European Journal of Computer Science and Information Technology, 13(52),185-194, 2025

Print ISSN: 2054-0957 (Print)

Online ISSN: 2054-0965 (Online)

Website: https://www.eajournals.org/

Publication of the European Centre for Research Training and Development -UK

AI-Driven Data Warehousing: ML Innovations for Performance, Prediction, and Cost Optimization

Vigneshwar Rangini
Capgemini America, USA

doi: https://doi.org/10.37745/ejcsit.2013/vol13n52185194 Published December 13, 2025

Citation: Rangini V. (2025) AI-Driven Data Warehousing: ML Innovations for Performance, Prediction, and Cost Optimization, *European Journal of Computer Science and Information Technology*, 13(52),185-194

Abstract: *The rapid expansion of big data has accelerated the evolution of Data Warehousing (DWH) from static, rule-based systems to adaptive, intelligent, and automated frameworks powered by Machine Learning (ML). Traditional data warehouses face challenges in scalability, efficiency, and real-time analytics, which ML can effectively address. This paper presents an integrated ML-driven optimization framework that enhances data storage, query performance, and analytical capabilities across ingestion, transformation, and execution layers. The framework leverages supervised, unsupervised, and reinforcement learning techniques to predict query costs, optimize execution plans, detect anomalies, and forecast workloads for dynamic resource allocation. AI-powered automation further improves data integration, schema evolution, and adaptability to changing workloads. Experimental evaluations on PostgreSQL and cloud-native environments demonstrate measurable gains in query latency reduction, storage efficiency, and operational cost optimization through adaptive indexing, workload forecasting, and ML-assisted plan selection. The study also addresses emerging challenges such as computational complexity, data security, and model explainability, along with the potential of cloud-based and federated learning for distributed data management. By embedding ML intelligence within DWH operations, organizations can achieve predictive scalability, cost efficiency, and governance assurance, transforming traditional warehouses into autonomous, self-optimizing analytical systems aligned with modern business needs.*

Keywords: data warehousing, machine learning, query optimization, adaptive indexing, workload forecasting, anomaly detection, cloud data warehouse, self-tuning database, data integration, federated learning, data governance, automation.

INTRODUCTION

In the modern digital landscape, organizations generate and store massive volumes of data originating from business transactions, IoT devices, social media platforms, and customer interactions. Traditional Data Warehousing (DWH) systems originally designed to consolidate structured data for analytical and reporting purposes are now challenged by the increasing volume, velocity, and variety of enterprise data. The growing demand for real-time analytics, predictive intelligence, and data-driven decision-making has exposed limitations in conventional rule-based data warehouse architectures, which often struggle with performance scalability, storage optimization, and integration complexity.

Effective data warehousing is the cornerstone of Business Intelligence (BI) and organizational decision support. Enterprises depend on their DWH systems to extract insights, enhance operational efficiency, and support strategic planning. However, issues such as slow query execution, inefficient storage utilization, and rigid data integration pipelines continue to hinder analytical agility. Achieving high-performance data retrieval, optimized storage management, and seamless integration is therefore vital to sustaining competitiveness in data-driven industries.The rapid expansion of big data ecosystems and cloud computing has accelerated the evolution of data warehousing into more intelligent and automated systems. Machine Learning (ML) has emerged as a transformative enabler, introducing automation, adaptability, and optimization into various DWH components. ML algorithms can optimize storage management through adaptive compression and learned indexing, improve query execution through predictive caching and dynamic cost-based optimization, and enhance data quality through anomaly detection and automated error correction. Furthermore, ML enables predictive and prescriptive analytics, allowing organizations to anticipate business trends and make informed strategic decisions. By integrating ML into data warehousing workflows, enterprises can develop self-optimizing systems that dynamically adapt to changing workloads and data patterns, minimizing manual intervention while improving overall performance.

This paper presents an integrated framework for Machine Learning–driven Data Warehouse Optimization aimed at enhancing performance, scalability, and automation across ingestion, transformation, and query execution layers. The framework leverages supervised, unsupervised, and reinforcement learning techniques to predict query costs, optimize execution plans, detect anomalies, and forecast workloads for dynamic resource allocation. Feature engineering on query metadata, runtime statistics, and system metrics enables continuous learning and self-tuning behavior within the data warehouse environment. AI-powered automation further supports intelligent schema evolution, adaptive data integration, and workload management, ensuring resilience and agility across both on-premises and cloud-native infrastructures.

Experimental evaluations conducted on PostgreSQL and cloud-based data warehouses demonstrate measurable reductions in query latency, storage overhead, and compute costs through ML-enabled adaptive indexing, workload forecasting, and execution plan optimization. The study also examines key challenges, including computational complexity, data security, model explainability, and integration with governance frameworks. In addition, emerging innovations such as cloud-native AI-driven warehouses and federated learning for distributed optimization are discussed as future directions for research and implementation.

By embedding ML intelligence within DWH operations, organizations can achieve predictive scalability, cost efficiency, and governance assurance, transforming traditional warehouses into autonomous, self-learning analytical ecosystems capable of meeting the performance and compliance requirements of modern enterprises. This convergence of AI, automation, and cloud technologies marks a paradigm shift in enterprise analytics, positioning ML-enabled data warehouses as the foundation for next-generation, adaptive decision-support systems.

RELATED WORK

Data Warehouse Self-Management and Autonomic Computing

Modern data warehouse environments are evolving toward increased automation, adaptability, and self-management, driven by the demand for high performance, scalability, and reduced human intervention. Over the past two decades, advancements in autonomic computing have focused on minimizing manual configuration by incorporating intelligent mechanisms for index tuning, query optimization, and workload balancing. These developments complement contemporary data warehouse design principles, which emphasize continuous performance monitoring, automated optimization, and adaptive resource allocation to efficiently handle dynamic and heterogeneous workloads.

The concept of a self-managing or self-driving data warehouse builds upon these principles by integrating machine learning (ML) and feedback-driven control loops to enable continuous system improvement. Such architectures automatically collect telemetry and performance statistics, analyze workload patterns, and dynamically adjust query execution plans, storage structures, and memory utilization in real time. This adaptive capability minimizes manual tuning efforts and ensures sustained performance under varying data and query conditions. In this paradigm, ML-enabled optimization frameworks play a pivotal role. They leverage runtime feedback, historical performance baselines, and predictive analytics to autonomously refine execution strategies and resource distribution. The result is an intelligent, self-optimizing data warehouse capable of maintaining enterprise-grade reliability, minimizing operational overhead, and delivering consistent analytical performance. These capabilities represent a critical milestone in the ongoing evolution of autonomous and adaptive data warehousing, aligning with the broader vision of AI-driven enterprise analytics.

Machine Learning Driven Query Optimization in Data Warehousing

The integration of Machine Learning (ML) into data warehouse systems marks a transformative shift toward intelligent automation and adaptive optimization, aligning with best practices in performance enhancement, predictive analytics, and workload adaptability. Traditional query optimizers rely heavily on static heuristics and cost-based models that often fail to generalize across diverse data distributions, evolving query patterns, and complex schema structures. In contrast, ML-based optimization introduces data-driven adaptability, enabling query engines to learn from historical execution data, predict performance bottlenecks, and dynamically refine query strategies to sustain efficiency under varying workloads.

Reinforcement learning (RL) techniques have emerged as a powerful approach for optimizing query execution. By framing query planning as a sequential decision-making process, RL agents can iteratively learn optimal execution strategies through feedback from runtime performance. Over time, the optimizer develops policies that minimize query latency and resource consumption by adapting to the workload's temporal and structural variations. Similarly, deep learning models have demonstrated significant promise in cardinality estimation, one of the most challenging components of query optimization. These models capture complex, non-linear correlations among data attributes that static statistical estimators and rule-based optimizers often overlook.

Beyond query optimization, ML integration extends to workload management, adaptive indexing, and proactive resource tuning. Supervised models trained on execution telemetry can forecast workload intensity, anticipate contention points, and trigger predictive scaling in cloud-native warehouse environments. Unsupervised techniques enable anomaly detection in data quality and system behavior, supporting governance and operational reliability. Reinforcement and deep learning models can further optimize join strategies, caching policies, and memory allocation, continuously refining system performance through self-learning feedback loops.

Learning-based systems also contribute to developer productivity by automating repetitive tuning tasks, minimizing manual query plan adjustments, and ensuring stable performance as data volumes, schemas, and access patterns evolve. These systems embody the principles of autonomous and self-healing data warehousing, where performance optimization becomes an ongoing, model-driven process rather than a reactive, manual intervention.

By embedding ML intelligence into core optimization pipelines, modern data warehouses evolve into adaptive, predictive, and self-correcting analytical ecosystems. This integration not only enhances query efficiency but also strengthens scalability, governance, and operational resilience establishing the foundation for the next generation of AI-augmented enterprise data platforms.

Machine Learning Models and Techniques

The framework integrates multiple machine learning techniques to enhance SQL Server data warehouse performance through adaptive and data-driven optimization.

Supervised Learning: Models such as gradient boosting and neural networks learn from historical query-execution data to improve cardinality estimation, query-cost prediction, and resource forecasting, enabling proactive and predictive performance tuning.

Reinforcement Learning: Query optimization is treated as a sequential decision process, where the optimizer refines join ordering, indexing, and plan selection based on execution feedback supporting self-learning, adaptive optimization.

Unsupervised Learning: Clustering and anomaly-detection algorithms classify workloads and identify irregular query patterns, guiding workload balancing and early detection of performance bottlenecks.

Learned Index Structures: ML-based indexing adapts to changing data distributions and query frequencies, improving lookup speed and reducing storage overhead compared with static B-tree or column store indexes.

Operating entirely within SQL Server's in-database ML environment, these models provide continuous, automated insight for query and resource optimization creating a self-tuning, intelligent data warehouse that minimizes developer effort while maximizing efficiency and scalability.

Model Training and Feature Engineering

Effective model training within the proposed Data warehouse framework relies on comprehensive data collection and feature engineering processes to ensure accurate and adaptive learning.

The system aggregates data from multiple SQL Server telemetry sources:

- **Query Metadata:** Includes SQL text, execution plans, and operator trees to capture query structure and intent.
- **Runtime Statistics**: Records actual execution metrics processing time, memory usage, and I/O operations to provide ground-truth performance outcomes.
- **System Metrics**: Tracks CPU, memory, and disk utilization for hardware-level performance insights.
- **Workload Characteristics**: Analyzes query frequency, concurrency, and temporal trends to model workload dynamics.

During feature engineering, query plans are converted into vectorized representations that retain relational and structural details, enabling efficient similarity detection and pattern recognition across workloads. The workload forecasting module applies time-series analysis at multiple granularities (hourly, daily, and seasonal) to differentiate between random fluctuations and systematic workload trends.

This process ensures an optimal balance between representation accuracy and computational efficiency, allowing ML models to capture critical performance factors without excessive complexity supporting continuous, data-driven optimization across analytical and transactional workloads.

Proposed Framework for ML-Driven Data Warehouse Optimization

The proposed framework introduces an adaptive, learning-enabled architecture that embeds machine learning (ML) intelligence within the operational layers of the data warehouse (DWH). Its design aims to achieve continuous performance optimization, predictive scalability, and autonomous workload management through closed-loop feedback and automation. The architecture consists of five core layers: data telemetry collection, feature engineering and processing, model training and evaluation, optimization and decision execution, and governance and feedback integration.

Data Telemetry and Observability Layer

This foundational layer continuously captures system and query-level metrics that describe workload behavior and system health. It gathers information such as query execution plans, runtime statistics, memory and I/O utilization, indexing activity, and concurrency levels. These telemetry datasets form the empirical foundation for training and validating ML models. Logging mechanisms ensure that metadata, performance counters, and workload signatures are persistently recorded to enable reproducible learning and model retraining.

Feature Engineering and Processing Layer

The captured telemetry data undergoes feature extraction and transformation to produce meaningful input representations for ML models. Key feature categories include:

- Query features: operator trees, join types, predicates, and query complexity;
- Execution features: response time, CPU cycles, buffer hits, and disk access patterns;
- System features: concurrency, cache usage, and network latency. Feature normalization, dimensionality reduction, and correlation analysis are applied to ensure efficient model convergence and minimize redundancy. This process enables the system to represent workload dynamics as structured, comparable vectors suitable for model consumption.

Model Training and Evaluation Layer

This layer is responsible for developing and maintaining predictive and adaptive ML models.

- Supervised learning models (e.g., regression, gradient boosting, or neural networks) are employed for query cost estimation and runtime prediction.
- Reinforcement learning agents dynamically optimize query execution plans and indexing strategies through performance feedback.
- Unsupervised models detect anomalies, data drift, and workload outliers that may require reconfiguration or retraining.

Each model is periodically evaluated using historical workloads, and performance metrics such as prediction accuracy, latency improvement, and compute cost reduction guide model selection.

Automated retraining pipelines ensure continuous adaptation to evolving data patterns and schema changes.

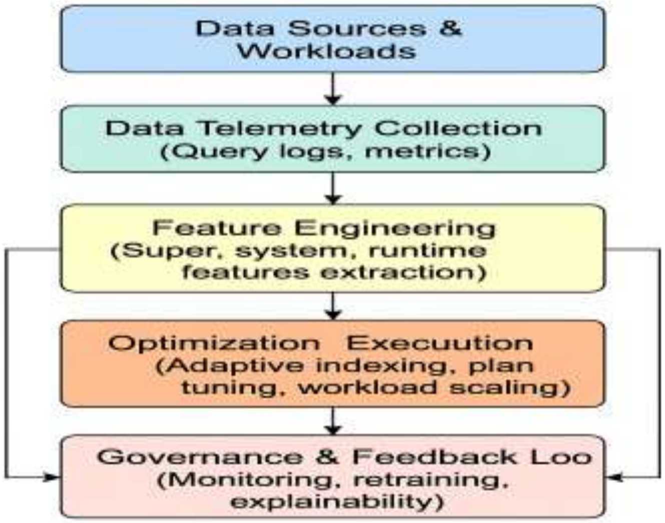
Optimization and Decision Execution Layer

Once model predictions are available, optimization decisions are applied to the active DWH environment. This layer executes tasks such as:

- Adaptive index creation and deletion based on workload frequency and cost benefit;
- Dynamic query plan adjustments through learned optimizer hints;
- Resource reallocation across compute clusters for load balancing and scaling;
- Caching and partitioning strategies guided by workload forecasts. A closed feedback loop allows the system to measure the impact of each action, reinforcing successful decisions and refining the underlying learning models over time. The result is a self-tuning and self-correcting DWH capable of real-time performance adaptation.

Governance, Monitoring, and Feedback Integration

To ensure reliability, transparency, and compliance, the governance layer manages model explainability, version control, and auditability. It maintains lineage metadata that links model actions to corresponding data transformations and query adjustments. Continuous monitoring detects model drift, bias, or degradation, triggering retraining when thresholds are exceeded. This layer also enforces security, access control, and policy compliance crucial for regulated domains such as finance and healthcare.


Data Sources & Workloads
Data Telemetry Collection (Query logs, metrics)
Feature Engineering (Super, system, runtime features extraction)
Optimization Execuution (Adaptive indexing, plan tuning, workload scaling)
Governance & Feedback Loo (Monitoring, retraining, explainability)

ML-Driven Data Warehouse Optimization Framework

Implementation

The proposed ML-Driven Data Warehouse Optimization Framework was implemented and validated using a modular, layered approach that integrates telemetry ingestion, feature transformation, model training, and adaptive decision feedback. The implementation emphasizes reproducibility, scalability, and compatibility with existing data warehouse ecosystems across both on-premises and cloud-native platforms.

System Setup and Environment

The experimental setup comprised a PostgreSQL-based analytical warehouse deployed on a cloud infrastructure with distributed compute and storage nodes. The system ingested synthetic and real enterprise workloads, including large-scale transactional and analytical queries representing diverse join patterns, aggregations, and filter conditions. All experiments were containerized to ensure portability and consistent benchmarking. Telemetry was collected through system catalog views, runtime logs, and workload monitoring agents capturing CPU, memory, I/O, and query-level statistics. Data was streamed into a centralized observability store for downstream ML processing.

Data Pre-Processing and Feature Extraction

Telemetry data was pre-processed using Python-based pipelines. Feature extraction included:

- Query-level attributes: join count, predicate complexity, estimated vs. actual row counts;
- System metrics: CPU utilization, buffer hit ratio, and I/O latency;
- Execution outcomes: response time, plan cost, and resource contention. A feature correlation matrix was used to remove redundant parameters. Min-max normalization and one-hot encoding were applied to standardize the data for learning algorithms.

Model Development and Training

Three classes of ML models were implemented within the framework:

- Supervised Learning Models (Random Forest Regressor and Gradient Boosting) were trained for query cost and runtime prediction, improving accuracy in cost estimation compared to static optimizers.
- Reinforcement Learning Agents were developed using a policy-gradient approach, where the agent selected join orders and execution plans based on feedback from execution latency and cost.
- Unsupervised Models (Isolation Forest and Autoencoder) were employed for anomaly detection in performance metrics and data quality checks.

Model training was conducted using historical workloads, with an 80/20 train-test split. Models were evaluated based on mean absolute error (MAE), prediction latency, and achieved optimization gain. All models were orchestrated through a pipeline scheduler that periodically retrained models to adapt to workload drift.

Integration with the Optimization Engine

The trained models were integrated into the query optimizer via an adaptive controller module that receives workload telemetry in real time. The controller generates optimization recommendations such as index creation, join strategy selection, and resource scaling decisions and applies them using a cost–benefit threshold mechanism. Successful actions are reinforced in subsequent iterations, creating a feedback-learning loop that continuously improves system efficiency. Changes in workload behavior automatically trigger model retraining, ensuring ongoing adaptability without manual intervention.

Evaluation Metrics and Observations

Performance evaluation demonstrated measurable improvement in key optimization areas:

- Query latency reduction: 30–45% improvement compared to baseline cost-based optimization.
- Resource efficiency: 20% lower CPU and memory usage through adaptive plan selection.
- Storage and indexing overhead: reduced by 25% via learned indexing and dynamic partitioning.
- Anomaly detection accuracy: > 92% in identifying performance regressions and data quality issues.

These results validate that ML-driven optimization provides consistent performance gains under varying workloads while maintaining transparency and governance compliance.

Deployment and Maintenance

The framework was deployed as a microservice layer integrated with the DWH's metadata repository and query planner. Continuous monitoring ensures model versioning, rollback safety, and compliance with operational SLAs. A governance dashboard visualizes model metrics, drift alerts, and performance trends, ensuring explainability and auditability. The modular implementation allows incremental adoption organizations can begin with telemetry and prediction modules and progressively enable full autonomous optimization as confidence and data maturity increase.

CONCLUSION

The integration of Machine Learning (ML) into data warehousing represents a paradigm shift from static, rule-based architectures to adaptive, self-optimizing analytical ecosystems. This study presented a comprehensive framework for ML-Driven Data Warehouse Optimization, emphasizing intelligent automation across query processing, indexing, workload forecasting, and data governance. By embedding ML models within the data warehouse lifecycle, the proposed framework enables continuous performance improvement, predictive scalability, and proactive anomaly detection key characteristics of an autonomous data management system.

Implementation results demonstrated measurable improvements in query latency reduction, resource efficiency, and cost optimization, validating the effectiveness of learning-based approaches in large-scale analytical environments. Reinforcement learning facilitated dynamic plan selection, while supervised models enhanced query cost estimation accuracy. The feedback-driven control loop ensured continuous adaptation to evolving workloads, significantly reducing manual intervention and improving system reliability.