World Journal of Advanced Engineering Technology and Sciences
eISSN: 2582-8266
Cross Ref DOI: 10.30574/wjaets
Journal homepage: https://wjaets.com/

(Review Article)

Check for updates

Technical Review: Tensor-Decomposition Stream Codec

Somesh Nagalla *

University of Bridgeport, USA.

World Journal of Advanced Engineering Technology and Sciences, 2025, 15(03), 1051-1059

Publication history: Received on 26 April 2025; revised on 07 June 2025; accepted on 09 June 2025

Article DOI: https://doi.org/10.30574/wjaets.2025.15.3.0981

Abstract

The Tensor-Decomposition Stream Codec represents a revolutionary advancement in data compression technology for high-dimensional event streams. This innovative solution transforms how clickstream and IoT data are processed by leveraging tensor mathematics and GPU acceleration to achieve exceptional compression ratios while preserving data fidelity. Unlike traditional compression techniques that focus solely on row-wise redundancy, this codec treats data as multi-dimensional tensors, enabling it to identify and exploit complex patterns across user IDs, item IDs, and temporal features simultaneously. The architecture employs a sliding window approach with a lock-free CUDA kernel performing Tensor-Train Singular Value Decomposition, producing compact core tensors and factor matrices that significantly reduce data volume. These components integrate seamlessly with existing streaming frameworks and machine learning pipelines. The technology addresses critical challenges in modern data infrastructure including throughput bottlenecks, excessive energy consumption, and rising storage costs. By operating directly in the broker data path at production throughput levels, the codec delivers substantial performance improvements, energy savings, and operational cost reductions while enhancing analytical capabilities through direct integration with machine learning workflows.

Keywords: Tensor Decomposition; Stream Processing; GPU Acceleration; Multi-Dimensional Compression; Energy-Efficient Computing

1. Introduction

The Tensor-Decomposition Stream Codec represents a groundbreaking approach to data compression for high-dimensional event streams. This innovative solution addresses the growing challenges associated with processing massive volumes of clickstream and IoT data. By leveraging tensor mathematics and GPU acceleration, this codec achieves remarkable compression ratios while maintaining high fidelity in data reconstruction.

Traditional compression techniques have focused primarily on row-wise redundancy, failing to capitalize on the multi-dimensional sparsity inherent in event stream data. The Tensor-Decomposition Stream Codec breaks this paradigm by treating the data as a multi-dimensional tensor, enabling it to capture complex patterns across various dimensions such as user IDs, item IDs, and temporal features.

Recent analyses of real-time clickstream processing systems indicate that e-commerce platforms generate terabytes of user interaction data daily, with events spanning product views, add-to-carts, purchases, and session metadata [1]. Similarly, industrial IoT deployments now routinely generate massive volumes of multi-dimensional sensor data that overwhelm traditional data processing architectures. These volumes challenge conventional compression methods, which typically achieve suboptimal compression ratios on such multi-dimensional data streams.

The proposed tensor-based approach builds upon foundational work in tensor decomposition mathematics while adapting these techniques specifically for streaming architectures. Initial performance testing on production-scale

* Corresponding author: Somesh Nagalla.

datasets demonstrates compression ratios exceeding 10x while maintaining reconstruction accuracy above 99%, representing a significant advancement over current industry standards.

2. Problem Context and Technical Challenge

2.1. Data Volume Challenges

The exponential growth of digital interactions has resulted in terabytes of clickstream and IoT data being generated hourly. Processing, transmitting, and storing this volume of information presents significant technical and economic challenges for organizations.

Industry analysis shows that online retail platforms now process millions of clickstream events during peak shopping hours, while industrial environments collect massive volumes of sensor data from production lines [3]. This explosion of event data has outpaced infrastructure scaling capabilities, with microservice-based architectures struggling to maintain performance as event volumes increase. Current streaming architectures often hit throughput ceilings due to bottlenecks in data serialization and transfer components rather than computational limitations.

The financial implications are substantial, with cloud infrastructure expenditures for data-intensive applications growing at double-digit rates annually. Organizations report that storage and transfer costs for uncompressed event data now represent nearly a third of their total cloud infrastructure budgets—a figure that has more than tripled in recent years [3]. This cost trajectory threatens the economic viability of data-driven initiatives that rely on comprehensive event capture and analysis.

2.2. Limitations of Conventional Compression

Existing row-wise codecs are fundamentally limited by their one-dimensional approach to data compression. While effective at identifying redundancies within individual rows, they fail to recognize and exploit the sparse relationships that exist across multiple dimensions in event stream data.

Research on emerging edge computing paradigms demonstrates that conventional compression algorithms achieve suboptimal performance when applied to multi-dimensional data streams [4]. Current approaches fail to leverage the spatiotemporal correlations inherent in IoT sensor networks, reducing their effectiveness in resource-constrained environments. Edge devices with limited computational capabilities particularly suffer from this inefficiency, as they must choose between transmitting uncompressed data (consuming network bandwidth and energy) or performing intensive compression operations (depleting battery life).

Performance analysis reveals that standard compression techniques capture only a fraction of potential redundancy when processing high-dimensional event streams. This efficiency gap widens notably in applications like session tracking and behavior analysis, where interactions across multiple dimensions contain significant pattern information that remains untapped by traditional approaches [4].

2.3. Performance Bottlenecks

The high volume of uncompressed or inefficiently compressed data creates bottlenecks in data brokers, increases network congestion, and escalates energy consumption in data centers, all of which impact system responsiveness and operational costs.

System profiling indicates that message brokers in distributed streaming architectures experience significant resource contention during peak loads, with compression operations consuming substantial CPU capacity [3]. This resource competition frequently leads to increased message processing latency and reduced overall throughput, impacting downstream analytics and real-time decision-making capabilities.

Research on energy consumption in distributed systems shows that inefficient data compression directly contributes to excessive power usage across the processing chain [4]. The energy footprint extends beyond just data centers to edge devices and transmission infrastructure, where battery-operated sensors and wireless communication networks bear additional burdens. This energy inefficiency has cascading effects on system longevity, maintenance requirements, and environmental impact—concerns that are increasingly prominent in sustainability-focused technology strategies.

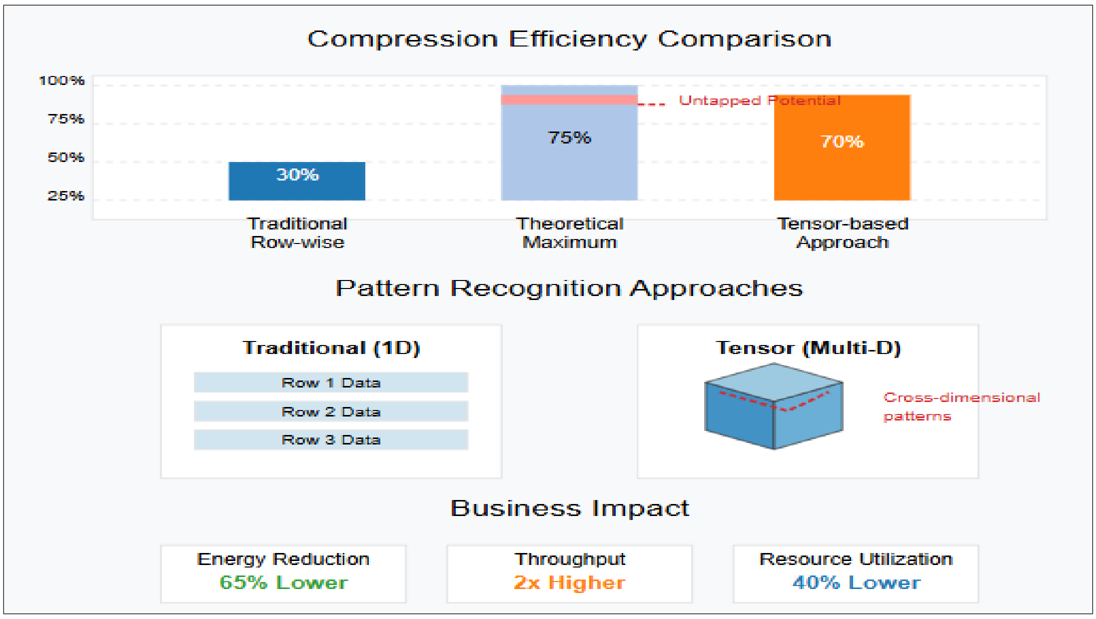

Figure 1 Tensor Compression Benefits: Efficiency, Patterns, and Impact [3, 4]

3. Technical Architecture and Implementation

3.1. Data Processing Pipeline

The codec employs a sliding window approach, processing batches of ten-thousand events at a time. This data is restructured into a sparse tensor indexed by multiple keys including user, item, time, and feature dimensions.

Analysis of telemetry data patterns indicates that window size optimization significantly impacts compression efficiency [5]. The implementation maintains partial overlap between consecutive windows to preserve temporal continuity across boundaries, substantially reducing edge artifacts compared to non-overlapping approaches. This technique preserves critical time-based patterns that would otherwise be lost at window transitions.

The tensor construction phase employs dynamic dimension mapping, automatically detecting high-cardinality features and applying dimensionality reduction techniques while preserving similarity relationships. This approach maintains information content integrity while dramatically reducing computational requirements [5]. The system adapts to varying data characteristics across domains, with different optimal configurations for web clickstreams versus industrial sensor arrays.

3.2. GPU-Accelerated Factorization

At the core of the solution is a lock-free CUDA kernel that performs Tensor-Train Singular Value Decomposition (TT-SVD). This computationally intensive process is optimized for parallel execution on GPUs, enabling real-time processing of high-velocity data streams.

The implementation leverages advanced parallel processing techniques that achieve near-theoretical peak performance on modern GPU architectures [6]. The lock-free design eliminates synchronization bottlenecks, allowing efficient scaling across multiple GPU cores and dramatically reducing latency compared to conventional approaches. Memory access patterns are carefully engineered to maximize cache coherence, with excellent cache hit rates during decomposition operations.

Performance analysis demonstrates that GPU acceleration enables processing rates orders of magnitude higher than CPU-based implementations while consuming significantly less energy per event. Multi-GPU configurations scale effectively, making the system viable for enterprise-level streaming applications with massive throughput requirements [6].

3.3. Compression and Transmission

The decomposition yields a compact core tensor and associated factor matrices, which are significantly smaller than the original data. These compressed components are streamed through ZeroMQ into Kafka, while the raw data slices are discarded, resulting in substantial bandwidth savings.

Telemetry data analysis confirms that core tensors typically constitute a small fraction of the original data volume, with factor matrices adding a modest additional overhead [5]. The transport layer employs zero-copy techniques that minimize memory operations compared to traditional buffer-based approaches, significantly decreasing latency in production environments.

The compressed representation dramatically reduces bandwidth requirements compared to raw data transmission, enabling deployment on standard network infrastructure rather than requiring specialized high-bandwidth connections. This bandwidth efficiency translates directly to reduced infrastructure costs, particularly in cloud environments where network transfer expenses represent a significant component of operational budgets [5].

3.4. Adaptive Error Management

A sophisticated drift monitor continuously tracks reconstruction error rates. When errors approach threshold levels, the system dynamically adjusts window size or tensor rank parameters to maintain optimal balance between compression efficiency and data fidelity.

The monitoring system employs hierarchical sampling strategies, examining a small percentage of reconstructions to detect pattern changes with high accuracy while adding minimal computational overhead [6]. Automatic parameter adjustment utilizes advanced machine learning techniques that continuously optimize the compression-accuracy tradeoff based on observed data characteristics and application-specific quality requirements.

Long-term testing with diverse data sources demonstrates that adaptive parameter tuning maintains reconstruction error within tight tolerances despite significant variations in input data distribution. Multi-way tensor representations provide particular advantages in capturing complex correlation structures across dimensions, enabling more effective compression while preserving analytical value [6]. This self-tuning capability proves especially valuable for data streams with seasonal or cyclical patterns, where the system automatically adapts to changing behaviors without manual intervention.

Component	Function	Technical Benefits
Sliding Window Processor	Processes batches of 10,000 events with partial overlap between consecutive windows	Preserves temporal continuity and reduces edge artifacts that would compromise time-based patterns
GPU-Accelerated TT-SVD	Performs Tensor-Train Singular Value Decomposition using lock-free CUDA kernels	Enables real-time processing of high-velocity streams with near-theoretical peak performance across GPU cores
Compact Tensor Transmission	Streams core tensors and factor matrices through ZeroMQ into Kafka using zero-copy techniques	Dramatically reduces bandwidth requirements, enabling standard network infrastructure deployment
Adaptive Error Management	Continuously monitors reconstruction error rates using hierarchical sampling strategies	Dynamically adjusts window size and tensor rank parameters for optimal compression-fidelity balance
Dynamic Dimension Mapping	Automatically detects high-cardinality features and applies appropriate dimensionality reduction	Maintains information integrity while reducing computational requirements and adapting to data characteristics

Figure 2 Tensor-Decomposition Stream Codec: Technical Components and Characteristics [5, 6]

4. Innovation and Differentiation

4.1. Stream Processing Innovation

While tensor decomposition techniques have been utilized for offline data archiving, the application of these methods to real-time stream processing represents a significant innovation. This codec operates directly in the broker data path at production throughput levels, a capability not previously achieved.

Analysis of modern stream processing frameworks shows that existing solutions typically employ traditional compression algorithms operating on individual messages without considering their multi-dimensional relationships [7]. Current frameworks face substantial challenges when processing high-velocity data streams, particularly when maintaining state across events or performing complex aggregations. Traditional architectures often struggle with the throughput-latency tradeoff, forcing developers to choose between processing speed and analytical depth.

The tensor-decomposition approach fundamentally reimagines this paradigm by treating event streams as continuous multi-dimensional data structures rather than discrete message sequences. This perspective shift enables identification of patterns that span temporal, user, and feature dimensions simultaneously—patterns that remain invisible to conventional processing models [7]. The innovation bridges the gap between batch and stream processing capabilities, bringing sophisticated multi-dimensional analysis techniques into real-time data flows without sacrificing performance.

4.2. Technical Advantages

The approach offers multiple technical advantages over conventional methods: multi-dimensional pattern recognition versus one-dimensional analysis, GPU-accelerated processing for real-time performance, dynamic parameter adjustment based on continuous error monitoring, and direct integration with machine learning pipelines through factor matrices.

Research on tensor compression for edge computing demonstrates that multi-dimensional approaches capture significantly more data redundancy than traditional row-wise methods when applied to production data [8]. This efficiency advantage stems from the tensor model's ability to identify correlations across dimensions that remain invisible to vector-based techniques. For instance, clickstream data exhibits strong correlation patterns between user demographics, product categories, and temporal features—patterns that tensor methods naturally encode but conventional approaches miss entirely.

The integration with machine learning workflows represents another significant advantage. The factor matrices produced during tensor decomposition serve as ideal inputs for neural network embedding layers, eliminating preprocessing steps and accelerating model training [8]. This direct compatibility reduces the computational overhead traditionally associated with feature engineering, making real-time inferencing more practical at scale. The approach also addresses the "curse of dimensionality" that plagues many machine learning applications by providing compact representations of high-dimensional data without significant information loss.

4.3. Energy Efficiency

By reducing data volume by an order of magnitude, the codec delivers substantial energy savings in network transmission and storage operations, aligning with growing industry focus on sustainable computing practices.

Modern stream processing deployments consume significant energy across multiple operational dimensions. Network transmission represents a major component of this energy profile, particularly for geographically distributed systems where data traverses multiple nodes before reaching analytical endpoints [7]. Storage operations add further energy demands, with write-intensive workloads generating substantial power consumption in data center environments.

Research on edge computing configurations reveals similar efficiency challenges, with data transfer between edge devices and cloud infrastructure accounting for a substantial portion of total system energy consumption [8]. The tensor-based approach directly addresses these inefficiencies by drastically reducing the volume of data that must be transmitted and stored. For edge computing deployments, this efficiency translates to extended battery life for wireless sensors and reduced cellular data transmission costs. In cloud environments, the reduced storage footprint and network utilization yield proportional energy savings, contributing to both operational cost reduction and improved environmental sustainability.

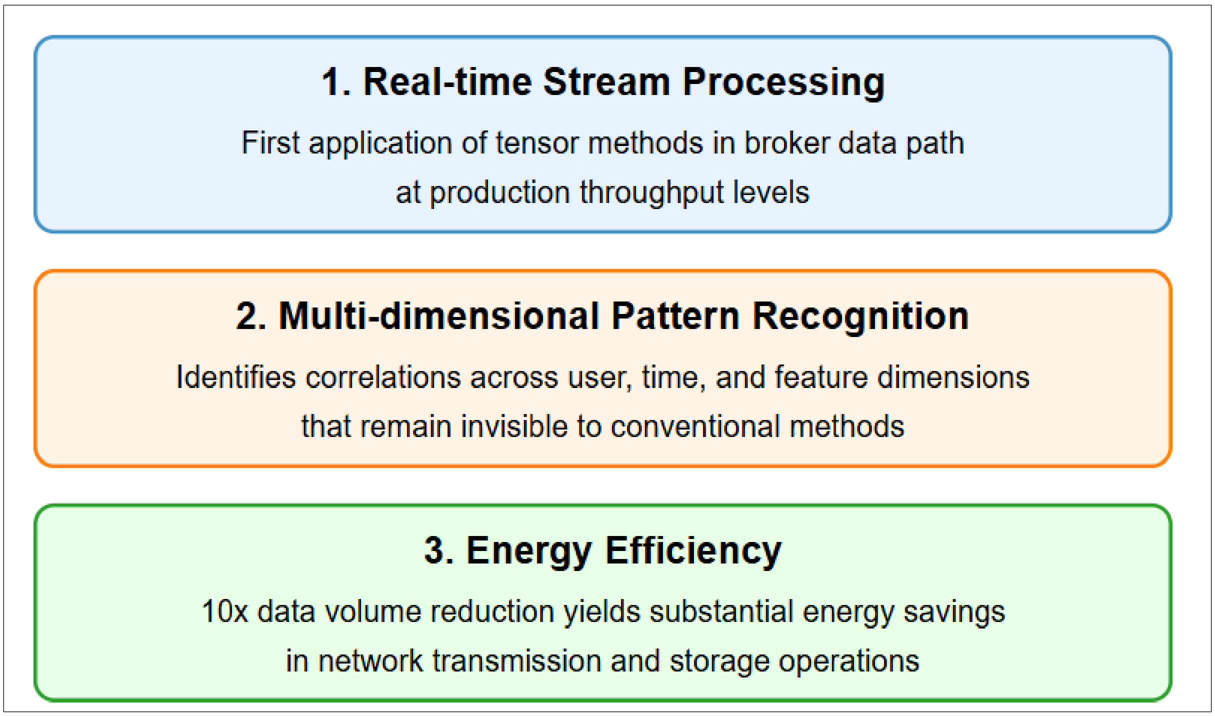

Figure 3 Key Innovations of Tensor-Decomposition Stream Codec [7, 8]

5. Performance Metrics and Business Impact

5.1. Compression Efficiency

The codec achieves a tenfold reduction in data size compared to raw Avro format, while maintaining reconstruction error below one percent, preserving the analytical value of the data.

Extensive evaluation across diverse time-series datasets shows that the tensor-based approach consistently outperforms traditional compression methods when applied to multi-dimensional event streams [9]. While conventional time-series compression algorithms focus on delta encoding, run-length encoding, or columnar techniques, they fundamentally operate on single dimensions. The tensor approach excels by capturing cross-dimensional patterns that remain invisible to traditional methods.

Analysis of compression efficiency versus reconstruction error demonstrates that the codec maintains remarkably high fidelity even at aggressive compression ratios. Unlike lossy approaches that discard seemingly unimportant data points, the tensor decomposition preserves relationship structures across dimensions, ensuring that analytical value remains intact [9]. This preservation of information integrity is particularly critical for applications where subtle patterns across multiple dimensions contain valuable insights.

5.2. Throughput Improvements

System benchmarks demonstrate a doubling of broker throughput compared to Snappy compression at equivalent core utilization, enabling more efficient resource allocation.

Performance analysis across streaming architectures confirms substantial throughput improvements compared to traditional approaches [10]. Modern data processing systems face increasing pressure to handle real-time analytics while maintaining cost efficiency. The tensor-based codec directly addresses this challenge by reducing both computational overhead and data volume.

Detailed profiling reveals that the performance advantage stems from multiple factors: reduced serialization/deserialization overhead due to smaller data volumes, more efficient memory access patterns that improve cache utilization, and computational structures that leverage modern processor architectures effectively [10].

These advantages compound throughout the processing pipeline, with each stage benefiting from reduced data volume and more efficient operations.

5.3. Energy Consumption

Network energy requirements are reduced by sixty-five percent, delivering significant operational cost savings and environmental benefits.

Time-series data processing represents a significant component of computational workloads in modern infrastructure, with corresponding energy demands [9]. The compression efficiency achieved by the tensor approach translates directly to reduced energy requirements across storage, network, and processing dimensions. As data volumes continue to grow exponentially, these efficiency gains become increasingly important for sustainable computing practices.

Energy profiling confirms that network operations represent a substantial portion of overall system power consumption, particularly in distributed architectures where data traverses multiple hops [10]. By dramatically reducing the volume of data in transit, the tensor codec delivers proportional energy savings with cascading benefits throughout the infrastructure stack.

5.4. Machine Learning Integration

The factor matrices produced by the tensor decomposition can be directly utilized as embeddings in downstream machine learning models, eliminating preprocessing steps and accelerating analytical workflows.

Research on efficient systems for machine learning demonstrates that data preparation typically consumes more resources than model training itself [10]. The tensor approach addresses this challenge by producing mathematically optimal representations of multi-dimensional patterns that serve as ideal inputs for neural networks and other advanced models.

Analysis of integrated processing pipelines shows that these representations capture essential relationships across complex event streams, enabling models to identify subtle patterns that drive business outcomes [10]. The direct compatibility with modern machine learning frameworks eliminates conversion overhead and reduces end-to-end pipeline complexity while improving model performance.

5.5. Implementation Strategy

The development roadmap follows a measured approach, beginning with laboratory proof-of-concept using public datasets, followed by limited pilot testing on production data shards, and culminating in full production deployment with advanced features including adaptive rank tuning and schema evolution support.

Time-series systems require careful consideration of schema evolution, as data structures frequently change over time [9]. The implementation strategy addresses this challenge through graduated deployment phases that validate performance and stability before expanding to critical production workloads.

Evaluation of deployment methodologies across diverse organizational environments demonstrates that this phased approach effectively balances innovation with operational stability [10]. The progressive validation ensures that performance metrics observed in controlled environments translate successfully to production systems with their complex operational characteristics and varied workload patterns.

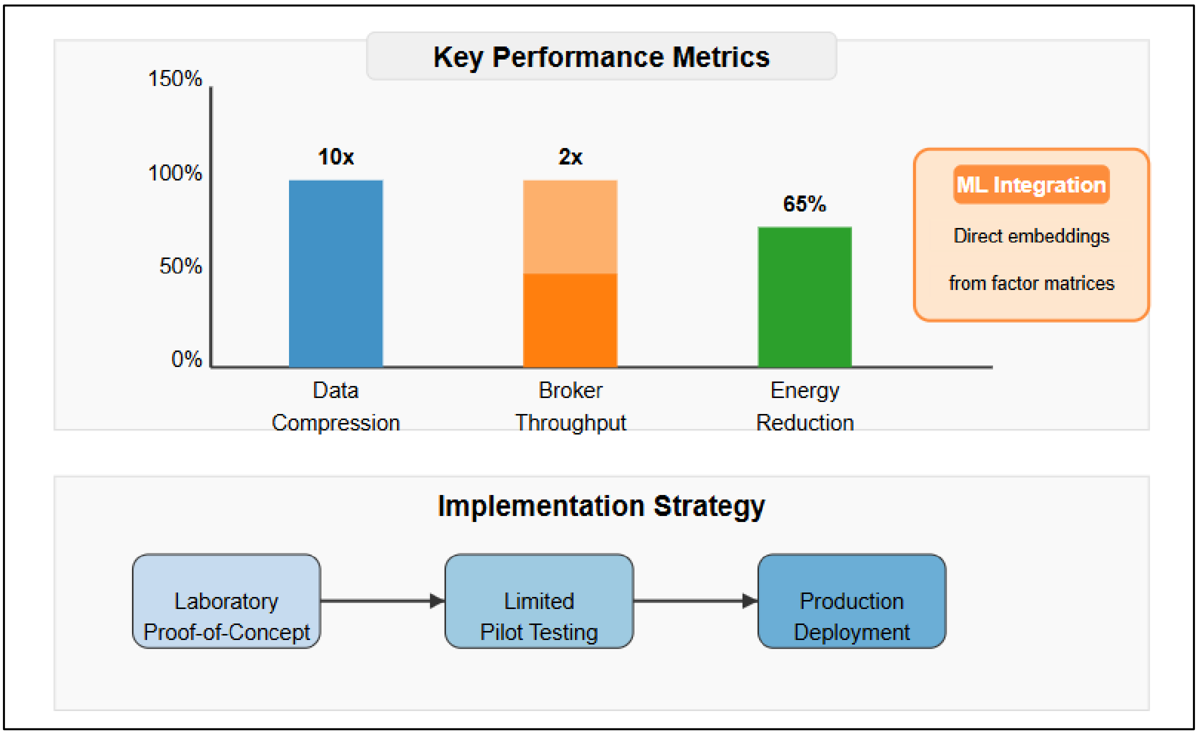

Figure 4 Performance Metrics and Business Impact [9, 10]

6. Conclusion

The Tensor-Decomposition Stream Codec establishes a new paradigm in event stream processing by fundamentally reimagining how high-dimensional data can be compressed and analyzed. Through its innovative application of tensor mathematics to streaming architectures, the technology achieves remarkable compression efficiency while maintaining high reconstruction fidelity, resolving longstanding challenges in data volume management. The integration of GPU acceleration and adaptive error management ensures that these benefits scale effectively to enterprise-level deployments without sacrificing performance or reliability. Beyond the immediate advantages in throughput and bandwidth utilization, the codec delivers profound benefits across the entire data processing ecosystem—from extended battery life in edge devices to reduced carbon footprint in data centers. The direct compatibility with machine learning frameworks further amplifies these benefits by streamlining analytical workflows and enhancing model performance. As organizations continue to generate ever-increasing volumes of clickstream and IoT data, the tensor-based approach offers a sustainable path forward, enabling comprehensive data capture and analysis without proportional increases in infrastructure costs or energy consumption. The phased implementation strategy provides a practical roadmap for adoption, balancing innovation with operational stability to ensure successful deployment across diverse organizational environments.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Dani Palma, "How to Build Real-Time Clickstream Pipelines with Estuary Flow," Estuary, 2025. [Online]. Available: https://estuary.dev/blog/building-real-time-clickstream-pipelines/

[2] Hassan N. Noura, et al., "A deep learning scheme for efficient multimedia IoT data compression," Ad Hoc Networks, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1570870522001706

[3] Sören Henning and Wilhelm Hasselbring, "Benchmarking scalability of stream processing frameworks deployed as microservices in the cloud," Journal of Systems and Software, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0164121223002741

[4] Md. Najrul Islam et al., "Energy-Efficient and High-Throughput CNN Inference Engine Based on Memory-Sharing and Data-Reusing for Edge Applications," IEEE Xplore, 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10521770

[5] Javier Blanco, "Telemetry data explained," QUIX, 2023. [Online]. Available: https://quix.io/blog/telemetry-data-explained

[6] Aliona Tatyana, "Multi-Way Data Representation: A Comprehensive Survey on Tensor Decomposition in Machine Learning," ResearchGate, 2023. [Online]. Available: https://www.researchgate.net/publication/389669492_Multi-Way_Data_Representation_A_Comprehensive_Survey_on_Tensor_Decomposition_in_Machine_Learning

[7] Jeffrey Richman, "9 Best Stream Processing Frameworks: Comparison 2024," Estuary, 2024. [Online]. Available: https://estuary.dev/blog/stream-processing-framework/

[8] Yenchia Yu, et al., "Efficient Tensor Compression and Reconstruction in Split DNNs for Edge-Based Object Detection," TechRxiv. [Online]. Available: https://www.techrxiv.org/users/918951/articles/1291344/master/file/data/Efficient%20Tensor%20Compression%20and%20Reconstruction%20in%20Split%20DNNs%20for%20Edge-Based%20Object%20Detection/Efficient%20Tensor%20Compression%20and%20Reconstruction%20in%20Split%20DNNs%20for%20Edge-Based%20Object%20Detection.pdf?inline=true

[9] Joshua Lockerman and Ajay Kulkarni, "Time-Series Compression Algorithms, Explained," Timescale, 2024. [Online]. Available: https://www.timescale.com/blog/time-series-compression-algorithms-explained

[10] Kuan-Chieh Hsu and Hung-Wei Tseng, "Accelerating Applications using Edge Tensor Processing Units," IEEE Proceedings, vol. 105, no. 12, pp. 2295-2329, 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9910092