Data Compression Strategies for Enhancing IoRT Communications over Heterogeneous Terrestrial-Satellite Networks

Alexander Karnaukhov[†], Artem Idelevich[†], Alexey Rolich[*†] , and Leonid Voskov[†]
[*]Dept. of Information Engineering, Electronics and Telecommunications (DIET), University of Rome Sapienza, Italy
[†]HSE University, Russia
alexey.rolich@uniroma1.it asidelevich@edu.hse.ru
ayukarnaukhov@edu.hse.ru lvoskov@hse.ru

***Abstract*—Data transmission over low-speed networks is a subject of global research interest, with a specific focus on overcoming limitations in satellite communication channels. However, little research has addressed the impact of compression on data transmission efficiency in heterogeneous networks using satellite links, especially in the context of the Internet of Remote Things and under poor terrestrial network infrastructure. This study explores various data compression algorithms customized for tiny data in such scenarios. It identifies effective algorithms when combined with ProtoBuf serialization, achieving compression ratios between 3.5 and 8.2 for JSON messages of 30 to 680 bytes using the Huffman algorithm with an extended dictionary. For messages sized 680 to 2,048 bytes, Protobuf combined with LZ78 achieves compression ratios of 3.5 to 4.5. Moreover, a novel data preprocessing method is introduced, boosting processing performance by up to 27 times for messages under 680 bytes in size. These findings contribute to IoRT data serialization and compression and can potentially enhance existing methods.**

***Index Terms*—Data compression, Serialization, Internet of Remote Things, IoRT, JSON, Protobuf, LEO, Terrestrial-Satellite**

I. Introduction

In areas without network infrastructure, satellite communication systems (SATCOM) [1] are used to collect and exchange data, such systems are called the Internet of Remote Things (IoRT) [2]. Integrated communication systems and heterogeneous networks for data collection using SATCOM are being intensively developed. These networks combine satellite and terrestrial networks. The generally recognized limitations of data acquisition systems using SATCOM are the high cost of transmitting messages from sensors and the low bandwidth of communication channels, which does not allow widespread implementation of SATCOM to connect Internet of Things (IoT) [3] devices.

The primary concern when transmitting messages in a heterogeneous network via satellite communication in the IoRT is the cost, payload increase, and reduction in energy consumption. These factors have a direct impact on the practical adoption of IoRT technologies across industries. One crucial element affecting both cost and energy efficiency in such transmissions is the data volume relative to the overhead, often referred to as the overhead-to-payload ratio. Optimizing this ratio can be accomplished by adjusting existing network algorithms, protocols, and employing data compression techniques. In a heterogeneous structure, novel algorithms, approaches, and compression methods can be implemented at the interface between two networks, a distinctive feature of this work compared to existing methods.

In previous articles [4]–[6] we have noted the problems when deploying data collection networks in regions without any existing infrastructure, several areas of application of the Internet of remote things. In [5], we proposed a data encoding and packaging scheme called Gateway Data Encoding and Packaging (GDEP). It is based on a combination of data format transformation at the junction points of a heterogeneous network and message packaging. GDEP has reduced the number of Short Burst Data (SBD) containers used and the total size of transmitted data by 4.82 times. To further improve efficiency, in [6] we proposed Gateway Data Encoding, Packaging and Compression (GDEPC) method to additionally use transmission data compression and put forward a hypothesis that the use of a hybrid data compression method can lead to a decrease in the total size of the transmitted payload data over the Iridium satellite channel, which in turn can lead to a decrease in the cost transmission of fixed data volumes in IoRT conditions.

In this article, we consider the scenario of using the Lora-Iridium heterogeneous network to organize network coverage in the IoRT for several reasons: LoRaWan is one of the most energy-efficient technologies for creating a data collection system from end devices, while Iridium technology is one of the most widespread and affordable satellite communication technologies that provides complete coverage of the globe. Furthermore, the primary objective of this research is to identify the most efficient algorithms for data serialization and compression tailored for the IoRT scenarios. Given the constraints of a satellite channel with a maximum message size of 2000 bytes, our investigation is specifically geared towards the serialization and compression of small-scale data in the realm of the IoRT.

The main contributions of this paper are to:

- conduct a study on the performance of compression methods within the context of SBD messages;
- investigate the impact of serialization on the compression performance of tiny IoRT data;
- propose a novel method for data preprocessing before serialization.

The paper is organized as follows. Section II provides an overview of data compression methods and analyzes the related works on the performance evaluation. Section III describes the effectiveness of different data compression techniques, emphasizing their suitability for tiny IoT message payloads. In Section IV, we conduct an investigation into the integration of compression and serialization employing

Protocol Buffers (Protobuf), followed by a discussion of the experiment's findings. Section V concludes the paper.

II. Background and Related Works

In [7], the authors conducted a systematic review of lossy data compression algorithms for IoT systems, aiming to address the challenges posed by the vast amount of data generated by IoT devices. They categorized and analyzed 30 different approaches, classifying them into four categories: hybrid, artificial intelligence, interpolation, and transform methods. The review highlighted that interpolation approaches are computationally efficient and suitable for real-time data compression, while hybrid methods combine lossy and lossless techniques but may have higher computational complexity. Transform methods convert data into different domains, showing good compression results but requiring a reasonable number of measurements. Artificial intelligence methods are complex and often involve pre-training. The study also noted that outlier treatment and data compression are typically addressed separately in the literature, suggesting the potential for future research to explore their combined analysis. Overall, lossy data compression was recognized as a valuable strategy to optimize IoT resource utilization and enhance economic and energy viability in industrial IoT systems facing data growth challenges.

In [8], the authors reviewed XML and JSON compression techniques, introducing a novel method that leverages the inherent structure of documents, resulting in significant compression improvements for web-based systems, particularly in cases involving small documents, outperforming existing state-of-the-art techniques.

Perez et al. [9] addressed the energy consumption challenges in battery-powered IoT applications using GNSS-based sensors by proposing a novel methodology for reducing the energy footprint. Their approach involved preprocessing online trajectory data without information loss and compressing it using standard lossless algorithms. Simulations across various trajectory shapes revealed that the proposed scheme achieved lower compression rates, effectively mitigating the energy footprint of embedded IoT devices. The methodology included generating simulated NMEA inputs for a travel path, identifying repetitions in geographic coordinate components, and employing counters for recording different entries. After preprocessing and encapsulating the data into a JSON document, compression was applied to minimize memory space and reduce transmission energy costs. The study highlighted a trade-off between compression methods relying on character occurrence probabilities and those using repetition counts, with Huffman coding performing well. Overall, the compressed set occupied an average of 30% of the original memory space, showcasing notable improvements. Future work may involve practical testing, such as aerial drone monitoring.

In [10], the authors addressed the challenges posed by the transmission of large volumes of data generated by electric smart meters in bandwidth and storage-constrained IoT systems. They proposed a novel two-step compression scheme, called AMDC, specifically designed for multivariate smart meter data. The AMDC algorithm leverages cross-correlation between different variables to reduce data dimensionality and then exploits temporal correlation in individual data streams to achieve efficient compression without significant information loss. Through extensive testing on real smart meter data, the authors demonstrated that AMDC can reduce bandwidth requirements for transmitting multivariate smart meter data over actual communication networks by up to 98.5%, all while ensuring faithful data reconstruction at the aggregator with an acceptable error threshold. The algorithm's performance was compared to state-of-the-art methods, showcasing its significant advantages in bandwidth saving without compromising data fidelity. Additionally, the authors highlighted the adaptability of AMDC to varying smart meter data dynamics through online parameter tuning. Overall, the proposed compression algorithm proved to be highly efficient, with minimal execution time overhead, making it a valuable solution for optimizing network resource usage in real-life smart metering applications while maintaining quality of service for consumers.

Sokol and Hubinský [11] addressed the challenge of reducing data size in IoT applications, specifically focusing on JSON format, and proposed a non-standard data compression method to minimize computational demands on devices, which is crucial for battery-powered devices with limited capacity; their tests, conducted with 10 devices, demonstrated a consistent 13% data compression rate, with the potential for further improvement to approximately 25% by implementing a 2-byte substitution, leading to promising compression efficiency, and future work will involve building and testing dictionaries in practical operational scenarios. In other paper Sokol et al. [12] addressed the challenges posed by the burgeoning IoT era, where millions of devices require efficient communication while minimizing transmission delay. Recognizing the strain this puts on communication channels, they propose a method for reducing data volume between IoT devices and servers. This reduction is achieved through data compression and the replacement of the SSL/TLS cryptographic protocol with lightweight cryptography based on the Vernam cipher principle, while still retaining SSL/TLS for device management. The focus is on transmitting short messages, which, in their original JSON form, are around 100 bytes in size (up to 500 characters). By employing this nonstandard compression technique, the size of these messages can be reduced to 20% of their original size. Additionally, eliminating the SSL/TSL protocol from the data transfer process can reduce the total transmitted data to less than 10% of its original volume, with the majority of savings coming from encrypted connection data. This solution addresses the growing demands on transmission bandwidth, energy efficiency, and data security in the IoT landscape, providing an approach that optimizes data transmission, accelerates communication, reduces energy consumption on IoT devices, and ensures the security of transmitted data.

In [13], the authors proposed and demonstrated an alternative approach to accessing and encapsulating synchrophasor data using the widely used JSON format, enabling easier data analysis, collaboration, and interoperability while accommodating all existing data within C37.118 dataframes. They emphasized the advantages of JSON, including its readability, flexibility, and compatibility with multiple programming languages, fostering rapid development and collaborations with accepted standards. The authors also highlighted the potential for moving towards IoT methods in PMU data handling, leveraging JSON and related security tools for communication, and encouraging novel applications by incorporating data from various sources.

Azar et al. [14] addressed the energy consumption challenges in IoT systems, particularly in the context of data transmission to the cloud, by proposing an energy-efficient approach for IoT data collection and analysis. They utilized a fast error-bounded lossy compressor to reduce data size before transmission, significantly decreasing the amount of transmitted data (up to 103 times) without compromising data quality. Their approach involved rebuilding and processing data on edge nodes using supervised machine learning techniques, particularly in the context of monitoring driver stress levels in intelligent vehicle systems with medical data collected from wearable devices. The results demonstrated that the compression did not impact data analysis and classification accuracy, highlighting the potential to extend the lifetime of IoT devices while reducing energy consumption and network latency.

In [15], the authors focused on addressing energy-saving challenges in IoT solutions, particularly by reducing energy consumption during data transmission. They developed a novel coding scheme for delta compression, designed to efficiently compress temporally correlated data, such as temperature measurements from various smart devices. Their coding scheme demonstrated the potential to achieve remarkable energy savings, up to 85%. Compared to other coding techniques, their approach offered a higher compression ratio and lower memory requirements. The study emphasized the applicability of delta coding for energy-saving purposes in IoT environments, especially in scenarios where real-time sensor measurements are not essential.

The studies discussed in Section 2 provide significant insights into the domain of LoRa-Iridium networks and IoT data compression methods. However, these studies primarily focus on conventional IoT scenarios, Industrial Internet of Things, edge computing, and the utilization of machine learning techniques to tackle data compression issues. They primarily address the compression of substantial volumes of data and do not explore the specific challenges and techniques associated with compressing tiny data, which is essential in specialized IoRT contexts.

III. Evaluating compression techniques for tiny IoT data

A. LoRa-Iridum network IoRT scenario

Today, there are two main providers of digital data transmission for the satellite transport network: Inmarsat and Iridium. Inmarsat's satellites belong to the class of geosystems that provide coverage up to 70° north and south latitude, i.e. they cover almost the entire globe. The satellite system provides Internet connectivity at speeds of up to 384 Kbps. A key feature of the platform is the presence of a controller that allows you to implement applications in a short time. There are ready-made machine-to-machine (M2M) solutions, the choice of which depends on the complexity, speed and volume of information sent and received. Such solutions include IsatM2M, IsatDataPro and BGAN M2M.

The Iridium satellite system is a Low Earth Orbit (LEO) constellation offering data rates of up to 128 Kbps. The operating company offers M2M equipment based on Short Burst Data (hereafter referred to as SBD) 9602, 9603N and 9523 satellite modems, which enable the transmission of short data messages. SBD satellite modems are characterized by low power consumption and the ability to exchange text messages.

The need to provide communication services in remote areas leads to the need to create a coverage area in the Far North and the Arctic. The only satellite network that meets this criterion is Iridium. Connecting an SBD Iridium modem to each IoT sensor is not feasible, both in terms of cost and power consumption. Therefore, to organize the connection of IoT sensors via satellite communication channels, it is necessary to provide a gateway between the local network and the satellite communication channels.

In the GDEPC method, data originating from cyber-physical systems can manifest in different formats, including JSON, GeoJSON, CSV, XML, and more. To reduce data volume, each structure undergoes serialization, converting it into a sequence of bytes. An efficient serialization tool utilized in this context is Protobuf, as demonstrated in a study by Lysogor et al. [16]. Protobuf can achieve a remarkable data reduction of up to 4.6 times when applied to JSON data. These serialized Protobuf messages are then packaged into compact packets, constrained by the maximum size of SBD containers compatible with the Iridium 9602 modem, which is 340 bytes.

B. IoRT data compression experiment

The study opted for JSON as the data format due to its advantages, including ease of reading and writing, lightweight nature, broad support, straightforward decoding, versatility with data types, platform independence, extensibility, support for Unicode characters, and widespread usage in the IoT for data transmission.

As a component of the experimental investigations, synthetic JSON messages of varying lengths were generated, incorporating both alphabetic and digital symbols, while the structure of JSON files closely emulated the traditional message formats commonly employed in message transmission protocols.The study operated under the assumption that the JSON object's structure and the predefined keys' values were known in advance. These keys encompassed parameters such as temperature, time, humidity, pressure, and other commonly utilized environmental monitoring system attributes, where each parameter's value could be represented by numerical or string values.

The primary objective of this experiment is to identify the optimal compression format that can achieve the highest compression ratio, a crucial factor when working within the constraints of the Iridium satellite channel, which imposes strict limitations on data transmission volume. Consequently, the experiment encompasses data sizes ranging from 100 to 2048 bytes. The central hypothesis guiding this research is that various data sizes may yield distinct compression ratios contingent on the chosen compression algorithm. Therefore, a comprehensive analysis of compression ratios was undertaken across a spectrum of file sizes, commencing at 100 bytes and progressively increasing in 10-byte intervals up to 2048 bytes. The study examined eight compression algorithms, including Brotli, Lzma, Smaz, Zstd, LZ77, LZ78, LZW, Huffman with extended dictionary. For each message size and every distinct compression algorithm, a minimum of 400 tests were conducted.

In Figure 1, you can observe the outcomes of experiments conducted using diverse compression algorithms and varying message sizes. Notably, the Huffman algorithm with an extended dictionary outperforms the others, particularly for very small messages (less than 360 bytes), achieving compression

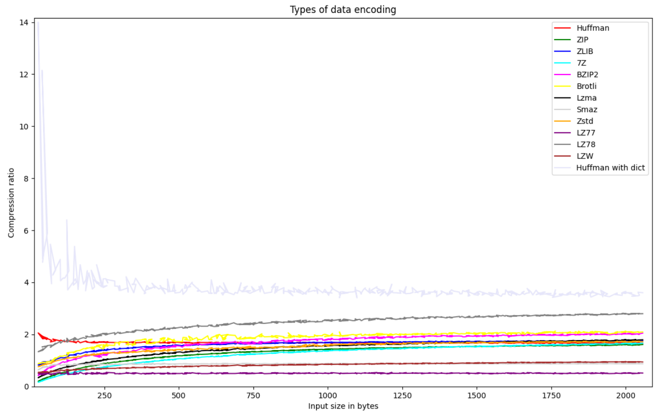

Figure 1. Comparison of compression algorithms. Compression ratio as a function of data size.

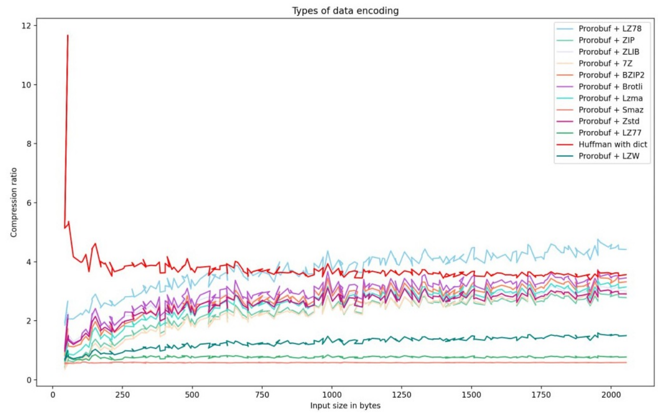

Figure 2. Comparison of compression algorithms after Protobuf serialization. Compression ratio as a function of data size.

ratios ranging from 3.6 to 4.1. However, it's essential to acknowledge that employing an algorithm with an extended dictionary imposes synchronization constraints between the transmitter and receiver, necessitating dictionary storage on both ends. This approach remains viable within the considered architecture, as it assumes message structures remain constant in IoRT conditions, obviating the need for dynamic dictionary alterations or additions. It's worth mentioning that the LZ-78 algorithm shows promise as a compression method for data in the context of the IoRT.

IV. Serialization and compression: an experimental study and discussion

A. *Serialization and compression performance in IoRT experiments*

In this experiment, we investigate the synergy between serialization in the form of Protobuf and the utilization of diverse compression algorithms to gain insights into enhancing data compression through these combinations. Consequently, graph two depicts the outcomes of compression and serialization, as discussed in Section III. The selection of ProtoBuf for serialization was not arbitrary; it was based on the findings presented in [16], which demonstrated that ProtoBuf outperforms other data serialization approaches in the context of the IoRT, showcasing superior performance.

The data preprocessing process unfolds as follows: starting with the initial raw JSON message, it undergoes serialization using Protobuf, transitioning the JSON message into a Protobuf structure. Following this, the complete implementation of this Protobuf structure is transformed into binary format. Subsequently, the data is compressed using the chosen compression algorithm, resulting in the final message ready for transmission across the heterogeneous IoRT network. It's important to highlight that, in this context, the comparison involving the Huffman algorithm excludes the use of serialization. This decision stems from the fact that after data serialization into binary format, applying the Huffman algorithm with an extended dictionary becomes challenging.

Figure 2 illustrates the outcomes of experimental investigations involving sequential serialization and data compression using Protobuf in conjunction with the compression algorithm discussed in section number three. Notably, the Huffman algorithm exhibits superior performance for messages with a size of less than 680 bytes. However, for messages exceeding 680 bytes, a different pattern emerges, where the LZ-78 compression algorithm, combined with ProtoBuf serialization, outperforms the Huffman algorithm with an extended dictionary. Therefore, based on the modeling results obtained, it can be inferred that for JSON messages spanning from 30 to 680 bytes in size, the Huffman algorithm with an extended dictionary can achieve data compression ratios ranging from 3.5 times to 8.2 times. For messages falling within the size range of 680 bytes to 2,048 bytes, it's possible to achieve a compression ratio of 3.5 to 4.5 by employing Protobuf in combination with LZ78. Consequently, we attain a minimum guaranteed data compression ratio of 3.5 times when employing both approaches.

B. *Protobuf performance tuning*

This experiment investigates the potential for enhancing the ultimate compression ratio through preliminary data preprocessing before serialization. Experiments were conducted with data sizes below 680 bytes, with a particular emphasis on messages commonly encountered in Iridium packet transmissions, typically measuring 340 bytes in volume. Given that the Huffman algorithm demonstrated superior performance in the preceding experiment for this message type, subsequent experimental studies were exclusively conducted using this compression algorithm. After a thorough examination of data converted to Protobuf format, it became evident that this data representation format exhibits suboptimal performance when handling string-type data, with the maximum compression efficiency achieved when dealing with numeric data.

As a result of this observation, a hypothesis was formulated: by transforming the data in a manner that allows all data, whether textual or numerical, to be represented in digital format, it is possible that when using the Protobuf format, an even higher compression ratio can be attained. Moreover, this approach might enable even greater compression ratios when employing additional compression algorithms, ultimately leading to a substantial reduction in the volume of transmitted messages.

Hence, the proposed method comprises the following stages of data preprocessing:

- In the initial stage, we choose both the original and binary formats for messages.
- In the second stage, we process the originally selected format, replacing string values with numeric equivalents and creating a specific dictionary mapping certain typed strings to corresponding numerical values.

- The third stage involves converting the original format into binary forms using Protobuf
- Finally, in the fourth stage, we compress the messages in binary format.

This approach entails the utilization of a designated database that stores mappings between string data and numeric data, essentially performing operations akin to the Huffman algorithm with an extended dictionary. It's important to emphasize that, in the scenario we are examining, both the data collection system's gateway on the Internet of remote things and the receiving gateway share identical dictionaries for encoding and decoding messages.

The experiment yielded a notable outcome, revealing that by implementing this approach, the message size is halved due to the substitution of textual data with numerical equivalents. After the processed file undergoes serialization and compression, a final compression ratio of up to 27 times the size of the original message can be achieved.

V. Final remarks

In this paper, we have presented a study of the data compression strategies for enhancing IoRT communications over heterogeneous terrestrial-satellite networks. In this study, we conducted a comprehensive exploration of various compression algorithms tailored for tiny IoRT data. We identified the most efficient compression algorithms when combined with ProtoBuf serialization. Furthermore, we introduced a novel data preprocessing method prior to serialization, offering the potential to enhance the overall performance of the data processing pipeline, encompassing serialization and compression.

Based on the obtained performance results, we can highlight the following takeaways:

1) Implementation of serialization and compression for tiny IoRT data improves performance.
2) Based on the modeling results, it can be inferred that for JSON messages ranging from 30 to 680 bytes, the Huffman algorithm with an extended dictionary achieves compression ratios between 3.5 and 8.2 times. For messages between 680 bytes and 2,048 bytes, using Protobuf in combination with LZ78 allows for a compression ratio between 3.5 and 4.5. In conclusion, employing both approaches guarantees a minimum data compression ratio of 3.5 times.
3) The suggested data preprocessing method, involving the replacement of string data with numeric data prior to serialization using ProtoBuf, significantly enhances the overall performance of serialization and data compression, achieving a remarkable improvement of up to 27 times for messages smaller than 680 bytes in size.

The results obtained from this study represent a significant contribution to the field of data serialization and compression in the context of the IoRT. It's worth noting that these findings have the potential to be applied to enhance and refine the method outlined in the referenced article [6], thereby advancing the state of knowledge and improving existing approaches in this domain. In future research endeavors, the intention is to conduct experimental studies and modeling encompassing the entire process of handling data from the Internet of Things, which includes data processing, data serialization, data compression, and the packaging of data into SBD containers. Subsequently, these processed data will be transmitted via satellite channels, allowing for the evaluation of the method's effectiveness in terms of message transmission costs, energy efficiency in message transmission, and communication channel capacity.

Acknowledgement

The publication was prepared within the framework of the Academic Fund Program at HSE University (grant №23-00-035 "Overcoming the limitations of the interaction of cyber-physical systems in heterogeneous networks of the Internet of Remote Things using machine learning methods").

References

[1] O. Kodheli et al., "Satellite Communications in the New Space Era: A Survey and Future Challenges," *IEEE Communications Surveys Tutorials*, vol. 23, no. 1, pp. 70–109, 2021.

[2] M. De Sanctis, E. Cianca, G. Araniti, I. Bisio, and R. Prasad, "Satellite Communications Supporting Internet of Remote Things," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 113–123, 2016.

[3] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, "A Review and State of Art of Internet of Things (IoT)," *Archives of Computational Methods in Engineering*, vol. 29, no. 3, pp. 1395–1413, May 2022.

[4] I. I. Lysogor, L. S. Voskov, A. Y. Rolich, and S. G. Efremov, "Energy efficient method of data transmission in a heterogeneous network of the Internet of things for remote areas," in *2019 International Siberian Conference on Control and Communications (SIBCON)*, 2019, pp. 1–6.

[5] I. Lysogor, L. Voskov, A. Rolich, and S. Efremov, "Study of Data Transfer in a Heterogeneous LoRa-Satellite Network for the Internet of Remote Things," *Sensors*, vol. 19, no. 15, 2019. [Online]. Available: https://www.mdpi.com/1424-8220/19/15/3384.

[6] L. Voskov, A. Rolich, G. Bakanov, and P. Podkopaeva, "Gateway Data Encoding, Packaging and Compression method for heterogeneous IoT-satellite network," in *2021 XVII International Symposium "Problems of Redundancy in Information and Control Systems" (REDUNDANCY)*, 2021, pp. 34–38.

[7] J. D. A. Correa, A. S. R. Pinto, and C. Montez, "Lossy Data Compression for IoT Sensors: A Review," *Internet of Things*, vol. 19, p. 100 516, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2542660522000208.

[8] G. P. Tiwary, E. Stroulia, and A. Srivastava, "Compression of XML and JSON API Responses," *IEEE Access*, vol. 9, pp. 57 426–57 439, 2021.

[9] R. Perez, V. R. Q. Leithardt, and S. D. Correia, "Lossless Compression Scheme for Efficient GNSS Data Transmission on IoT Devices," in *2021 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, 2021, pp. 1–6.

[10] M. R. Chowdhury, S. Tripathi, and S. De, "Adaptive Multivariate Data Compression in Smart Metering Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1287–1297, 2021.

[11] I. Sokol and P. Hubinský, "Internet of things - nonstandard data compression," *Journal of Electrical Engineering*, vol. 71, no. 4, pp. 281–285, 2020. [Online]. Available: https://doi.org/10.2478/jee-2020-0038.

[12] I. Sokol, P. Hubinský, and L'. Chovanec, "Lightweight Cryptography for the Encryption of Data Communication of IoT Devices," *Electronics*, vol. 10, no. 21, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/21/2567.

[13] P. Brogan et al., "Representing Synchrophasor Data Using JSON," in *2021 32nd Irish Signals and Systems Conference (ISSC)*, 2021, pp. 1–6.

[14] J. Azar, A. Makhoul, M. Barhamgi, and R. Couturier, "An energy efficient IoT data compression approach for edge machine learning," *Future Generation Computer Systems*, vol. 96, pp. 168–175, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18331716.

[15] B. R. Stojkoska and Z. Nikolovski, "Data compression for energy efficient IoT solutions," in *2017 25th Telecommunication Forum (TELFOR)*, 2017, pp. 1–4.

[16] I. I. Lysogor, L. S. Voskov, and S. G. Efremov, "Survey of data exchange formats for heterogeneous LPWAN-satellite IoT networks," in *2018 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, 2018, pp. 1–5.