Comparison of Statistical and Machine Learning-Based Approaches for Telemetry Data Size Reduction

Ahmad El Sayed[1], Marc Ruiz[1]*, Hassan Harb[2], and Luis Velasco[1]

[1] *Optical Communications Group (GCO), Universitat Politècnica de Catalunya, Barcelona, Spain*

[2] *College of Engineering and Technology, American University of the Middle East, Kuwait*

**e-mail: marc.ruiz-ramirez@upc.edu*

ABSTRACT

The implementation of next generation beyond 5G use cases such as the Digital Twin (DT) requires the transportation of increasingly large amounts of sensor telemetry data. The reduction of the size of these time series shaped data through compression has many benefits, ranging from energy savings in systems where sensors have limited power to transport network bandwidth reduction. Both statistical analysis-based and machine learning (ML)-based approaches have been employed to tackle this task with varying degrees of success. In this paper, we compare two methods for time series data compression: a statistical similarity-based one, and an autoencoder (AE)-based one.

Keywords: Telemetry Data, B5G, Compression, Statistical Similarity, Auto-Encoders

1. INTRODUCTION

One of the most notable use cases of the beyond 5G (B5G) era is the Digital Twin (DT) [1]. It is a high bandwidth consuming application and requires innovation in the manner which the collected sensor telemetry data, essential for its operation. This data is typically transported from its distributed components to its central component, i.e., DT manager (DTM). This DT use case, as well as other B5G use cases, need from pervasive monitoring solutions to deal with efficient and reliable collection and processing of telemetry data from largely distributed and heterogeneous data sources [2].

One of the goals of telemetry data processing is to reduce the volume to be conveyed from sources to the central manager without losing neither veracity nor value of the collected data [3]. The reduction of the size of transported telemetry data is key in order to avoid overwhelming the transport layer, and to reduce the network operation cost and complexity. It also has other benefits such as reducing the amount of power the sensors consume during data transmission, which is extremely important in the cases where the sensors have a limited power source such as a battery [4]. Moreover, data analysis in order to perform compression at the sensor node offer the opportunity to perform important tasks, such as anomaly detection, which can provide early signs of problems [5].

Traditionally, reducing the size of collected data can be achieved by either lowering the sampling rate, i.e., increasing the period between the collection of data points, or by compacting the volume of the data through compression [6]. Another approach is to use machine learning (ML) based models to perform the compression, which indeed is a recent trend in autonomous network operation, e.g., using auto-encoders (AE) with the potential to perform such compression tasks with higher efficiency [7].

This paper presents a comparison of two distinct approaches for telemetry data size reduction to be applied close to telemetry data sources. The first, named *Zoom-In Zoom-Out* (ZIZO) [8], exploits the redundancy in collected telemetry data to compress it by aggregating values of high similarity, with an algorithm called index bit encoding (IBE). In addition, ZIZO also tries to minimize collected data by adapting the sensing rate to the redundancy levels measured in a group of close proximity sensors through statistical analysis, using an algorithm called Sensing Rate Adaption (SRA). The second approach, named *Adaptive AE-based Compression* (AAC) [9] employs AEs for compression, while also performing localized anomaly detection. It makes use of their intrinsic dimensionality reduction abilities to reduce the size of transported data by sending through the network only the set of latent features encoded at the source.

2. REFERENCE ARCHITECTURE

Figure 1 presents the hierarchical architecture needed to run the proposed telemetry data compression and analysis. The first level is at the sensors layer where data is generated periodically. For the sake of simplicity, let us assume that sensors are those physical elements that are able to monitor one specific metric, e.g., temperature, pressure, etc. Then, a number of these sensors are integrated in a monitoring device, that provides the support (computing, power) to those sensors, as well as contains the needed transceivers and interfaces (wired or wireless) required to eject the data out of the device. Since the vast majority of multi-purpose monitoring devices are built on top of powerful boards, a software-based Device Agent (DA) is deployed in the device for several purposes including telemetry data processing and device control and management. Specifically, in the context of our work, we consider that the DA contains the algorithms necessary to compress the collected telemetry data.

Then, the DA sends the compressed data to the DTM that is hosted in the remote location. Along with the compressed data, metadata containing device/sensors identification data and compression method metadata are included.

The second element in the proposed hierarchical architecture is the Cluster Agent (CA) that runs as one of the processes in DTM, and aggregates the inputs received from a number of devices that form a group (*cluster*). The meaning of a cluster is open: can represent any subset of monitoring devices in a physical subsystem. Without loss of generality, we assume that the creation of clusters is part of the design of both the physical system and DT, which is out of the scope of this paper.

Each CA is in charge of decompressing the data received from its nested DAs and store such decompressed data in the Data Lake. Moreover, it is also in charge of training models (if needed) as soon as new relevant data is collected and uploading new models to the DAs in an automatized manner. Finally, it processes other actions such as: higher level aggregation, anomaly detection diagnosis reports received from DAs, and multiple anomaly detection if needed.

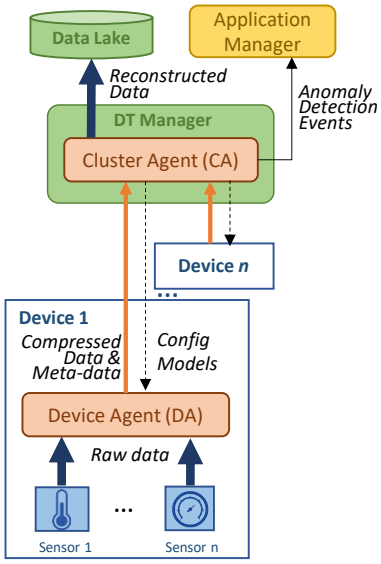

Figure 1: Overall Architecture

3. STATISTICAL SIMILARITY-BASED REDUNDANCY REDUCTION (ZIZO)

In telemetry systems (such as most of sensing-based systems), there are some monitored parameters that have a slow variation rate in their values. As a consequence of this, the sensors collecting such monitoring data would have very high redundancy levels, especially if this is done periodically. The values could remain constant for a long period of time or vary only occasionally from this fixed value. ZIZO exploits this, and acts at two different stages.

Firstly, the IBE algorithm runs in the DA and groups similar values in the set of measurements of fixed length T and saves the position of these similar values in an index byte. Thus, for telemetry data stream i, two different measurements r^i_j and r^i_k are similar if the absolute value of the difference between them is less than the parameter ε, which is formally defined in eq, (1). Figure 2 shows an illustrative example of the compression and decompression of the set of measurements through IBE using similarity function with parameter ε = 0.8 and T=8. As can be observed, applying the similarity function will find three different groups of similar data (each one represented by a color). Then, the coding stores, for each group, the reference value and the coding of the bits defining the position. Note that, once the coded data is decompressed, the information lost by aggregation is not recovered.

At a second stage, a set of measurements from different sensors are collected in the CA, where matrix M^p is obtained (Eq. 2). This matrix is created by stacking the transposed sets, making each row of the matrix containing measurements of different sensors at the same instant, and making each couple of consecutive rows a set of paired observations.

The statistical *T-test* is then used to calculate the level of similarity between all the possible pairs of consecutive rows. The test is formally described in Eq. (3), where $\bar{X}^{t,t+1}$ is the mean of the difference set $\{R^{t+1} - R^t\}$, μ is the population mean of the whole matrix M^p, and $\sigma^{t,t+1}$ is the standard deviation of the difference set $\{R^{t+1} - R^t\}$.

$$\Delta_{r^i_j, r^i_k} = |r^i_j - r^i_k| \leq \varepsilon \qquad (1)$$

$$R_1{}^p = \begin{matrix} \\ r_1^1 \\ r_1^2 \\ \vdots \\ r_1^{\mathcal{T}} \end{matrix} \begin{matrix} s_1 \\ \begin{pmatrix} r_1^1 & r_2^1 & \dots & r_n^1 \\ r_1^2 & r_2^2 & \dots & r_n^2 \\ \vdots & \vdots & \dots & \vdots \\ r_1^{\mathcal{T}} & r_2^{\mathcal{T}} & \dots & r_n^{\mathcal{T}} \end{pmatrix} \end{matrix} \begin{matrix} s_2 & \dots & s_{\mathcal{T}} \end{matrix} \qquad (2)$$

$$T - test(R^t, R^{t+1}) = \frac{\bar{X}^{t,t+1} - \mu}{\frac{\sigma^{t,t+1}}{\sqrt{n}}} \qquad (3)$$

R_i^p : [10.0, 12.1, 12.2, 10.6, 11.4, 14.4, 13.9, 9.2]

⇓ *Similarity function*

ε = 0.8 : [10.0, 12.1, 12.2, 10.6, 11.4, 14.4, 13.9, 9.2]

⇓ *code of reading*

8-bit : [(10.0, 10010001), (12.1, 01101000), (14.4, 00000110)]

⇓ *code weight*

$R_i'^p$: [(10.0, 145), (12.1, 104), (14.4, 6)]

⇓ *decompression*

$R_i''^p$: [10.0, 12.1, 12.1, 10.0, 11.4, 11.4, 11.4, 10.0]

Figure 2: IBE Example

Finally, the redundancy ratio of the period is calculated through dividing the number of similar consecutive pairs of sets of measurements by the total number of possible successive pairs ($T - 1$). After calculating the redundancy

ratio, the telemetry frequency can be adapted for all sensors and probes, increasing or decreasing within certain boundaries defined also by the application, with the aim of reducing the redundancy ratio in the next period. Other metrics such as the criticality of the application are also used to determine the rates.

4. AUTOENCODER-BASED COMPRESSION (AAC)

As already introduced, AAC uses AEs to perform time series data compression. It builds upon the innate ability of AEs to reduce the dimensionality of the input data through extracting a set of latent feature representations. Specifically, the DA contains a pool of AEs of different latent space size in order to perform such adaptive compression. Figure 3 sketches the main ACC procedure running at DA.

At the end of each period, when a set of measurements equal to the input size of the AEs is ready, it passes it through all the AEs in the pool. Starting from the one with the smallest latent space size, the input data is compressed and then decompressed locally in order to compute the reconstruction error (RE) between the original and decompressed data. If RE is below the Acceptable Reconstruction Error (ARE) parameter, it selects such AE and uses its encoder part to perform the compression before sending the set latent space to the destination. Otherwise, if RE is higher than ARE, it moves on to the larger latent space AEs.

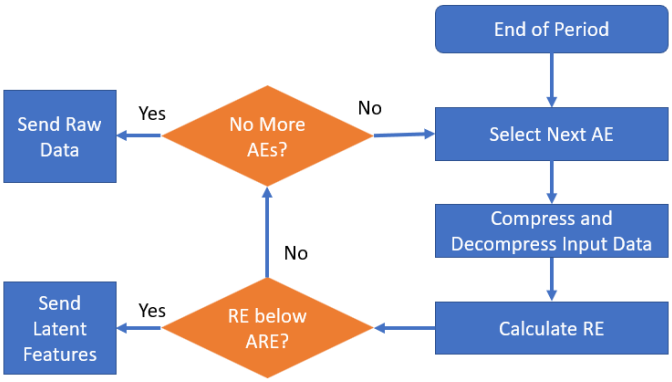

Figure 3: ACC procedure

Note that, if none of the AEs can compress with RE below ARE, raw uncompressed data is retrieved. Although this algorithm is ML-based and requires data for the models to be trained, generic models, using synthetic data, or data with characteristics like the ones to be collected by the deployed sensor, can be trained for early deployment. These can perform a compression under less strict constraints while data from the target sensor is being collected to train the specific models. Then, these models are introduced to the DA by the CA.

5. PERFORMANCE COMPARISON

Different datasets were used to evaluate the performance of each algorithm. In the case of ZIZO, two small datasets were considered: *i*) the Intel Berkeley Research Lab sensor data which contained sensors for temperature, humidity, and luminosity, and *ii*) experimental dataset using several telsoB sensors. The variables involved included the input size, the similarity threshold, and the application criticality. The dataset used to test AAC was a subset of the Water Distribution test bed dataset, selected sensors monitored the water pressure valve, measuring pressure, voltage, and flow. Figure 4 shows an illustrative example of the telemetry data included in the aforementioned data sets [10].

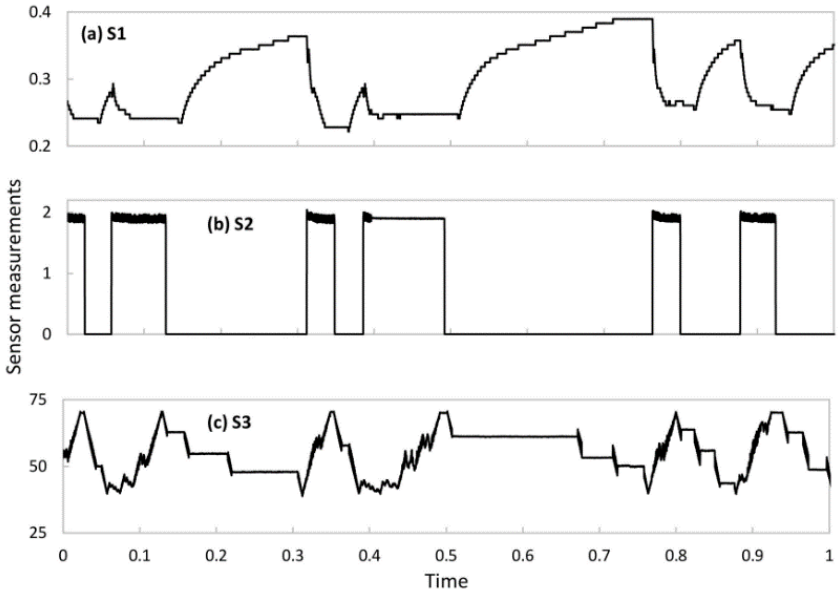

Figure 4: Example of considered telemetry data

Table 1 shows the eventual configuration of each algorithm after exhaustive testing on the data described above. While the input size for ZIZO was variable between 32 and 64, it was fixed at 256 for AAC. This means longer data collection periods for AAC, however it makes it more scalable than ZIZO since IBE is constrained by the index bytes, while AAC can be scaled down to 128 or scaled up to 512, for example. The application criticality in ZIZO also is variable between low and high, which is reflected in the adapted sensing periods. However, the criticality of the application can be adjusted for by changing the ARE in AAC, from 0.01 being very strict and 0.05 being very loose. Finally, the size of the compressed data depends on the similarity threshold only for ZIZO. In AAC, it depends both on the ARE and the available latent space sizes, with a possibility of having a compressed data size equal to 1% of the original data size if selected AE is the one with latent space size equal 4.

Table 1: Parameter Ranges for each Algorithm

ZIZO		**AAC**	
Period Size	32 and 64	*Input Size*	256
Application Criticality	high and kow	*ARE*	[0.01, 0.05]
Similarity Threshold	temperature: 0.07, 0.1, 0.2 humidity: 0.2, 0.5, 1 light: 10, 15, 25	*Latent Space Size*	4, 8, 16, 32

6. CONCLUSIONS

In this paper two algorithms for time series sensor data compression were compared, the first having a statistical analysis-based approach to aggregate values based on their similarity, and vary sensing rates based on redundancy, and the other using pools autoencoders to adaptively compress input data based on acceptable reconstruction errors. Figures that detail how the algorithms are applied were supplied. Table 2 summarizes the comparison of the characteristics of each algorithm.

Table 2: Summary of characteristics

	ZIZO	**AAC**
Basis	Statistical Analysis	ML
Data Collection	Periodic	Periodic
Requires Training	No	Yes
Adaptive	Yes	Yes
Scalable	No	Yes
Configurable	Yes	Yes

ACKNOWLEDGEMENTS

This research was funded by European Commission through the HORIZON DESIRE6G project (G.A. 101096466), by the AEI through the IBON project (PID2020-114135RB-I00), and by ICREA institution.

REFERENCES

[1] B. R. Barricelli, C. Elena and D. Folgi, "A Survey on Digital Twin: Definitions, Characteristics, Applications, and Design Implications," IEEE Access, vol. 7, pp. 167653-167671, 2019.

[2] L. Velasco, M. Ruiz, P. Gonzalez, F. Paolucci, A. Sgambelluri, L. Valcarenghi, C. Papagianni, "Pervasive Monitoring and Distributed Intelligence for 6G Near Real-Time Operation," accepted in European Conference on Networks and Communications (EUCNC), 2023.

[3] L. Velasco, S. Barzegar, and M. Ruiz, "Is Intelligence the Answer to Deal with the 5 V's of Telemetry Data?," in Proc. Optical Fiber Communication Conference (OFC), 2023.

[4] S. Tarannum, "Energy Conservation Challenges in Wireless Sensor Network: A Comprehensive Study," Wireless Sensor Network, vol. 2, pp. 483-491, 2010.

[5] L. Velasco *et al*, "Monitoring and Data Analytics for Optical Networking: Benefits, Architectures, and Use Cases," IEEE Network, vol. 33, no. 6, pp. 100-108, 2019.

[6] S. Pushpalatha and K. Shivaprakasha, "Energy-Efficient Communication Using Data Aggregation and Data Compression Techniques in Wireless Sensor Networks: A Survey," in Advances in Communication, Signal Processing, VLSI, and Embedded Systems, Singapore, Springer Singapore, 2020, pp. 161-179.

[7] M. Ruiz, D. Sequeira, and L. Velasco, "Deep Learning -based Real-Time Analysis of Lightpath Optical Constellations [Invited]," IEEE/OPTICA Journal of Optical Communications and Networking (JOCN), vol. 14, pp. C70-C81, 2022.

[8] A. El Sayed, H. Harb, M. Ruiz and L. Velasco, "ZIZO: A Zoom-In Zoom-Out Mechanism for Minimizing Redundancy and Saving Energy in Wireless Sensor Networks," IEEE Sensors Journal, vol. 21, no. 3, pp. 3452-3462, 2021.

[9] A. El Sayed, M. Ruiz, H. Harb and L. Velasco, "Deep learning-based Adaptive Compression and Anomaly Detection," MDPI Sensors Journal, vol. 23, no. 2, p. 1043, 2023.

[10] iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design, Water Distribution Dataset, Singapore, 2019