Check for updates

RESEARCH ARTICLE

REVISED A Multiple Compression Approach using Attribute-based Signatures

[version 2; peer review: 2 approved]

Constantinos Costa[1], Panos Chrysanthis[1], Herodotos Herodotou[2], Marios Costa[1], Efstathios Stavrakis[3], Nicolas Nicolaou[3]

[1]Rinnoco Ltd, Limassol, 3047, Cyprus
[2]Electrical Engineering and Computer Engineering and Informatics, Cyprus University of Technology, Limassol, 3036, Cyprus
[3]Algolysis Ltd, Limassol, 4630, Cyprus

v2 **First published:** 10 Feb 2025, **5**:49
https://doi.org/10.12688/openreseurope.19247.1
Latest published: 08 Sep 2026, **5**:49
https://doi.org/10.12688/openreseurope.19247.2

Abstract

Background

With the increasing volume of data collected for advanced analytical and AI applications, data storage remains a significant challenge. Despite advancements in storage technologies, the cost of maintaining vast datasets continues to grow. Compression techniques have been widely used to address this issue, but existing systems primarily rely on a single, typically lossless method, which limits adaptability to varying data characteristics.

Methods

This paper introduces COMPASS, a multiple compression approach that applies different compression techniques to distinct subsets of data within a database. COMPASS partitions relational data into rows or columns and selects the most suitable compression scheme for each column or column group. Two versions of COMPASS are proposed: (i) COMPASS-D, which utilizes K-Means clustering based on data values; and (ii) COMPASS-E, which employs K-Means clustering based on column entropy to group similar columns efficiently. The effectiveness of COMPASS is evaluated using the Envmon dataset, a real-world environmental monitoring database, and compared against monolithic compression methods.

Results

Open Peer Review

Approval Status ✓ ✓

	1	2
version 2 (revision) 08 Sep 2026	✓ view	✓ view
version 1 10 Feb 2025	? view	? view

1. **Alexandra Theodoropoulou**, University of Patras, Patras, Greece

2. **Martin Burtscher**, Texas State University, San Marcos, USA

Any reports and responses or comments on the article can be found at the end of the article.

Experimental results demonstrate that COMPASS significantly reduces disk space usage compared to traditional compression techniques. COMPASS-E achieves superior performance in terms of compression time and proximity to the optimal compression ratio, outperforming COMPASS-D. In worst-case scenarios, COMPASS methods offer 22% more savings compared to baseline techniques, with best-case savings reaching 56% (~2× improvement).

Conclusion

The proposed COMPASS framework offers a flexible and adaptive approach to database compression by leveraging multiple schemes tailored to different data subsets. This results in improved storage efficiency and reduced computational overhead. Future work will explore additional data characteristics and clustering methods to further enhance COMPASS's adaptability and efficiency.

Keywords
big data, signature based, compression, column stores, hybrid store



This article is included in the Horizon Europe gateway.

Corresponding author: Constantinos Costa (costa.c@rinnoco.com)

Author roles: Costa C: Conceptualization, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Chrysanthis P**: Conceptualization, Formal Analysis, Resources, Software, Supervision, Validation, Writing – Original Draft Preparation, Writing – Review & Editing; **Herodotou H**: Data Curation, Methodology, Resources, Software, Supervision, Validation, Writing – Review & Editing; **Costa M**: Conceptualization, Supervision, Validation, Writing – Review & Editing; **Stavrakis E**: Investigation, Validation, Writing – Review & Editing; **Nicolaou N**: Investigation, Validation, Writing – Review & Editing

Competing interests: No competing interests were disclosed.

Grant information: This work is implemented under the programme of social cohesion "THALIA 2021-2027" co-funded by the European Union, through Research and Innovation Foundation, under project COMPASS - CONCEPT/0823/0002, and is also partially supported by the European Union's Horizon Europe program for Research and Innovation through the HYPER-AI project under Grant No. 101135982. The views, findings, conclusions, or recommendations expressed in this material are solely those of the author(s) and do not necessarily represent those of the sponsors.
The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Copyright: © 2026 Costa C *et al*. This is an open access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

How to cite this article: Costa C, Chrysanthis P, Herodotou H *et al.* **A Multiple Compression Approach using Attribute-based Signatures [version 2; peer review: 2 approved]** Open Research Europe 2026, **5**:49 https://doi.org/10.12688/openreseurope.19247.2

First published: 10 Feb 2025, **5**:49 https://doi.org/10.12688/openreseurope.19247.1

REVISED **Amendments from Version 1**

Version 2 includes revisions made in response to reviewer feedback. The main changes are: (i) addition of runtime and resource-usage results, along with clearer reporting of the experimental environment; (ii) clarification of methodological choices, including the clustering rationale, silhouette-score interpretation, and reproducibility details; (iii) expansion of the related work to include additional studies on heterogeneous and automated compression selection; and (iv) a strengthened conclusion discussing limitations and future work. Minor typographical, formatting, and figure-quality improvements were also made throughout the manuscript.

Any further responses from the reviewers can be found at the end of the article

1. Introduction

In the current era of AI that transforms the way we live and work, data management and processing techniques are continuously challenged as AI methods' effectiveness depends on the availability of vast amounts of data.[1] Despite the annual doubling of electronically stored data volume, the cost of storage capacity decreases at a rate of less than one-fifth per year.[2] For this reason, both lossless[3] and lossy[4] compression techniques have been developed and optimized to reduce data storage for specific applications as well as for general use.

The current approach to database compression, which we refer to as *monolithic*, utilizes a single, typically lossless method to reduce the disk space of a database. These systems can result in significant savings in storage costs, but cannot adapt to data distribution changes. Thus, they cannot select the best encoding/compression scheme for a given dataset.[5,6]

This paper proposes a *multiple compression* approach, dubbed *COMPASS*, that uses different compression techniques for distinct data subsets in a database. It is anchored on the hypothesis of SIBACO that multi-scheme data compression is more effective for complex big data.[7] Specifically, COMPASS breaks down relational data into rows or columns and applies the most suitable compression scheme to individual columns or groups of columns.

The paper presents and evaluates two versions of COMPASS: COMPASS-D, which exploits K-Means clustering on data, and COMPASS-E, which exploits K-Means on column entropy to group similar columns. Finally, the best compression scheme is selected for each column group. The experimental results using the Envmon database, a real dataset from the Environmental Monitoring Platform [1], show that COMPASS significantly reduces disk space usage compared to monolithic methods, where COMPASS-E outperforms COMPASS-D in terms of compression time as well as proximity to the optimal compression ratio.

The remainder of the paper is structured as follows: Section 2 reviews the current state-of-the-art in compression techniques. Section 3 presents the methodology of COMPASS, and Section 4 presents our experimental results. Section 5 concludes our paper by discussing the next steps of our work.

2. Related Work

The realm of big data storage has seen various advancements in compression techniques aimed at reducing storage expenses while preserving efficient data retrieval. We focus on *lossless* compression techniques in this work as the majority of data-intensive applications need to be able to support *exact* queries over stored data.

Lossless data compression can be categorized into four main types. *Dictionary-based compression* uses a dictionary to capture frequently appearing values within the data, replacing them with a shorter index code (examples include LZ77 and LZ78). *Statistical compression* relies on statistical models to estimate the frequency of data values (such as Huffman coding and LZMA). *Transform-based compression* applies mathematical transformations to condense the data into a more compressed format (like Burrows-Wheeler Transform and BZIP2). Finally, *Hybrid compression* merges techniques from the aforementioned categories to enhance compression efficiency (for instance, DEFLATE, which combines LZ77 and Huffman coding). Previous studies have investigated the automated selection of compression algorithms for heterogeneous datasets, including synthesis-based approaches[8] and evolutionary strategies.[9] In addition, simple machine-learning methods have been proposed for selecting the most appropriate compression algorithm based on extracted features/characteristics (e.g., entropy, data distribution), following a principle similar to SIBACO.[10]

[1]Envmon Database was produced by the STEAM: https://steam.cut.ac.cy

Big companies have developed specialized lossless compression techniques tailored to their specific application needs. Google's Snappy [2] for example, performs compression through byte-level operations and bit-stream encoding, whereas Facebook's Zstandard [3] employs dictionary-based methods designed for real-time data compression. Additionally, LZ4 [4] is recognized for its speed, employing a byte-oriented variant of LZ77. In our experiments, we utilized several well-known compression algorithms from the *zipfile* library, including LZMA, BZIP2, and DEFLATE, representing the three compression types mentioned previously. However, the framework itself is not limited to these compression schemes and can be extended to incorporate additional modern compressors such as Zstandard or LZ4.

The research topic of database compression has been under exploration for more than three decades. To meet the substantial demands of OLAP systems processing big data, columnar storage formats have been developed that incorporate compression techniques to reduce storage costs.[11–13] Recent advancements in hardware, such as CPUs, RAM, and storage devices, have enabled new optimizations that minimize memory accesses by leveraging compression technologies.[14,15] Building a compression-aware database management system can reduce storage costs and improve query response times by increasing compression ratios.[16] Additionally, cutting-edge approaches are employing machine learning to manage and store vast amounts of data more efficiently.[17–20]

Moreover, leveraging data organization and data types has proven effective in enhancing query speeds and reducing storage requirements through compression.[5,6,21] While applying a uniform compression method across different data partitions can increase compression effectiveness,[22] we suggest employing a variety of compression strategies to achieve superior results. Although our approach is closer to a black-box technique, it aligns with principles of white-box compression by making the compression mechanisms visible to applications via database metadata.[23]

The evolving landscape of AI models necessitates continuous storage of large-scale data to iteratively improve model accuracy and precision. Storing data indefinitely is a problem that traditional methods (e.g., compression) address by exploiting the computational resources. For instance, while lossless and lossy compression techniques exist, they fall short in supporting big data analytics.[24] Data reduction methods such as sampling,[25] aggregation for OLAP,[26] dimensionality reduction techniques like LDA and PCA,[27] and synopsis/sketches[28] provide critical insights by simplifying the complexity of large datasets. Another way to deal with ever-increasing huge amounts of data is to utilize the concept of *data rotting* or *data amnesia*[29,30] to progressively remove detail from stored information as the data ages, thereby decreasing storage costs.

Recent advancements in data compression and querying focus cloud and big data systems, such as BtrBlocks,[24] optimizing decompression and compression ratios for data lakes, enhancing cloud interoperability. Gorilla[31] effectively manages vast amounts of data through float data compression techniques, reducing storage demands significantly. Additionally, Decomposed Bounded Floats[32] handles low-precision float data using a decomposed columnar storage format. Moreover, Chimp[33] enhances floating-point time series data compression, significantly improving compression ratios and access times. These developments underscore the evolving landscape in data compression and querying, providing a crucial context for our COMPASS project.

3. Methods

The underlying hypothesis of COMPASS is that employing multiple data compression schemes can be more effective for complex big data, as it enables the selection of different compression methods for groups of data with similar characteristics. These groups can then be decompressed individually to meet the requirements of specific queries, without decompressing the entire dataset. This approach utilizes several compression schemes, each tailored to optimize the compression of different data subsets according to their unique characteristics.

COMPASS breaks down relational data into columns and applies the most suitable compression scheme to each column or group of columns. COMPASS uses K-Means clustering to group *similar* columns together, based on their data values or entropy, and then applies the best compression technique to each individual cluster. COMPASS determines the compression scheme for a single column or a group of columns by exhaustively testing all possible combinations.

Next, we elaborate on COMPASS' two phases, illustrated in Figure 1: (i) partitioning and grouping compatible columns; and (ii) selection of best performing compression.

[2]Snappy: https://google.github.io/snappy/
[3]Zstandard: https://facebook.github.io/zstd/
[4]LZ4: https://lz4.github.io/lz4/

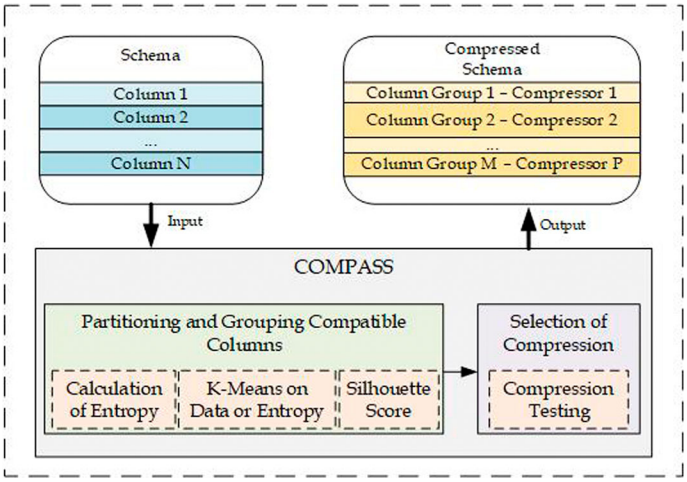

Figure 1. COMPASS operates in two phases: (i) partitioning and grouping; and (ii) choosing the most effective compression method.

3.1 Segmentation and clustering of columns

In the first stage, COMPASS partitions each table in the database into a set of columns and utilizes K-Means as the primary method to create groups of similar columns. We apply K-Means in two ways, leading to two different versions of COMPASS: *COMPASS-D*, which utilizes K-Means clustering on data values, and *COMPASS-E*, which exploits K-Means clustering on column entropy to group similar columns. The latter is very fast as it reduces dimensionality through entropy. Specifically, *COMPASS-E* uses Shannon's entropy[34] to identify the compressibility of the columns.

To verify and evaluate our proposed approach, we use the *silhouette coefficient*[27] to select the number of clusters for the K-Means algorithm. Particularly, the silhouette coefficient measures the quality of the clusters based on the distance within each cluster and the average distance to the nearest cluster for each data point. The silhouette coefficient for a data point is calculated using the following:

$$S_i = \frac{y_i - x_i}{max\{x_i, y_i\}}$$

where:

- x_i is the average distance between the data point i and the rest of the points within the cluster.
- y_i is the minimum average distance from the data point i to points in a different cluster.

3.2 Selection of best performing compression

In the second stage, COMPASS exhaustively searches for the most suitable compression scheme for the combinations of clusters with low silhouette scores and all compression schemes (e.g., DEFLATED, BZIP2, LZMA). While the silhouette coefficient is typically maximized in clustering tasks, our objective differs. Lower silhouette values indicate clusters that are less strictly separated, which in our context may allow columns with similar compression characteristics to be grouped together even if their statistical profiles differ. Empirically, we observed that such groupings can yield higher compression ratios because the selected compression scheme can exploit shared structural patterns across columns.

4. Results & Discussion

This section discusses the results and provides details on the experimental setup, including datasets and techniques used for the experimental evaluation.

4.1 Experimental methodology

In our experimentation, we chose the same readily available compression algorithms (i.e., LZMA, BZIP2, and DEFLATE) from the *zipfile* library, used in SIBACO.

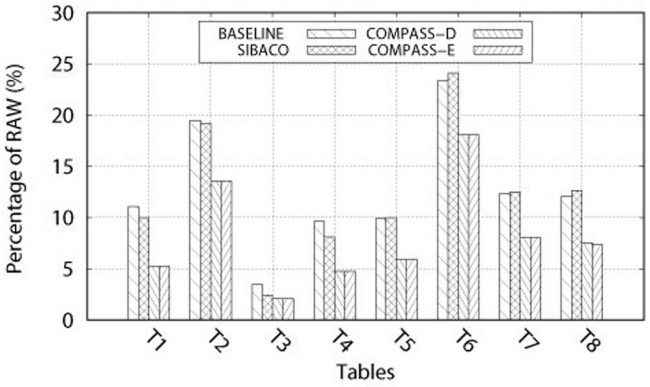

Figure 2. Disk Space for the eight largest tables in the Envmon Database.

We utilized K-Means to group the columns, after scaling the input data with StandardScaler, and encoding all string columns with the OrdinalEncoder from the *sklearn 1.4.2* library. To ensure reproducibility, the K-means clustering algorithm was initialized with a fixed random seed (seed = 42).

Compared Techniques: Our goal in the first experimental series is the comparison of the following four techniques:

BASELINE: This baseline technique compresses the data without considering the data characteristics, using the best compression scheme from the *zipfile* library for each table.

SIBACO: This is our previous technique that employs multiple compression schemes.[7] SIBACO splits the columns into only two groups based on their entropy and applies the best compression scheme to each column group.

COMPASS-D: This is our first proposed technique, which uses K-Means clustering to group columns based on data values to achieve the best possible compression ratio using multiple schemes if needed.

COMPASS-E: This is our second proposed technique, which applies K-Means clustering based on column entropy and has significantly lower computational complexity than COMPASS-D.

4.2 Experimental testbed

To validate our proposed ideas and evaluate COMPASS, we conduct the following experiments on two Ubuntu 22.04 servers, each featuring 24GB of RAM and an Intel(R) Xeon(R) E5-2630 CPU.

Envmon Dataset: This is a real dataset collected from the Environmental Monitoring Platform, developed under the STEAM project (Sea Traffic Management in the Eastern Mediterranean) [5]. The database contains primarily environmental, meteorological, and oceanographic data. The dataset was collected over the course of three years and has a total size of ∼1GB.

4.3 Experimental results

Across all tables, COMPASS-D and COMPASS-E consistently achieve the most efficient disk space reduction, resulting in 2–18.2% of the original RAW size. In contrast, the BASELINE and SIBACO methods reduce the disk space requirements by 4.6–23.4% and 4–23.6% of the original RAW size, respectively. COMPASS-D and COMPASS-E outperform BASELINE and SIBACO by more than 22% in the worst case and 56% (i.e., ∼2×) in the best case, as shown in Figure 2.

4.4 Selecting the number of clusters

In this experimental series, we examine the silhouette coefficient as we vary the number of clusters to verify that selecting a low-silhouette-score number of clusters can yield good compression performance.

[5]STEAM: https://steam.cut.ac.cy

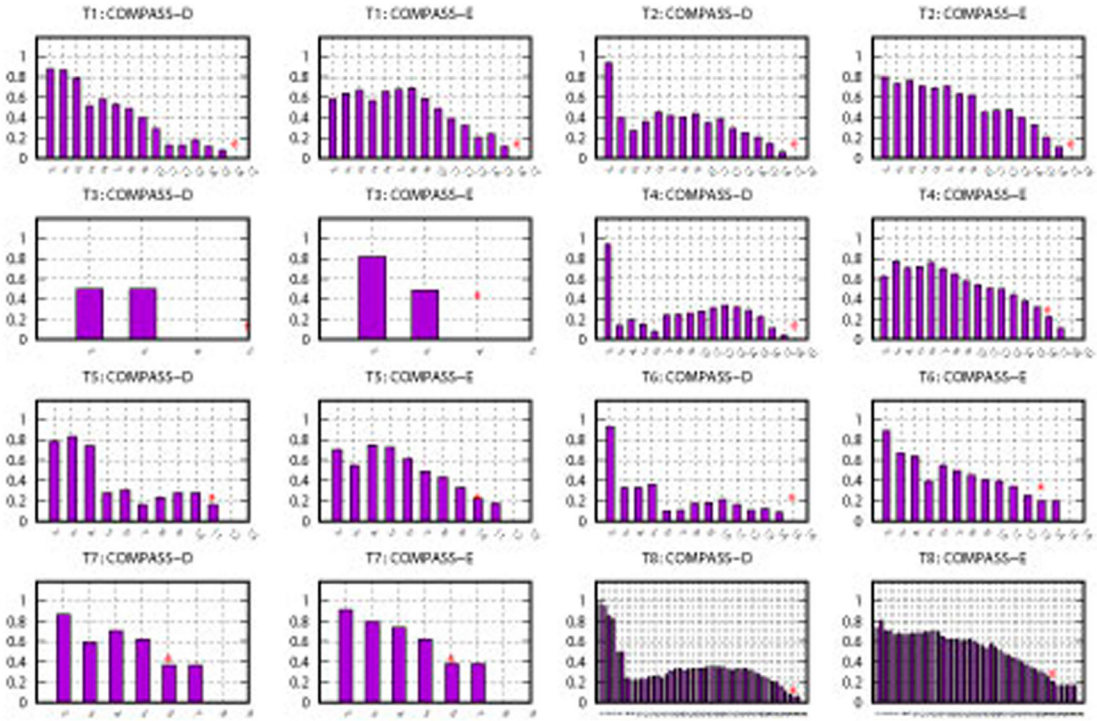

Figure 3. Average Silhouette Score for the eight largest tables in the Envmon Database.

Particularly, Figure 3 shows silhouette coefficient scores for the eight largest tables in the Envmon Database for all possible numbers of clusters. The asterisk on top of a bar indicates the number of clusters that yields the best compression ratio after the second stage of COMPASS (i.e., Selection of Compression). The clustering with the lowest silhouette coefficient often coincides or is very close to the one with the star, showing that the proposed approach is a very good proxy for finding the (near) optimal compression clustering. It is important to note that the silhouette coefficient is only defined if the number of clusters is between two and N-1, where N is the total number of columns (i.e., the maximum number of clusters).

The asterisk on top of a bar shows which number of clusters was selected with the best disk space efficiency (X-axis: Number of clusters, Y-axis: Average Silhouette Score).

To better understand the results, we calculate the relative difference between the disk storage of the compressed data for the number of clusters based on the lowest silhouette score and the COMPASS Compression Testing (i.e., the best performing one), using the following formula:

$$\text{Relative Difference} = \frac{|DS - DC|}{\frac{DS+DC}{2}} \times 100\%$$

where:

- D_S is the disk space for the number of clusters based on the lowest silhouette score.
- D_C is the disk space for the number of clusters based on COMPASS testing.

The maximum relative difference for COMPASS-D is ∼8%, with an average of ∼1%. COMPASS-E exhibits less than ∼1% maximum relative difference and ∼0.1% on average, revealing that COMPASS-E's results are closer to the optimal compression ratios compared to COMPASS-D.

Table 1. Compression and decompression runtime of the evaluated methods on the Envmon dataset.

Method	Compression Time (s)	Decompression Time (s)	CPU (%)	RAM (MB)
RAW	0.000	0.000	0.000	0.000
Baseline (single scheme)	36.082	0.726	100.805	438.143
COMPASS-D	17.150	0.922	100.640	451.231
COMPASS-E	16.311	0.846	100.645	437.548

4.5 Compression and Decompression Time

In addition to the compression ratio, we evaluated the runtime overhead of COMPASS. Table 1 reports compression and decompression times, along with CPU and RAM usage, measured on a machine with 40GB of RAM and a 12th Gen Intel(R) Core i9-12900H processor. Although COMPASS introduces a clustering step, the additional cost remains modest relative to the overall compression process.

5. Conclusion

This paper presents a novel multiple compression approach, named COMPASS, that uses different compression techniques for distinct data subsets in a database. We introduce and evaluate two versions of COMPASS, namely COMPASS-D and COMPASS-E, designed to enhance the compression of relational data. COMPASS-D leverages K-Means clustering on the data values, while COMPASS-E uses column entropy to achieve more efficient compression ratios in relational databases. In our experimental evaluation, we observe that COMPASS offers substantial reductions in disk space utilization compared to traditional monolithic methods and our previous work, SIBACO. Specifically, COMPASS-D and COMPASS-E outperform the BASELINE and SIBACO techniques in terms of disk storage savings by more than 22% in the worst case and 56% (i.e., $\sim 2\times$) in the best case.

While the results demonstrate the effectiveness of COMPASS on the Envmon dataset, the current evaluation focuses on a single real-world dataset. Although this dataset represents a realistic environmental monitoring workload, additional experiments on datasets with different schema structures and data characteristics would further validate the generality of the approach. Future work will therefore evaluate COMPASS across a broader range of relational datasets and explore integrating additional compression schemes.

Furthermore, in addition to entropy and the silhouette coefficient, we plan to explore other data characteristics and indicators to enhance the speed and efficiency of COMPASS. Additionally, we aim to use these metrics to construct a comprehensive knowledge base that provides deeper insights into attribute-based compression signatures. Lastly, query-time evaluation is an important direction for future work, but it was outside the scope of the current study.

Ethics and Consent

Ethical approval and consent were not required.

Data Availability

The data supporting the findings of this study have been deposited in Zenodo and are publicly available at https://doi.org/10.5281/zenodo.14751777.[35] Data are available under the terms of the Creative Commons Attribution 4.0 International license (CC-BY 4.0).

Software Availability

- Source code available from: https://github.com/Rinnoco/compass_com
- Archived software available from: https://doi.org/10.5281/zenodo.14759015
- License: Apache 2.0

The software used in this study has been archived in Zenodo and is publicly available at https://doi.org/10.5281/zenodo.14759015. It is released under the Apache 2.0 license.[36]

Acknowledgments

This work is implemented under the programme of social cohesion "THALIA 2021-2027" co-funded by the European Union, through Research and Innovation Foundation, and is also partially supported by the European Union's Horizon Europe program for Research and Innovation through the HYPER-AI project under Grant No. 101135982. The views, findings, conclusions, or recommendations expressed in this material are solely those of the author(s) and do not necessarily represent those of the sponsors. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

References

1. Raman V, Swart G: **How to wring a table dry: entropy compression of relations and querying of compressed relations**. In: *Proceedings of the 32nd International Conference on Very Large Data Bases.* 2006; 858–869.

2. Data Center Journal: **The cost of data storage and management: where is it headed in 2016?** 2016.

3. Gopinath A, Ravisankar M: **Comparison of lossless data compression techniques**. In: *Proceedings of the 2020 International Conference on Inventive Computation Technologies (ICICT '20)*. 2020; 628–633.
Publisher Full Text

4. Zhang J, Liu G, Ding D , *et al.*: **Transformer and upsampling-based point cloud compression**. In: *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing, and Analysis (APCCPA '22).* New York, NY, USA, 2022; 33–39.
Publisher Full Text

5. Foufoulas Y, Sidirourgos L, Stamatogiannakisn E , *et al.*: **Adaptive compression for fast scans on string columns**. In: *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD '21)*. 2021; 554–562.
Publisher Full Text

6. Jiang H, Liu C, Paparrizos J , *et al.*: **Good to the last bit: data-driven encoding with CodecDB**. In: *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD '21)*. 2021; 843–856.
Publisher Full Text

7. Costa C, Chrysanthis PK, Costa M , *et al.*: **Towards a signature based compression technique for big data storage**. In: *Proceedings of the 39th IEEE International Conference on Data Engineering Workshops (ICDEW '23).* 2023; 100–104.
Publisher Full Text

8. Hsu WH, Zwarico AE: **Automatic synthesis of compression techniques for heterogeneous files.** *Softw. Pract. Exper.* 1995; **25**: 1097–1116.
Reference Source

9. Kattan A, Poli R: **Evolution of human-competitive lossless compression algorithms with GP-zip2.** *Genet Program Evol M.* 2011; **12**: 335–364.
Reference Source

10. Burtchell BA, Burtscher M: **Using Machine Learning to Predict Effective Compression Algorithms for Heterogeneous Datasets.** *Data Compression Conference (DCC).* Snowbird, UT, USA, 2024; 183–192.
Publisher Full Text

11. Abadi DJ, Madden S, Ferreira M: **Integrating compression and execution in column-oriented database systems**. In: *Proceedings of the ACM SIGMOD international conference on management of data*. 2006; 671–682.
Publisher Full Text

12. Idreos S, Kersten ML, Manegold S: **Self-organizing tuple reconstruction in column-stores**. In: *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD '09)*. 2009; 297–308.
Publisher Full Text

13. Yan Y, Chen LJ, Zhang Z: **Error-bounded sampling for analytics on big sparse data.** *Proceedings of the VLDB Endowment.* 2014; **7**(13): 1508–1519.
Publisher Full Text

14. Choukse E, Erez M, Alameldeen AR: **Compresso: pragmatic main memory compression**. In: *Proceedings of the 51st annual IEEE/ACM international symposium on microarchitecture (MICRO '18).* 2018; 546–558.
Publisher Full Text

15. Habich D, Damme P, Ungethüm A , *et al.*: **MorphStore - in-memory query processing based on morphing compressed intermediates LIVE**. In: *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD '19).* 2019; 1917–1920.
Publisher Full Text

16. Podolski P, Ludziejewski T, Nguyen HS: **Data management in training AI chatbot with personality based on granular computing**. In: *Proceedings of the 2022 IEEE International Conference on Big Data (Big Data '22)* 2022; 6247–6252.
Publisher Full Text

17. Berezkin A, Slepnev A, Kirichek R , *et al.*: **Data compression methods based on neural networks**. In: *Proceedings of the 5th International Conference on Future Networks & Distributed Systems (ICFNDS 2021)*. 2021; 511–515.
Publisher Full Text

18. Costa C, Charalampous A, Konstantinidis A , *et al.*: **Decaying telco big data with data postdiction**. In: *Proceedings of the 19th IEEE International Conference on Mobile Data Management (MDM '18)*. 2018; 106–115.
Publisher Full Text

19. Costa C, Konstantinidis A, Charalampous A, *et al.*: **Continuous decaying of telco big data with data postdiction.** *GeoInformatica.* 2019; **23**(4): 533–557.
Publisher Full Text

20. Zhang R: **Data reduction**. In: *Encyclopedia of Database Systems.*, Liu L, Tamer Özsu M (Eds.), Springer, 2016; 1–6.
Publisher Full Text

21. Abebe M, Lazu H, Daudjee K: **Proteus: autonomous adaptive storage for mixed workloads**. In: *Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD '22).* 2022; 700–714.
Publisher Full Text

22. Reddy GT, Reddy MPK, Lakshmanna K, *et al.*: **Analysis of dimensionality reduction techniques on big data.** *IEEE Access.* 2020; **8**: 54776–54788.
Publisher Full Text

23. Ghita B, Tomé DG, Boncz PA: **White-box compression: learning and exploiting compact table representations**. In: *Proceedings of the 10th Conference on Innovative Data Systems Research (CIDR '20).* 2020.

24. Kuschewski M, Sauerwein D, Alhomssi A, *et al.*: **BtrBlocks: efficient columnar compression for data lakes.** *Proceedings of the ACM on Management of Data.* 2023; **1**(2): 1–26.
Publisher Full Text

25. Yuan Z, Hendrix W, Son SW , *et al.*: **Parallel implementation of lossy data compression for temporal data sets**. In: *Proceedings of the 2016 IEEE 23rd International Conference on High Performance Computing (HiPC '16).* 2016; 62–71.
Publisher Full Text

26. Stonebraker M, Abadi DJ, Batkin A , *et al.*: **C-Store: a column-oriented DBMS**. In: *Proceedings of the 31st International Conference on Very Large Data Bases.* 2005; 553–564.

27. Rousseeuw PJ: **Silhouettes: a graphical aid to the interpretation and validation of cluster analysis.** *J Comput Appl Math.* 1987; **20**(20): 53–65.
Publisher Full Text

28. Cormode G, Garofalakis M, Haas PJ, *et al.*: **Synopses for massive data: samples, histograms, wavelets, sketches.** *Foundations and Trends in Databases.* 2011; **4**(1–3): 1–294.
Publisher Full Text

29. Kersten ML: **Big data space fungus**. In: *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR '15).* 2015.

30. Kersten ML, Sidirourgos L: **A database system with Amnesia**. In: *Proceedings of the 7th Biennial Conference on Innovative Data Systems Research (CIDR '17).* 2017.

31. Pelkonen T, Franklin S, Teller J, *et al.*: **Gorilla: a fast, scalable, in-memory time series database.** *Proceedings of the VLDB Endowment.* 2015; **8**(12): 1816–1827.
Publisher Full Text

32. Liu C, Jiang H, Paparrizos J, *et al.*: **Decomposed bounded floats for fast compression and queries.** *Proceedings of the VLDB Endowment.* 2021; **14**(11): 2586–2598.
Publisher Full Text

33. Liakos P, Papakonstantinopoulou K, Kotidis Y: **Chimp: efficient lossless floating point compression for time series databases.** *Proceedings of the VLDB Endowment.* 2022; **15**(11): 3058–3070.
Publisher Full Text

34. Shannon CE: **A mathematical theory of communication.** *Bell System Technical Journal.* 1948; **27**(3): 379–423.
Publisher Full Text

35. Costa C, Chrysanthis P, Herodotou H , *et al.*: Envmon dataset. *Zenodo.* 2025.
Publisher Full Text

36. Costa C, Chrysanthis P, Herodotou H, *et al.*: **COMPASS: big data compression tool using attribute-based signatures.** *Zenodo.* 2025; **5**.
Publisher Full Text

Open Peer Review

Current Peer Review Status: ✔ ✔

Version 2

Reviewer Report 12 September 2026

https://doi.org/10.21956/openreseurope.25401.r80664

© **2026 Burtscher M.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Martin Burtscher**
Texas State University, San Marcos, Texas, USA

The authors have addressed my previous comments.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: high-speed data compression

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Reviewer Report 11 September 2026

https://doi.org/10.21956/openreseurope.25401.r80665

© **2026 Theodoropoulou A.** This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

✔ **Alexandra Theodoropoulou**
University of Patras, Patras, Greece

My suggestions have been sufficiently addressed. I therefore recommend the manuscript's indexation.

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: ICT in Management, Digital Marketing, Benchmarking big-data systems and distributed ML, Decision-support analytics, Performance evaluation of data

pipelines, Reproducible data engineering, Clustering for systems decisions

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Version 1

Reviewer Report 11 October 2025

https://doi.org/10.21956/openreseurope.20832.r61752

© 2025 Burtscher M. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Martin Burtscher
Texas State University, San Marcos, Texas, USA

Summary:

This paper describes and evaluates COMPASS, a compression approach for databases. It clusters the columns of the database and then finds the best compression algorithm from a list of a few compressors for each cluster (i.e., group of similar columns). It significantly outperforms prior approaches that only use one or two compression algorithms for the entire database.

Recommendations:

The report does not include any speed results. If the operation of COMPASS is too slow, it may not be practical even if it delivers superior compression ratios. Hence, I recommend performance results be included.
Using only a single database likely biases the results and makes them non-general. It would be good to evaluate COMPASS on a set of different databases.
The report mentions Zstandard but does not compare to it. I have found that Zstandard works very well in practice and recommend using it as one of the compressors that can be chosen by COMPASS.
The statement "The hypothesis of COMPASS is that [a] multiple data compression scheme can be more effective for complex big data because it allows for incremental compression and partial decompression" should be rephrased. Incremental compression and partial decompression are independent and can also be supported by a single-compression scheme.

Citations:
There is quite a bit of related work on compressing heterogeneous datasets with different compressors.

[Reference 1]
[Reference 2]

[Reference 3]

References

1. Hsu W, Zwarico A: Automatic synthesis of compression techniques for heterogeneous files. *Software: Practice and Experience*. 1995; **25** (10): 1097-1116 Publisher Full Text
2. Kattan A, Poli R: Evolution of human-competitive lossless compression algorithms with GP-zip2. *Genetic Programming and Evolvable Machines*. 2011; **12** (4): 335-364 Publisher Full Text
3. B. Burtchell and M. Burtscher: Using Machine Learning to Predict Effective Compression Algorithms for Heterogeneous Datasets.2024.

Is the work clearly and accurately presented and does it cite the current literature?
Partly

Is the study design appropriate and does the work have academic merit?
Yes

Are sufficient details of methods and analysis provided to allow replication by others?
Yes

If applicable, is the statistical analysis and its interpretation appropriate?
Yes

Are all the source data underlying the results available to ensure full reproducibility?
Yes

Are the conclusions drawn adequately supported by the results?
Yes

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: high-speed data compression

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 18 Mar 2026

Constantinos Costa

We would like to sincerely thank Dr. Burtscher for the constructive feedback and helpful suggestions. We have revised the manuscript accordingly, and all modifications are reflected in the new version (V2) of the manuscript.
- Runtime evaluation
The reviewer noted that the original manuscript did not report runtime results. Response: We agree that runtime performance is an important practical consideration. The revised manuscript now includes runtime measurements for compression and decompression for

both COMPASS and the baseline methods. These results are presented in Table 1, allowing readers to better assess the computational overhead associated with clustering and the selection of a compression scheme. - Dataset diversity
The reviewer pointed out that evaluating the method on a single dataset may limit generalizability. Response:
We acknowledge that evaluating the method on a single dataset may limit generalizability. A paragraph discussing this limitation and outlining future evaluation on additional datasets has been added to the Conclusion section. - Zstandard comparison
The reviewer recommended including Zstandard as a reference compressor. Response:
We appreciate this suggestion. The current prototype implementation of COMPASS selects compression algorithms from those available in the Python zipfile library for simplicity and portability. We have clarified this design choice in Section 2 and noted that the framework is extensible and can incorporate additional compressors such as Zstandard or LZ4 in future implementations. - Hypothesis statement clarification
The reviewer suggested rephrasing the hypothesis statement. Response:
We have revised the hypothesis statement in Section 3 to clarify that the main objective of COMPASS is to exploit heterogeneity in relational datasets by selecting compression schemes tailored to groups of columns with similar statistical characteristics. Related work
The reviewer suggested additional relevant references. Response:
We thank the reviewer for highlighting these studies. The Related Work section has been expanded to include the suggested references on automated and heterogeneous compression techniques. We thank Dr. Burtscher again for the helpful comments and for pointing out relevant related work that strengthened the manuscript.

Competing Interests: No competing interests were disclosed.

Reviewer Report 08 October 2025

https://doi.org/10.21956/openreseurope.20832.r61751

© 2025 Theodoropoulou A. This is an open access peer review report distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

?

Alexandra Theodoropoulou
University of Patras, Patras, Greece

Summary of the article:

The manuscript proposes COMPASS, a “multiple compression” approach for relational data. It partitions a table’s columns into groups and selects a compression scheme per group. Two variants are evaluated: COMPASS-D (K-Means on data values) and COMPASS-E (K-Means on column entropy). The method chooses the number of clusters using the silhouette coefficient and then “exhaustively” tests compression schemes for low-silhouette clusterings. Experiments on a real environmental monitoring database (Envmon, ~1 GB) show disk-space reductions: COMPASS

variants store data at roughly 2–18.2% of RAW, versus 4–23% for baselines; best-case gains over baseline reach ~56%. Code and data are reported as available on GitHub/Zenodo.

Is the work clearly and accurately presented and does it cite the current literature?
Partly — the main idea is clear and the citations are generally relevant, but there are wording/typo issues and a few gaps in positioning against common codecs.

Is the study design appropriate and does the work have academic merit?
Yes — the problem is meaningful, the approach is reasonable, and the contribution is of interest; the evaluation set could be broader, but the design itself has merit.

Are sufficient details of methods and analysis provided to allow replication by others?
Partly — data/code are provided, yet some key details are missing (e.g., seeds, clustering settings, small ablations), which makes exact replication harder.

If applicable, is the statistical analysis and its interpretation appropriate?
Not applicable — the paper is primarily systems/engineering with clustering and measurements; it does not hinge on formal statistical inference.

Are all the source data underlying the results available to ensure full reproducibility?
Partly — availability is claimed, but there are minor inconsistencies (e.g., DOI formatting) that should be fixed to ensure smooth access.

Are the conclusions drawn adequately supported by the results?

Partly — storage gains are shown convincingly; however, efficiency (time/overhead) and generalization claims would benefit from a bit more evidence or careful scoping.

SUGGESTED REVISIONS

Runtime and resources:

It would strengthen the paper to include a few basic efficiency metrics. Even a compact table (or a short note per experiment) with compression time, decompression time, and rough CPU/memory usage would help readers interpret the practical trade-offs. If full profiling feels heavy, indicative averages with a line or two on the test environment may be enough.

Baselines and codecs

Since many readers will be familiar with Zstd, LZ4, or Snappy, adding at least one of these as a reference point could situate COMPASS in everyday practice. If those codecs are out of scope for a reason (e.g., tooling constraints or focus on a specific library), a brief explanation would also work. Either way, a small note here would make the comparisons feel more complete.

Clustering rationale
The choice to prefer lower silhouette values is interesting, but it may not be clear to all readers. A brief note explaining the idea—why lower values can help compression in your case—would help. If possible, add a small example table with k, average silhouette, and the resulting size/time (or a

short ablation). It can be very short; just enough to show the pattern clearly.

Variability / stability
It would help to show that results are stable. Could you run each experiment a few times (e.g., with different random seeds) and report the average plus a small error bar or standard deviation? Even 3 runs are enough. This adds credibility, especially because clustering can vary slightly.

Query-time/read-time
Could you add 2–3 very small timing tests to show how fast data can be read after compression? For example: (a) time to read a whole table, (b) time to run a simple filter (e.g., value > threshold), and (c) time to compute an average on one column. Please report the time (in seconds) for COMPASS and for the baseline. If this is outside the paper's scope at this time, a one-line note is also okay (e.g., "Query-time tests are valuable but out of scope; we plan them for future work").

Study limitations / threats to validity
It would help to add a short, dedicated limitations section. You might briefly note the dataset scope and domain, any sensitivity to the encoding choices, how the silhouette heuristic behaves across settings, and the computational overhead. If you have already checked some of these, a concise summary would be great.

Future research directions
The conclusion could sketch a few next steps in a little more detail. For example, you might point to richer clustering features, possible learning-based selection of codecs, links to DBMS metadata/optimizers, broader codec/encoding evaluations, to name a few. Feel free to frame these in whatever way best fits your vision of the project.

Figure quality
Figure 3 has quite low quality. It would benefit the paper if the authors could increase its quality.

Consistency and references
To improve readability and accuracy, please correct minor errors. There are a few typographical slips—e.g., "COMPASS-E" appears as "COMAPASS-E," and "Envmon dataset" in the Reference section is misspelled. Also, the Shannon citation is labeled as reference 31 in the text but appears as 32 in the reference list; aligning the in-text citation with the bibliography numbering would resolve this mismatch.

Is the work clearly and accurately presented and does it cite the current literature?
Partly

Is the study design appropriate and does the work have academic merit?
Yes

Are sufficient details of methods and analysis provided to allow replication by others?
Partly

If applicable, is the statistical analysis and its interpretation appropriate?
Not applicable

Are all the source data underlying the results available to ensure full reproducibility?
Partly

Are the conclusions drawn adequately supported by the results?
Partly

Competing Interests: No competing interests were disclosed.

Reviewer Expertise: ICT in Management, Digital Marketing, Benchmarking big-data systems and distributed ML, Decision-support analytics, Performance evaluation of data pipelines, Reproducible data engineering, Clustering for systems decisions

I confirm that I have read this submission and believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

Author Response 18 Mar 2026

Constantinos Costa

We would like to sincerely thank Mrs. Theodoropoulou for the constructive feedback and helpful suggestions. We have revised the manuscript accordingly, and all modifications are reflected in the new version (V2) of the manuscript. - Runtime and resource metrics
The reviewer suggested including efficiency metrics such as compression and decompression time, as well as information about the computational environment. Response:
We agree that runtime information is valuable for evaluating the practical implications of the proposed approach. We have therefore added runtime measurements for compression and decompression for COMPASS and the baseline methods. These results are presented in Table 1, along with a brief description of the experimental environment in Section 4.5. - Baselines and commonly used codecs
The reviewer suggested including commonly used compression schemes such as Zstandard, LZ4, or Snappy to better situate COMPASS in practical contexts. Response:
We appreciate this suggestion. This prototype solution implementation of COMPASS selects compression algorithms from those available in the Python zipfile library (e.g., DEFLATE, BZIP2, and LZMA). We have clarified this design choice in the implementation section and added an explanation in Section 2 that the COMPASS framework itself is not limited to these compression schemes and can be extended to incorporate additional compressors such as Zstandard or LZ4. - Clustering rationale and silhouette interpretation
The reviewer asked for clarification regarding the use of lower silhouette values in the clustering process. Response:
We have expanded the explanation in Section 3.2 (Selection of best performing compression) to clarify this design choice. Unlike typical clustering tasks that aim to maximize cluster separation, our objective is to group columns that exhibit similar compression behavior. In this context, lower silhouette values may produce groupings that allow compression algorithms to better exploit structural similarities across columns. - Variability and experimental stability

The reviewer suggested reporting multiple runs with different random seeds. Response:
To improve reproducibility, we have clarified that the clustering experiments use a fixed random seed (seed = 42). This information has been added to the methodology description in Section 4.1. - Query-time/read-time considerations
The reviewer suggested including small experiments showing read/query performance. Response:
We appreciate this suggestion. The focus of the current work is on compression effectiveness and storage reduction. Query-time performance is an important aspect that we plan to explore in future work. We have added a brief clarification in Section 5 of the manuscript to acknowledge this point. - Limitations and threats to validity
The reviewer recommended including a limitations section. Response:
We thank the reviewer for the suggestion. We have added a paragraph in the Conclusion section discussing the study's limitations, including the dataset's scope used in the evaluation, and directions for future work. - Future work
The reviewer suggested expanding the discussion of future research directions. Response:
The conclusion (Section 5) has been extended to include several future research directions, including richer clustering features and evaluation on additional datasets and compression schemes. - Minor corrections
The reviewer identified several typographical and formatting issues. Response:
We have carefully revised the manuscript to correct typographical errors (e.g., “COMAPASS-E” → “COMPASS-E”), corrected the spelling of the Envmon dataset reference, aligned the Shannon citation numbering, and improved the quality of Figure 3. We thank Mrs. Theodoropoulou again for the valuable feedback, which helped improve the clarity and completeness of the manuscript.

Competing Interests: No competing interests were disclosed.