Space Complexity in Modern Systems: Challenges and Solutions

# Introduction

In computer science, time complexity is often prioritized when evaluating algorithms and system performance. However, with the explosion of digital data in modern systems—from server logs and IoT telemetry to transactional and user-generated data—the significance of space complexity is becoming impossible to ignore. Uncontrolled data growth can degrade performance, exhaust storage resources, and compromise long-term scalability.

# Literature Review

## Data Retention and Deletion Policies

Retention and deletion policies help organizations define how long data should be stored. Data that no longer holds business or compliance value can be safely discarded, reducing storage needs.

## Log Rotation and Pruning

Log rotation prevents excessive accumulation of log files by archiving or deleting older logs. Pruning further reduces unnecessary information while preserving critical records.

## Data Deduplication

Deduplication eliminates redundant copies of data by storing only unique records. This significantly reduces storage overhead in databases and distributed systems.

## Compression and Archiving

Compression techniques such as gzip and zlib reduce file sizes, making data storage more space-efficient. Archiving groups old or rarely accessed data, preserving resources while ensuring accessibility when needed.

## Real-time Data Aggregation and Summarization

Aggregation reduces storage by summarizing raw data into key metrics, trends, or statistical overviews. This is particularly useful in IoT systems where high-frequency data may not always require detailed retention.

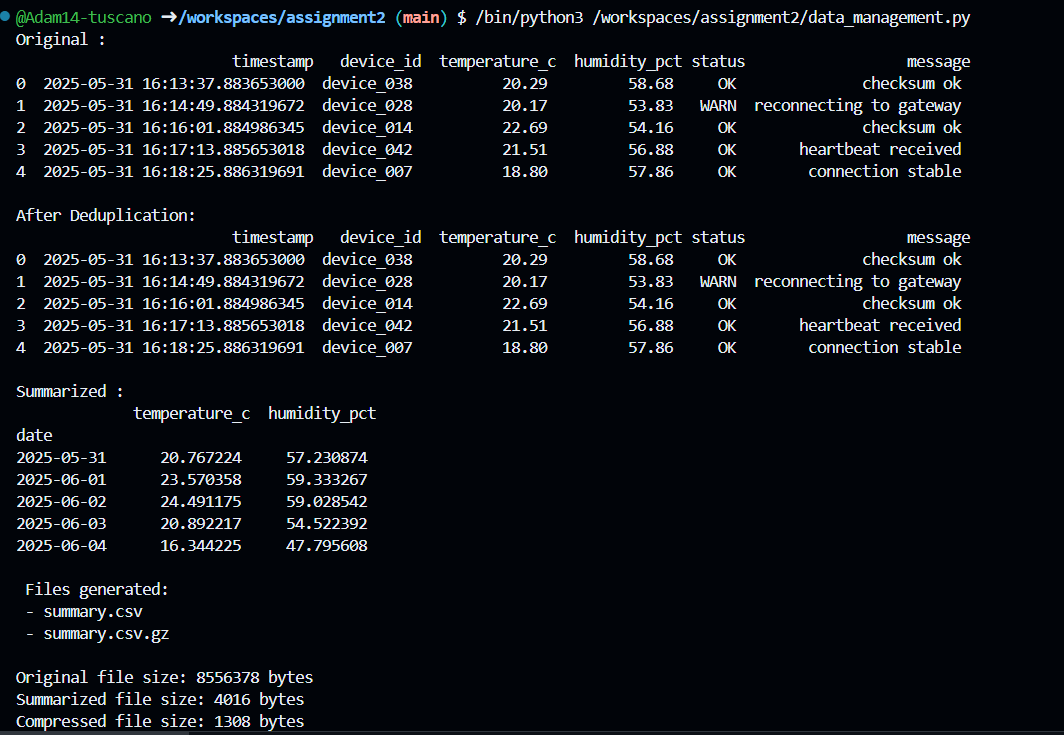
# Proposed Solution

A hybrid approach combining deduplication, compression, and summarization is proposed. This method ensures that redundant and unnecessary data is removed, essential data is compressed, and large data streams are summarized in real-time to prevent uncontrolled growth.

# Methodology and Implementation

The solution is demonstrated through a Python script that operates on sample IoT-like data. The script performs deduplication, trims unnecessary rows, compresses the data into gzip format, and summarizes metrics such as average temperature and humidity per day.

## Sample Python Implementation output :



# Evaluation and Discussion

The hybrid solution effectively reduces data growth through structured space management. However, trade-offs exist: compression introduces CPU overhead, deduplication requires indexing, and summarization may cause loss of granularity. Future extensions may include adaptive algorithms that dynamically adjust space management techniques based on workload and system constraints.