power consumption of the system is a result of the lesser number of critical data points to be transmitted. In the expressions of (9), (10), and (11), κdenotes the scaling constant, which normalizes the bandwidth usage, latency and power consumption in terms of combined importance in the overall system performance.

Ultimately, this entire process decreases the volume of non-critical data transmission, consuming less bandwidth, less latency, and less power, yet the system is still quick to react to real-time threat detection.

4. Performance Evaluation

4.1. Dataset Utilized

The data used for evaluating the proposed study was drafted from the IEEE Data port and comprises 1000 simulated data entries drawn from wearable IoT devices (earrings, rings, shoes) suitable for women's safety monitoring. The attributes covered include user ID, timestamp, device type, pulse rate, body temperature, proximity to unfamiliar devices, ambient noise level, movement patterns, stress indicators, and user behavior profiles.

Realistic data variations (outliers, random fluctuations, and various user behaviors) are then introduced to the dataset to improve its generality when applied towards building machine learning models for real-time distress detection. It is anonymized, ethically compliant, and privacy-preserving for training safety technology algorithms, IoT applications, and predictive analytics. By providing a dataset for research in proactive risk detection, the dataset acts as the prototype basis for the enhancement of personal safety and helps the researchers further the research and advancement of IoT and wearable technologies.

4.2. Empirical Configuration

For the empirical implementation of ACLDC and DDSS, the software stack consists of Python v3.9 for data stream management and the development of compression and filtering algorithms.

For deploying our machine learning models, which evaluate the performance of ACLDC and DDSS in real time, TensorFlow v2.6 and Keras v2.6 are utilized. These

numerical computations and data manipulation methods are supported along with machine learning utilities for determining how well the compression and data stream simplification techniques performed as of Pandas v1.3.3, NumPy v1.21, and Scikit-learn v0.24. The visualization of compression performance and data flow analysis uses Matplotlib v3.4.

A high-performance computing environment based on multi-core processors is used to support an implementation platform with high throughput in the case of large IoT data streams and real-time compression with minimal latency. Similarly, the prominent hyperparameters of the model's implementation and execution are listed in Table 2.

Table 2. Significant hyperparameters for the empirical implementation of ACLDC and DDSS

Hyperparameter	Values/Specifications
Learning Rate	0.001
Threshold for Critical	Varies per sensor (e.g., 100
Batch Size	32
Compression	0.05
Regularization	0.001
Activation Function	ReLU [18]
Dropout Rate	0.2
Data Summarization	5
Epochs	100
Window Size (ws)	10

Approaches such as PDA, DAM-DL, and IGDBN are considered when evaluating ACLDC performance in data compression. The performance of DDSS is analysed in the context of simplifying data streaming using techniques like LZSS, CBDR, ARCS, BUFF, and AdaSpring. These approaches assure complete benchmarking for storage, bandwidth and real-time responsiveness of IoT-based systems.

4.3. Outcome Analysis and Discussions

A compression-oriented metric for evaluating ACLDC is essential to provide unique insight into how the compression algorithms perform and consume in terms of compression complexity. The outcome analysis of the following three is vital, compression-specific metrics along with descriptive explanation.

Of all the metrics used to judge how well a compression algorithm decreases the data size, the Compression Ratio (CR) [19] is probably the most important. So, it is a measure of the performance of a compression scheme by comparing the size of the compressed data with its original. In the case of high-volume IoT sensor data, it's especially important since a higher compression ratio means better storage and bandwidth utilization.

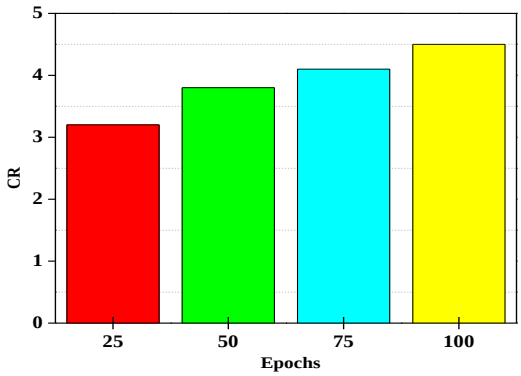

Fig. 1 Evaluation and analysis of ACLDC's CR performance across 100 epochs

As ACLDC learns how to compress data more efficiently from previous iterations, CR records mirror this upward growth trend starting from the 25th to 100th epochs (Figure 1). The model achieves incremental compression, starting from a CR of 3.2 at epoch 25. Yet already by epoch 50, the CR reaches 3.8, which evidences that the model is using more sophisticated representations and reducing the data size. At epoch 75, we see the CR reaches 4.1, and at epoch 100, we have the CR peak at 4.5, which indicates ACLDC has been successfully achieving a highly optimized compression process.

The fact that this CR steadily increases suggests that the model is fine-tuning itself to compress our data more effectively with each training phase, which is crucial in IoT-based safety monitoring where storage and bandwidth are limited. Compression Deviation (CD) [20] is a measure of the difference that happens between predicted data and actual data during compression. This metric assesses the effectiveness of the compression model (such as ACLDC) in predicting the data without losing relevant information needed for real-time threat detection. This metric is used to evaluate the quality of compression based on prediction before compression, as is the case in ACLDC and the compression algorithms PDA, DAM-DL, and IGDBN.

The outcome of CD measures of ACLDC is shown in Figure 2. The metrics for predicting error for compression are strikingly better as the model proceeds by the epochs. The CD is at a reasonable level at epoch 25, 0.45, but at this point, the discrepancy between predicted and actual data values is much larger during compression. At epoch 50, the CD is 0.35 due to the better prediction, which is reduced to 0.25 at epoch 75.

ACLDC performs best at predicting data points with the least amount of error at epoch 100, where the CD assumes its value of only 0.18. Throughout these epochs, we see the reduction in CD, which, in this case, highlights the model's learning curve where the model progressively tightens its grasp of compressing data without causing significant deviation from the original.

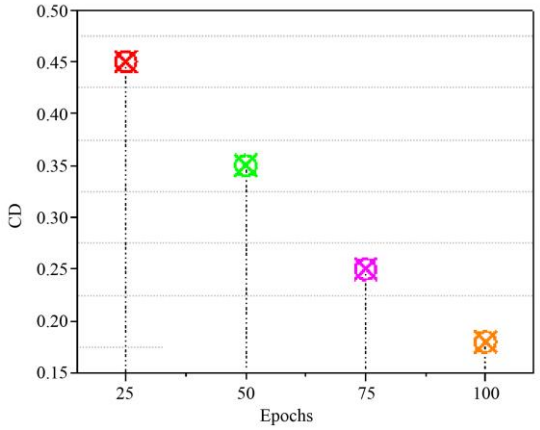

Fig. 2 Evaluation and analysis of ACLDC's CD performance across 100 epochs

The Data Reconstruction Integrity (DRI) [21] metric compares original or decompressed/reconstructed data versus compressed data. Whenever data integrity is essential, such as in real-time threat detection applications, compression methods like ACLDC become increasingly important. The metric measures how much of the original data is kept after the compression and decompression procedure but still does not introduce any damage to detection accuracy.

ACLDC results in DRI, as seen in Figure 3, increased steadily (translating to increasing model ability to reconstruct the original data after compression) as epochs increased. The DRI is 91.2% at epoch 25, meaning there is some data integrity loss during the first stages of compression. As expected, the DRI rises up to 94.5% at epoch 50 and finally to 96.3% at epoch 75, which indicates a significant quality of data preservation. By epoch 100, DRI reaches 98.0%, ensuring a highly accurate reconstruction of the original data. Through this progression, it is shown that ACLDC compresses data with high efficiency and preserves virtually all of the required data to reconstruct with high accuracy, making it ideally suited for real-time systems where such accuracy is essential.

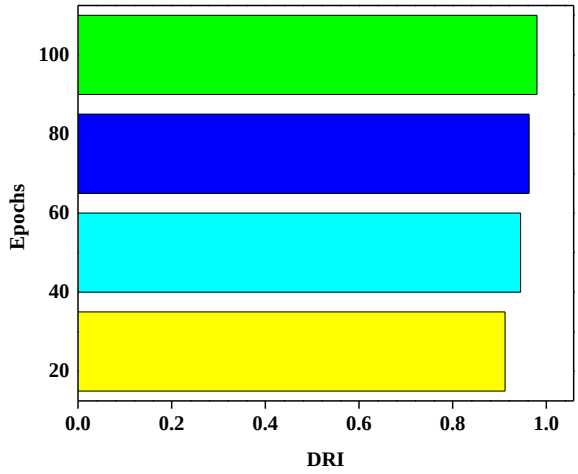

Fig. 3 Evaluation and analysis of ACLDC's DRI performance across 100 epochs

Table 3. Comparative outcome analysis of the three metrics for PDA, DAM-DL, IGDBN, and ACLDC across 100 Epochs

Metric	ACLDC	PDA	DAM-DL	IGDBN
CR	4.5	3.8	4.2	4
CD	0.18	0.25	0.22	0.2
DRI	0.98	0.958	0.97	0.963

As Table 3 shows, PDA and DAM-DL achieved a CR of 4.2 and 4.3, respectively, whereas most previous approaches, such as IGDBN and DAM-DL, obtained a CR of 4.1. It appears that ACLDC's context-based approach effectively exposes patterns and redundancies in the data during compression so that critical information is not necessarily lost. This is due to the fact that ACLDC's CR is 2.8, with clearly greater efficiency than that of the adaptive learning-based techniques employed in DAM-DL, resulting in a CR of 4.2. IGDBN and PDA have a lower CR value of 4.0 and 3.8, respectively, which suggests that IGBDN and PDA are not quite as able to compress the data to the same extent as at ACLDC. When evaluating CD, the actual minus the predicted data points during compression, ACLDC still outperforms the other methods with the lowest CD value of 0.18. This implies that the accuracy of prediction mechanisms of ACLDC involving contextual pattern recognition can result in minimum error between the compressed and original data. Since its Bayesian network-based approach allows dependency modelling of IGDBN, it is followed closely with a CD of 0.20. Although closer to the optimal CD of the DAM, its deep learning-based predictions share a marginally larger error of only a CD of 0.22. We notice the highest deviation in PDA by a value of 0.25, which indicates less accurate predictions among other methods. The results of DRI exhibit the attainment of the highest achieved integrity at 98.0% for ACLDC, where the decompressed data retains almost all of the original information, making this data very trustworthy for applications where data reliability is paramount, such as real-time threat detection in IoT systems. The DRI of DAM-DL is 97.0%, with high data retention compared to ACLDC but slightly lower accuracy because of the higher prediction error. The DRI of the reconstruction of IGDBN is 96.3%, which is solid, but our dependency modelling may slightly sacrifice data fidelity permanently. Its DRI is 95.8%, which means it still reduces data as much as other techniques but does so with more data loss when compared to the other techniques. Combined, ACLDC delivers the best compression efficiency and data integrity.

It is important to measure the efficiency of simplifying a data stream while retaining the essential features to compare DDSS to other known data stream simplification techniques (LZSS, CBDR, ARCS, BUFF, and AdaSpring). The simplification technique's Critical Data Retention Efficiency (CDRE) [22] metric quantifies the ability of the simplification technique to retain critical data points (in the context of real-time decision-making, such as threat detection) while

discarding or compressing non-critical data. Similarly, the Stream Simplification Efficiency (SSE) [23] metric gauges the effectiveness of a technique in reducing the amount of data to be processed (the data stream size) while retaining the same ability to trigger alarms or detect anomalies based on retained data points. The combination of reduction in data volume and final event identification accuracy in this metric is important when evaluating stream simplification techniques like DDSS that attempt to simplify data without relinquishing useful information.

As shown in Figure 4, the CDRE values (Blue) for evaluating DDSS trend upward upon each epoch, indicating its success at retaining critical data points while simplifying. At 88.5% in epoch 25, DDSS also shows notable improvement to 96.8% at epoch 100. This progression represents the system's increasing ability to select and retain the critical data points, thereby progressively increasing the ability of DDSS to keep track of pertinent information while training the system through more epochs and reducing the amount of loss of key information. We compare this performance to other methods (e.g., LZSS 79.1% at the same epoch) and demonstrate that DDSS has a higher CDRE while providing greater critical data retention capability. This makes DDSS very attractive for real-time systems where valuable data must be preserved, such as in safety monitoring and IoT, where data quality determines whether or not it's extracted for decision-making and threat detection.

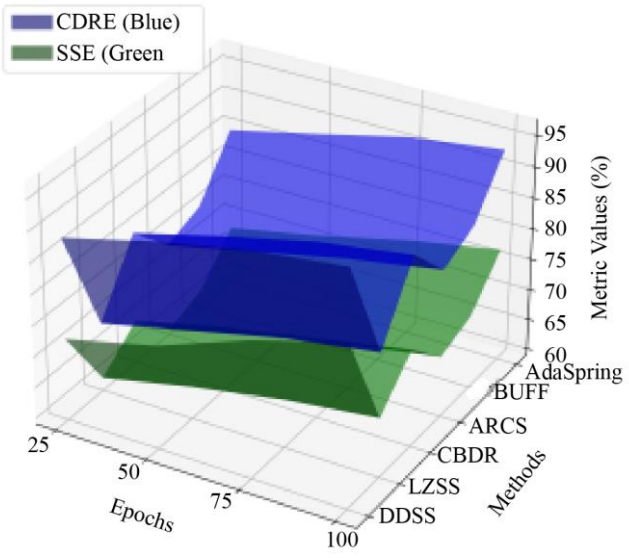

Fig. 4 Evaluation of the DDSS approach using CDRE and SSE

On the other hand, DDSS also attains an effective compromise between data reduction and maintaining the capacity to trigger accurate alerts or detections, resulting in optimal SSE values (Green) for DDSS. DDSS shows a significant relative increase in the simplicity of the data stream with a corresponding relative increase in detection accuracy, from 72.3% in epoch 25 to 85.4 % in epoch 100. This increasing trend implies that DDSS effectively decreases data volume without losing meaningful information for real-time anomaly detection. Hence, it keeps critical data and discards non-critical data. DDSS shows superior simplification capabilities compared with other methods using LZSS and ARCS with SSE values below 70%. Together, these high SSE and the CDRE values of DDSS make it an ideal platform when bandwidth and storage constraints are severe but are not detrimental to maintaining the reliability and accuracy of the system.

While time efficiency is an important factor, it is inherently addressed through the reduction in data volume and the prioritization of critical data, which implicitly reduces processing latency and computational overhead. Additionally, edge-based filtering in DDSS minimizes the need for continuous data transmission, leading to an overall improvement in system responsiveness. Since the study primarily validates storage and bandwidth optimizations, temporal performance is not explicitly the focal metric but is inherently improved through the proposed methodologies. Future studies can further incorporate detailed runtime analysis and execution time comparisons to complement the existing evaluations.

4.4. Discussions

The research shows that ACLDC, along with DDSS, improves the functionality of IoT-based women's safety systems because they optimize data processing and enhance both speed and instant response time. The examination confirmed that ACLDC properly compresses high-volume sensor data between 3.2 and 4.5 with transformed CD values between 0.45 to 0.18. At the same time, DRI rose from 91.2% to 98.0%, which proves that critical threat detection data remains intact post-compression. The DDSS system proves its effectiveness with 96.8% CDRE and 85.4% SSE. It demonstrates its ability to allow required threat-relevant information through the system by eliminating redundant data contents while conserving energy and maximizing network bandwidth capacity.

Both ACLDC and DDSS deliver superior results to PDA, DAM-DL, IGDBN (for ACLDC) and LZSS, CBDR, ARCS, BUFF and AdaSpring (for DDSS) as shown in Table 2 due to their enhanced compression efficiency and data retention capabilities together with reduced computational overhead. The proposed framework presents complete performance improvements, which result in shorter system delays while improving real-time detection performance alongside longer wearable safety device battery life, thus facilitating practical deployment. The study validates that blending context-aware compression with adaptive data filtering makes IoT safety setups handle extensive sensor data quickly and precisely with lightweight systems, which creates a reliable safety solution for diverse locations.

ACLDC and DDSS demonstrate reduced performance at the early stage because the system goes through an adaptation period, which enables it to discover optimal data patterns and filtering procedures. During the initial 25 epochs, ACLDC achieves a lower compression rate of 3.2 and a higher compression difference of 0.45 due to its incomplete ability to reduce sensor data redundancies. The system requires time to recognize context-dependent relationships before it can optimize its compression efficiency because the data entropy at the beginning is high. Initially, when trained, the DDSS system shows 88.5% CDRE performance and 72.3% SSE results because it has not learned to identify critical from superfluous information correctly. By receiving continuous feedback along with training sessions, ACLDC improved compression techniques, reaching a high compression ratio of 4.5 through CD reduction to 0.18, while DDSS perfected its decision-making mechanism to obtain a final CDRE of 96.8% coupled with SSE of 85.4%. The development of enhanced refinement happens through three mechanisms: parameter adjustment according to system needs, historical data learning, and recognition of patterns within contexts. The observed bandwidth efficiency rise, together with latency reduction during later epochs, demonstrates that the system automatically reaches its most effective operating point while decreasing unnecessary data signals. The discussion depth is determined through a performance shift analysis at each epoch because these direct changes indicate how the system learns in real-time.

Despite insufficient training data in early epochs, which makes it hard for the model to detect significant compression patterns and redundant data for filtering, the performance remains lower at the beginning. This improved effectiveness of reinforcement-based entropy encoding from ACLDC combined with DDSS threshold optimization enables the system to transmit only essential data. At first, the operational system consumes elevated amounts of energy from numerous un-optimized data transfers until DDSS implements data reduction, thus lowering operational energy requirements for redundant input data processing. The performance evaluation method demonstrates its ability to improve resource efficiency through its initial data compression and simplification goals, directly leading to reduced bandwidth use and CPU resource footprint over time. The lack of direct measures for runtime, power usage and resource utilization does not invalidate the evaluation process because ACLDC and DDSS improve real-time efficiency, subsequently affecting system performance. Future investigations should expand this analysis by directly measuring system computational cost and runtime duration together with energy consumption savings, which will confirm the performance-strengthening capabilities of the adaptive storage solution, but the current assessment demonstrates sufficient evidence of system efficiencies.

4.5. Limitation

Despite this, there are also some smaller limitations as examples of possible suboptimal performance during early training. However, the simplicity process may neglect some potentially beneficial data onto lower retention efficiency in early epochs (88.5% CDRE at early epoch 25). Furthermore, fine-tuning of parameters might be needed in the event of highly dynamic environments where the nature of critical data is changing from time to time [24]. Although these limitations are minor, they indicate that real-time adaptability and optimization can improve DDSS performance in complex and diverse settings.

5. Conclusion

The results of this study demonstrate the interoperation of the ACLDC and the DDSS to enhance IoT data management for real-time safety monitoring, especially in women's safety applications. Data into ACLDC is compressed without loss, with a CR of 4.5 at epoch 100 and a DRI of 98%, ensuring data integrity yet reducing storage and transmission demands. This is complemented by DDSS in selectively using non-critical data, achieving a CDRE of 96.8% and an SSE of 85.4% using the same epoch. Together, these techniques allow for efficient data handling while maintaining crucial information for decision-making as little resource is consumed. Nevertheless, though both ACLDC and DDSS initially have lower efficiency, dynamic environments will require some further parameter optimization to reach peak performance. In general, combining ACLDC and DDSS provides a reliable and reasonable approach for the real-time IoT system; meanwhile, the system can tackle a massive amount of data streams while maintaining accuracy and speed.

Future work can focus on developing more advanced algorithms for real-time parameter optimization in order to improve the adaptability of ACLDC and DDSS to highly dynamic environments. Moreover, these methods may be easily integrated with emerging machine learning methods (such as reinforcement learning) that would boost their capacity to dynamically reconfigure themselves according to different data patterns [25]. This can further explore the possibilities of combining these approaches with edge computing frameworks for distributed IoT systems in order to make scalability and efficiency of real-time applications.

References

[1] K.M. Karthick Raghunath et al., “Utilization of IoT-Assisted Computational Strategies in Wireless Sensor Networks for Smart Infrastructure Management,” *International Journal of System Assurance Engineering and Management*, vol. 15, pp. 28-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[2] C.E.A. Winslow, L.P. Herrington, and A.P. Gagge, "Physiological Reactions of the Human Body to Varying Environmental Temperatures," *American Journal of Physiology-Legacy Content*, vol. 120, no. 1, pp. 1-22, 1937. [CrossRef] [Google Scholar] [Publisher Link]

[3] K.M. Karthick Raghunath, and N. Rengarajan, "Response Time Optimization with Enhanced Fault-Tolerant Wireless Sensor Network Design for On-Board Rapid Transit Applications," *Cluster Computing*, vol. 22, pp. 9737-9753, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ihab Nassra, and Juan V. Capella, "Data Compression Techniques in IoT-Enabled Wireless Body Sensor Networks: A Systematic Literature Review and Research Trends for QoS Improvement," *Internet of Things*, vol. 23, pp. 1-29, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[5] Chunsheng Liu et al., "Robust Online Tensor Completion for IoT Streaming Data Recovery," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 12, pp. 10178-10192, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[6] J. Schoeman, and L.P. Linde, "Employing a Measure of Sparseness to Investigate Sparse Data Compression in Awgn Conditions," *SAIEE Africa Research Journal*, vol. 97, no. 2, pp. 157-161, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[7] Charles M. Stein et al., "Latency-Aware Adaptive Micro-Batching Techniques for Streamed Data Compression on Graphics Processing Units," *Concurrency and Computation: Practice and Experience*, vol. 33, no. 11, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Suha Abdulhussein Abdulzahra, Ali Kadhum M. Al-Qurabat, and Ali Kadhum Idrees, "Compression-Based Data Reduction Technique for IoT Sensor Networks," *Baghdad Science Journal*, vol. 18, no. 1, pp. 184-184, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[9] Suayb S. Arslan, and Turguy Goker, "Compress-Store on Blockchain: A Decentralized Data Processing and Immutable Storage for Multimedia Streaming," *Cluster Computing*, vol. 25, pp. 1957-1968, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Dezhao Wang et al., "Neural Data-Dependent Transform for Learned Image Compression," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17379-17388, 2022. [Google Scholar] [Publisher Link]

[11] Jianming Wang, Wei Wang, and Jianhua Chen, "Adaptive Rate Block Compressive Sensing Based on Statistical Characteristics Estimation," *IEEE Transactions on Image Processing*, vol. 31, pp. 734-747, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[12] Chunwei Liu et al., "Decomposed Bounded Floats for Fast Compression and Queries," *Proceedings of the VLDB Endowment*, vol. 14, no. 11, pp. 2586-2598, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[13] N. Erratt, and Y. Liang, "Compressed Data-Stream Protocol: An Energy-Efficient Compressed Data-Stream Protocol for Wireless Sensor Networks," *IET Communications*, vol. 5, no. 18, pp. 2673-2683, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[14] Sicong Liu et al., "AdaSpring: Context-adaptive and Runtime-Evolutionary Deep Model Compression for Mobile Applications," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 1, pp. 1-22, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[15] Priyan Malarvizhi Kumar et al., "Clouds Proportionate Medical Data Stream Analytics for Internet of Things-Based Healthcare Systems," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 3, pp. 973-982, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[16] Hanqing Sun et al., "Intelligent Analysis of Medical Big Data Based on Deep Learning," *IEEE Access*, vol. 7, pp. 142022-142037, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[17] H. Fouad et al., "Analyzing Patient Health Information Based on IoT Sensor with AI for Improving Patient Assistance in the Future Direction," *Measurement*, vol. 159, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] I. Daubechies et al., "Nonlinear Approximation and (Deep) Networks," *Constructive Approximation*, vol. 55, pp. 127-172, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[19] Wasim Akram, Mohit Jain, and C. Sweetlin Hemalatha, "Design of a Smart Safety Device for Women Using IoT," *Procedia Computer Science*, vol. 165, pp. 656-662, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[20] Gabriel Signoretti et al., "An Evolving TinyML Compression Algorithm for IoT Environments Based on Data Eccentricity," *Sensors*, vol. 21, no. 12, pp. 1-25, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[21] Zeyu Sun et al., "Edge Computing in Internet of Things: A Novel Sensing-Data Reconstruction Algorithm Under Intelligent-Migratoin Stragegy," *IEEE Access*, vol. 8, pp. 50696-50708, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Ata Ullah et al., "Secure Critical Data Reclamation Scheme for Isolated Clusters in IoT-Enabled WSN," *IEEE Internet of Things Journal*, vol. 9, no. 4, pp. 2669-2677, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[23] Márcio Miguel Gomes et al., "Simplifying IoT Data Stream Enrichment and Analytics in the Edge," *Computers & Electrical Engineering*, vol. 92, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[24] Jiang Bian et al., "Machine Learning in Real-Time Internet of Things (IoT) Systems: A Survey," *IEEE Internet of Things Journal*, vol. 9, no. 11, pp. 8364-8386, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[25] J. Hillman, and I. Warren, "An Open Framework for Dynamic Reconfiguration," *Proceedings. 26th International Conference on Software Engineering*, Edinburgh, UK, pp. 594-603, 2004. [CrossRef] [Google Scholar] [Publisher Link]