TransNav
http://www.transnav.eu
the International Journal
on Marine Navigation
and Safety of Sea Transportation
Volume 19
Number 4
December 2025

DOI: 10.12716/1001.19.04.16

Optimizing AIS Data Format Based on HELCOM Datasets

K. Šustov & I. Zaitseva-Pärnaste
Tallinn University of Technology, Tallinn, Estonia

ABSTRACT: Automatic Identification System (AIS) data plays a vital role in a wide range of maritime research areas, including logistics optimization, navigational safety analysis, economic activity monitoring, and environmental impact assessment. The HELCOM (Helsinki Commission) organization collects and maintains extensive AIS data for the Baltic Sea region, offering researchers valuable insights into vessel movement and marine traffic patterns. However, the raw AIS data (typically provided in CSV plaintext format) is often large and inefficient to store due to a) plain-text redundancy, b) high levels of duplication and repetitive information. For effective storage and transmission, AIS data is usually compressed as it is, using widely used compression tools (e.g. zip archive). In this study, we investigate techniques for optimizing the storage of HELCOM AIS data by manipulations of data format and structure. Our research reveals that after the undertaken steps, the size of the uncompressed dataset decreased by approx. 60%; the compressed dataset size decreased by approx. 90% compared to the original, revealing the potential for substantial storage savings.
To further improve data handling, we experimented with various structural optimizations of the CSV format, including data arranging by core attributes, column ordering optimization, dataset normalization involving the segregation of mutable and immutable parts. For example, vessel-specific attributes such as ship name, MMSI (Maritime Mobile Service Identity) code, IMO (International Maritime Organization), origin, and dimensions, which stay the same across records for a vessel, can be moved into a separate file during normalization, which significantly reduces the dataset size. The article compares several AIS data persisting strategies to identify the most memory-efficient approaches. Furthermore, we introduce a data generation tool that produces synthetic AIS datasets in customizable formats and patterns. This tool enables reproducibility of the study and supports further experimentation with AIS data optimization approaches.

1 INTRODUCTION

The Automatic Identification System (AIS) has become a foundational technology for maritime traffic monitoring, safety, and navigation. By broadcasting dynamic and static information about vessels—such as position, speed, course, and identity—AIS supports a wide range of applications, from collision avoidance and fleet tracking to environmental monitoring and maritime research. However, the increasing volume of AIS data [1], driven by the growing number of vessels and higher-frequency transmissions, presents significant challenges for storage, processing, and real-time transmission. As maritime analytics evolve into large-scale, data-driven operations, efficient handling of AIS datasets has become a pressing concern, particularly for long-term archiving, real-time processing, and satellite-based AIS systems.

To address these challenges, data compression offers a practical and cost-effective solution. Effective compression techniques can significantly reduce the data footprint while maintaining the integrity and accessibility of the information. However, the choice of compression method must account for multiple factors, including compression ratio, processing speed (both compression and decompression time), and resource consumption, especially in bandwidth-limited or computationally constrained environments.

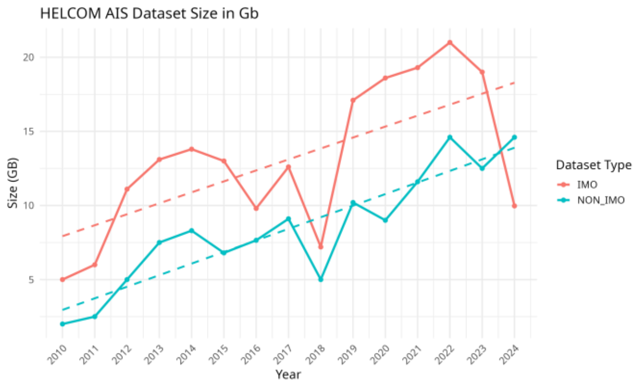

Figure 1. Historical HELCOM AIS dataset size for Gulf of Riga area 2010-2024

Beyond performance and storage efficiency, data compression also has implications for environmental sustainability. For example, an AIS dataset for the Gulf of Riga in 2022 occupies nearly 36 GB in CSV format (see Figure 1) as compared to 1.15 Tb (around 1150 Gb) of data for 6 years period for Arctic region and the Europe, Middle East, North Africa areas [2]. Applying efficient compression reduces the dataset size 12-fold. This stark difference highlights not only the inefficiency of plain text formats, but also the significant potential for reducing digital carbon footprints for data processing, transferring and storing.

According to estimates from Stanford University, storing 100 gigabytes of data in a typical data center produces approximately 0.2 tons of CO_2 emissions annually. More broadly, the carbon footprint of data storage and transmissions is an often-overlooked contributor to greenhouse gas emissions, accounting for about 330 megatons of CO_2 per year, or roughly 2% of global emissions. Though this may seem like a small percentage, it translates to a significant environmental burden when aggregated across the ever-expanding digital ecosystem. [3]

Therefore, compressing AIS data can contribute meaningfully to sustainability goals by lowering the energy demand and emissions associated with both storage and transmission. This impact is particularly relevant when considering not only the storage footprint but also the hidden costs of data movement and access over networks. By optimizing data formats used in HELCOM datasets, we can advance both operational efficiency and environmental responsibility. [3-6]

2 MOTIVATION

The rapid expansion of maritime operations has led to a surge in AIS data, a crucial source for vessel tracking and maritime research. However, storing and processing these vast datasets remains challenging, especially when they are in verbose plaintext formats such as CSV [2].

This inefficiency impacts not just storage infrastructure and computational performance, but also contributes significantly to environmental harm through elevated energy use and data center carbon emissions. Given the growing concerns over digital sustainability, optimizing AIS data formats is no longer just a technical concern—it is an ecological imperative. This study investigates the structural optimization of HELCOM AIS data to enhance storage efficiency and minimize its environmental footprint, achieving reductions of up to 90% in file size after compression.

2.1 *Objective of the research*

The primary objective of this research is to optimize the storage efficiency of HELCOM AIS data by implementing a series of structural and compression-based improvements. These include: (a) enhancing raw data compression efficiency through the use of optimized archival formats; (b) applying structural improvements: columns arrangement, data grouping by core attributes; (c) normalizing the dataset by separating static vessel information into a dedicated file to avoid repeated storage across entries. As a result of these optimizations, the research has demonstrated the following improvements: (a) uncompressed (plain) datasets can be reduced by up to 3-fold; (b) compressed datasets can be reduced by up to 12-fold. Significant improvements in data transfer speeds and processing time, enhancing overall system performance.

2.2 *Automated Identification System (AIS) and Data Structure*

AIS transmits two broad types of information: a) static information: vessel name, MMSI code, IMO number, call sign, flag state, dimensions, and b) dynamic information: accurate position (Lattitude/Longitude), timestamp, SOG (speed over ground), COG (course over ground), etc. Table 1 outlines the structure and core attributes of the raw HELCOM AIS dataset and Table 2 displays a representative snippet of the AIS CSV file utilized in the analysis.

The dataset (Table 3) follows a flat tabular schema. The header row defines the field names corresponding to AIS message attributes, including positional and static vessel information such as timestamp, mmsi, lat, long, sog, cog, shipType, imo, and identifiers like name and callsign, etc. Each subsequent row represents a complete AIS message or tracking point for a vessel, containing time-stamped positional data and redundant static metadata. The dataset includes messages from both IMO-registered and non-IMO vessels.

Despite structural consistency, static fields such as country, name, shipType and ship attributes are repeated across millions of records for each vessel, contributing significantly to storage inefficiency in uncompressed formats. This redundancy highlights the need for format optimization, as explored in this study, particularly in the context of the HELCOM dataset's uncompressed CSV structure.

Table 1. The raw HELCOM AIS dataset attributes

Attribute	Description
timestamp_pretty	Human-readable time format
Timestamp	Unix/epoch timestamp
Msgid	Message type identifier
targetType	Vessel or target classification
Mmsi	Unique vessel ID
lat, long	Geographic coordinates
Posacc	Position accuracy
sog, cog	Speed and course over ground
shipType	IMO classification
dimBow, dimStern, dimPort, dimStarboard	Ship dimensions
Draught	Ship draught
month, week	Redundant time indicators
imo, country, name, callsign	Static identifiers

Table 2. AIS CSV example snippet used in this study

```
"timestamp_pretty";"timestamp";"msgid";"targetType";"mmsi";"lat";"
long";"posacc";"sog";"cog";"shipType";
"dimBow";"draught";"dimPort";"dimStarboard";"dimStern";"month";
"week";"imo";"country";"name";"callsig"

"01/04/2010
06:01:22";1270101682764;1;"A";209322000;57.302517;19.209988;0;17.3;
36.6;"CARGO";
58;8;12;12;94;4;13;9396696;"Cyprus Republic of";"ANNE
SIBUM";"C4YC2"
```

3 METHODOLOGY

The HELCOM AIS dataset has grown significantly in volume due to three converging factors: a) continuous ship tracking on a 24/7 basis, b) increased frequency and precision of AIS message transmissions, c) ongoing expansion of historical archives for long-term maritime analysis d) increased marine traffic density.

Given this rapid growth, storage and processing of AIS data now represent both a technical and environmental challenge. AIS data plays a vital role across multiple maritime domains, including: a) fleet management and operations; b) maritime safety and security; c) port and traffic management; d) maritime research and environmental monitoring. There are also other applications of AIS data, such as: monitoring buoy positions, integration with onboard navigation systems, assessing port coverage, analyzing maritime product performance and emissions, and supporting data fusion applications.

The increasing demand for these applications further emphasizes the importance of scalable, efficient storage formats for AIS data. To address the problem of inefficient storage, we evaluated five formatting schemes for AIS data optimization (Figure 2). These schemes were applied to both the original HELCOM AIS dataset and a synthetically generated AIS dataset to ensure reproducibility and broader applicability. The results of each optimization step are visually summarized in Figure 2, which outlines the transformation of AIS data across all schemes.

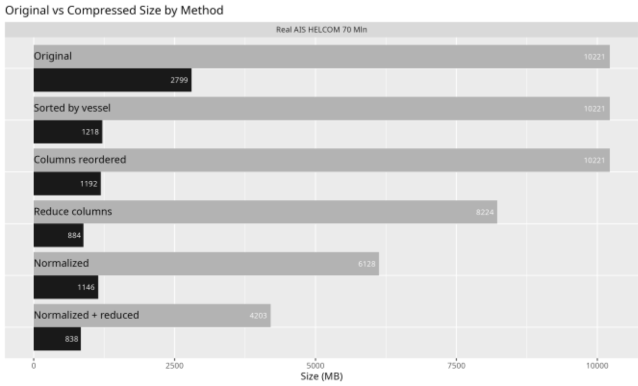

Figure 2. Transformation of AIS data across all schemes (methods)

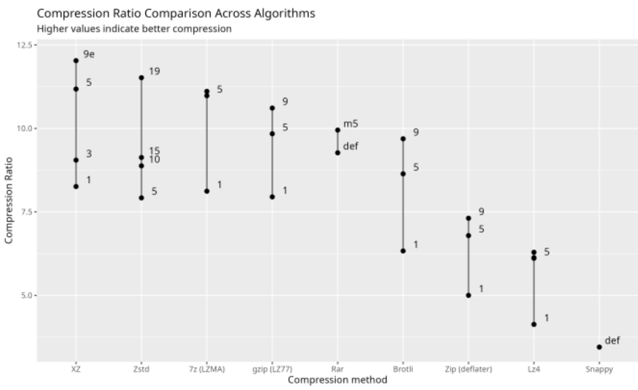

Figure 3. Comparison of compression methods

Schemes 1 and 2 retain the original HELCOM CSV format structure, serving as baselines for comparison. Although various compression methods were initially tested (Figure 3), we selected the Zstandard (Zstd) compressor for all experiments due to its superior balance of compression ratio and speed. [7-12]

The use of Zstd contributed significantly to the 12.2x improvement in compressed file size achieved. This scheme retains the original structure of the HELCOM AIS data, with no modifications or optimizations applied (Table 5). It serves as the baseline reference (method 1) for evaluating subsequent improvements (methods 2-6).

Despite its lack of structural refinement, even basic compression using the Zstandard (Zstd) algorithm reduces the dataset size by approximately 3.65x, demonstrating that compression alone can yield significant storage savings, even without further optimization (Figure 2).

Table 3. AIS Data structure method 1, 2, 3

Method 1 and 2: Original AIS Helcom format		Method 3: Helcom format with reordered tokens	
Token	**Sample**	**Token**	**Sample**
timestamp_pretty	01/07/2019 01:41:59	timestamp_pretty	01/07/2019 01:28:26
timestamp	1561945319000	timestamp	1561944506000
msgid	18	lat	58.483913
targetType	B	long	19.283783
mmsi	203049100	sog	8.4
lat	58.451134	cog	154.2
long	19.302147	posacc	1
posacc	1	month	7
sog	8	week	27
cog	163.1	msgid	18
shipType	SAILING	targetType	B
dimBow	7	mmsi	203049100
draught	NA	shipType	SAILING
dimPort	3	dimBow	7
dimStarboard	1	draught	NA
dimStern	4	dimPort	3
month	7	dimStarboard	1
week	27	dimStern	4
imo	NA	imo	NA
country	Austria	country	Austria
name	BREVA	name	BREVA
callsign	OEX5080	callsign	OEX5080

The next step was sorting data by vessels with saving the original structure of the data. In this step, the dataset columns were reordered to place static metadata (e.g., ship type, country, IMO number) at the beginning of each record. This reordering was intended to leverage Zstd's ability to compress repeated values more efficiently. Improvements are illustrated in Figure 2, where compressed size became 1218 MB from the original which is 2.4x times better than method 1. Method 3 was to reorder columns (Table 6) and method 4 - remove duplicate columns (timestamp_pretty, week, month); see Table 4.

Method 5 was to split static and dynamic data. Building on the previous scheme, static vessel information was extracted from the AIS messages and stored in a separate registry file, reducing redundancy across records. The resulting data size reduces to 61% of original size, and after compression takes only 1146MB. This structural optimization is visualized in Table 5.

Table 4. Dataset with removed duplicate columns

Method 4: Removed dublicate columns	
Token	**Sample**
timestamp	1561944506000
lat	58.483913
long	19.283783
sog	8.4
cog	154.2
posacc	1
msgid	18
targetType	B
mmsi	203049100
shipType	SAILING
dimBow	7
draught	NA
dimPort	3
dimStarboard	1
dimStern	4
imo	NA
country	Austria
name	BREVA
callsign	OEX5080

Table 5. Method 5 - AIS dataset is normalized (separation static vessel data)

AIS data	
Token	**Sample**
timestamp_pretty	01/07/2019 01:01:56
timestamp	1561942916000
lat	58.546993
long	19.24858
sog	7.9
cog	175.6
msgid	18
month	7
week	27
posacc	1
targetType	B
mmsi	203049100

Vessel data	
Token	**Sample**
mmsi	637621745
type	HSC
dimBow	80.6
draught	23.1
dimPort	18.8
dimStarboard	2.5
dimStern	126.2
imo	2224309
country	Congo (Congo-Brazzaville)
name	DALAROE
callsign	UCKU

Table 6. Method 6. AIS dataset is normalized and data reduced

AIS data	
Token	**Sample**
timestamp	1561942076000
lat	58.580536
long	19.233147
sog	9.2
cog	161.3
posacc	1
msgid	18
targetType	B
mmsi	203049100

Vessel data	
Token	**Sample**
mmsi	637621745
type	HSC
dimBow	80.6
draught	23.1
dimPort	18.8
dimStarboard	2.5
dimStern	126.2
imo	2224309
country	Congo (Congo-Brazzaville)
name	DALAROE
callsign	UCKU

The final step (data cleanup and normalization) introduced additional improvements on top of schemes 4 and 5: removed non-essential and redundant columns (timestamp_pretty, month, and week) and further normalized the data structure by separating static vessel metadata entirely from the dynamic AIS position reports (Table 6). In this step the raw data size became 4.2 GB, and after compression: 838MB.

The full impact of these final optimizations is shown in Figure 2 and Table 7.

Table 7. Comparison of applied method for original AIS Helcom raw data and AIS generated data

Dataset Compression Comparison

Original vs. Zstd-compressed sizes across different preprocessing methods

	Original	Compressed	Abs.Ratio[1]	Rel.Ratio[2]	Plain Size %	Comp. vs Orig[3]	Efficiency %[4]
Generated 70M							
Original	10,690 Mb	1,549 Mb	6.90	6.90	100.00%	14.49%	100.00%
Sorted by vessel	10,690 Mb	1,271 Mb	8.41	8.41	100.00%	11.89%	82.05%
Columns reordered	10,690 Mb	1,261 Mb	8.48	8.48	100.00%	11.80%	81.41%
Reduce columns	8,922 Mb	1,106 Mb	8.07	9.67	83.46%	10.35%	71.40%
Normalized	5,842 Mb	1,173 Mb	4.98	9.11	54.65%	10.97%	75.73%
Normalized + reduced	4,081 Mb	1,022 Mb	3.99	10.46	38.18%	9.56%	65.98%
Real AIS HELCOM 70M							
Original	10,221 Mb	2,799 Mb	3.65	3.65	100.00%	27.38%	100.00%
Sorted by vessel	10,221 Mb	1,218 Mb	8.39	8.39	100.00%	11.92%	43.52%
Columns reordered	10,221 Mb	1,192 Mb	8.57	8.57	100.00%	11.66%	42.59%
Reduce columns	8,224 Mb	884 Mb	9.30	11.56	80.46%	8.65%	31.58%
Normalized	6,128 Mb	1,146 Mb	5.35	8.92	59.95%	11.21%	40.94%
Normalized + reduced	4,203 Mb	838 Mb	5.02	12.20	41.12%	8.20%	29.94%
Ultra compression mode							
Sorted by vessel	10,221 Mb	824 Mb	12.40	12.40[5]	100.00%	8.06%	29.44%
Normalized and reduced	4,203 Mb	560 Mb	7.51	18.25[6]	41.12%	5.48%	20.01%

[1] Abs.Ratio = Plain / Compressed Size

[2] Rel.Ratio = Original Size / Compressed Size

[3] Comp. vs Orig = Compressed size / Plain original

[4] Efficiency % = (Method's Compressed Size / Original's Compressed Size) × 100

[5] Compression time: 12005.25s (13:42min on 16 cores)

[6] Compression time: 5278.93s (6:15min on 15 cores)

4 RESULTS AND DISCUSSION

The applied optimization methods show consistent and significant improvements across both the real HELCOM AIS dataset and the synthetically generated data. As presented in Table 7, each successive method improved the data size reduction, compression ratio, and storage efficiency.

For the generated 70M AIS dataset, the original raw size was 10.69 GB, compressing to 1.55 GB using Zstd. After applying all optimization step: reordering, columns reduction, and normalization - the final compressed size was reduced to 1.02 GB, with a relative compression ratio of 10.46 times and a total space saving of over 90% of storage; reducing the plain dataset size by 38% of the original size without compression.

In the case of the real HELCOM AIS dataset (70M rows), the unoptimized data compressed to 2.8 GB. The most effective method—normalization combined with column reduction—reduced the raw dataset from 10.2 GB to 4.2 GB, and the compressed size to 838 MB, achieving a relative compression ratio of 12.20. This corresponds to an 8.2% compressed size compared to the original dataset, meaning over 91% space saving.

Further, in an ultra-compression mode, where maximum compression settings were used, the compressed size dropped even further to 560 MB, achieving a compression ratio of 18.25x, albeit with significantly longer compression times (CPU time over 3 hours).

These results show that: a) structural data cleanup and normalization lead to 2–3x better compression than compression alone; b) reordering and deduplicating static fields is a high-impact, low-effort optimization; c) the Zstandard (Zstd) compressor consistently outperforms other methods in both compression ratio and efficiency; d) the methodology is effective across both real and synthetic AIS datasets, confirming its generalizability.

In terms of efficiency percentage (calculated as the compression performance relative to the original method), the most optimized method achieved over 65% storage efficiency improvement on the generated dataset and about 30% improvement on the real dataset.

The study revealed that applying method 2, which combines the corresponding compression method over sorted datasets without any schema modification, reduced storage usage by 12 times. For instance, the historical AIS dataset for the Gulf of Riga decreased from 323 GB to 26 GB. However, applying more advanced techniques, such as schema modification and denormalization, enabled reductions of up to 18 times. These approaches, however, require additional data pre- and post-processing steps, as well as modifications to existing IT solutions.

5 CONCLUSION

Optimizing AIS data formats significantly enhances both storage efficiency and ecological sustainability. By applying structural reorganizations—grouping, column reordering, and deduplication—storage savings of up to 66% were achieved. This research demonstrates the potential of data-aware format engineering over traditional compression methods alone. The development of a synthetic data generator further enables reproducibility and broader experimentation.

Specialized algorithms, like adaptive Douglas-Peucker variants and top-down kinematic compression, exploit spatial and kinematic attributes (position, speed, course) to achieve higher compression ratios while maintaining critical trajectory features. Combining these specialized methods with general archivers as a second compression stage can maximize storage savings. [13]

This composite view highlights that structural optimization, dedicated trajectory compression, and general archiving together offer the best approach for efficient AIS data storage and sustained ecological benefits.

These results are not just relevant for maritime analysts, but also for data engineers, archivists, and policymakers aiming for sustainable data practices in marine science and beyond.

ACKNOWLEDGMENTS

The authors gratefully acknowledge HELCOM (Helsinki Commission) for providing access to the AIS datasets used in this study.

BIBLIOGRAPHY

[1] Clissa, L. (2022). Survey of big data sizes in 2021. arXiv. https://doi.org/10.48550/arXiv.2202.07659
[2] Corvino, M., Daffinà, F., Francalanci, C., Giacomazzi, P., Magliani, M., Ravanelli, P., & Stahl, T. (2025). A Methodology to extract Geo-Referenced Standard Routes from AIS Data [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2503.22734
[3] Monserrate, S. G. (2022). The Cloud Is Material: On the Environmental Impacts of Computation and Data Storage. MIT Case Studies in Social and Ethical Responsibilities of Computing, Winter 2022. https://doi.org/10.21428/2c646de5.031d4553
[4] Safdie, S. (2024). What is the carbon footprint of data storage? Greenly. https://greenly.earth/en-gb/blog/industries/what-is-the-carbon-footprint-of-data-storage
[5] Aujoux, C., Kotera, K., & Blanchard, O. (2021). Estimating the carbon footprint of the GRAND project, a multi-decade astrophysics experiment. Astroparticle Physics, 131, 102587. https://doi.org/10.1016/j.astropartphys.2021.102587
[6] Istrate, R., Tulus, V., Grass, R. N., Vanbever, L., Stark, W. J., & Guillén-Gosálbez, G. (2024). The environmental sustainability of digital content consumption. Nature Communications, 15(1), 3981. https://doi.org/10.1038/s41467-024-47621-w
[7] Stecuła, B., Stecuła, K., & Kapczyński, A. (2022). Compression of text in selected languages—Efficiency, volume, and time comparison. Sensors, 22(17), 6393. https://doi.org/10.3390/s22176393
[8] Sobczyński, S. (2025, May 25). zstd vs zip vs 7-Zip (LZMA2): .NET compression benchmark. hasto.pl. https://hasto.pl/compression-benchmark-zip-vs-7-zip-lzma2-vs-zstandard

[9] Shadura, O., Bockelman, B. P., Canal, P., Piparo, D., & Zhang, Z. (2020). ROOT I/O compression improvements for HEP analysis. EPJ Web of Conferences, 245, 02017.
[10] Marcon, C., Mete, A. S., Van Gemmeren, P., & Carminati, L. (2024). Optimizing ATLAS data storage: The impact of compression algorithms on ATLAS physics analysis data formats. EPJ Web of Conferences, 295, 03027. https://doi.org/10.1051/epjconf/202429503027
[11] Mao, Y., Cui, Y., Kuo, T., & Xue, C. J. (2022). A fast transformer-based General-Purpose lossless compressor. arXiv.org. https://arxiv.org/abs/2203.16114
[12] Gastegger, M. (2020, June). A performance comparison of 7z/LZMA and 7z/bzip2/tar [Unpublished manuscript, TU Wien]. ResearchGate. Retrieved from https://www.researchgate.net/publication/350049637_A_Performance_Comparison_of_7zLZMA_and_7zbzip2tar
[13] Zhang, T., Wang, Z., & Wang, P. (2024). A method for compressing AIS trajectory based on the adaptive core threshold difference Douglas–Peucker algorithm. Scientific Reports, 14(1). https://doi.org/10.1038/s41598-024-71779-4