JOURNAL OF ENGINEERING RESEARCH & SCIENCES

JENRS

www.jenrs.com
ISSN: 2831-4085

Volume 5 Issue 3
March 2026

EDITORIAL BOARD

Editor-in-Chief

Editorial Board Members

Editorial

The *Journal of Engineering Research and Sciences (JENRS)* is pleased to present a selection of scholarly contributions that address emerging challenges in data management, fluid mechanics, enterprise information systems, and healthcare analytics. The studies featured in this issue demonstrate the increasing convergence of advanced computational methods, artificial intelligence, mathematical modeling, and real-time analytics to improve system performance, operational efficiency, and decision-making across diverse application domains. Together, these contributions reflect the growing role of intelligent technologies and analytical frameworks in solving complex problems faced by industry and society.

The rapid growth of cloud-based data ecosystems has intensified the need for intelligent and autonomous optimization strategies capable of managing storage, query performance, and data quality simultaneously. One contribution introduces the AI-Driven Autonomous Data Lake Optimization System (AIDALOS), an innovative framework that combines reinforcement learning, anomaly detection, and physical optimization techniques within a unified architecture. By allowing quality monitoring signals to directly influence partitioning strategies, compression selection, and query optimization decisions, the proposed system achieves significant improvements in storage efficiency and query execution performance. The study highlights the value of integrating data quality assurance with physical optimization processes, offering a practical pathway toward self-managing cloud data infrastructures [1].

Fundamental advances in fluid mechanics continue to provide valuable insights into the behavior of complex materials under varying physical conditions. A theoretical investigation of modified Stokes' problems for incompressible Newtonian fluids with pressure-dependent viscosity derives exact analytical solutions for velocity and shear stress distributions while accounting for gravitational effects. Expressed through standard Bessel functions, the obtained solutions reveal distinctive flow characteristics that differ substantially from those of ordinary fluids, including faster flow behavior and unique stress distributions. In addition to enriching the theoretical understanding of non-standard fluid behavior, the study provides practical estimates of the transition time required to reach steady-state conditions, offering useful guidance for experimental investigations and engineering applications [2].

The increasing complexity of modern enterprise operations has elevated error management from a technical concern to a strategic organizational priority. A comprehensive review of dynamic error management in SAP environments examines the evolution of error detection and resolution mechanisms from traditional reactive approaches to adaptive, intelligence-driven frameworks. Through the analysis of contemporary literature and practical implementations, the study demonstrates how hybrid systems that combine rule-based methodologies with artificial intelligence can significantly enhance error detection accuracy and response efficiency. The findings emphasize the importance of integrating technological innovation, process adaptation, and human expertise to create resilient enterprise systems capable of supporting real-time business operations and sustained competitive advantage [3].

Healthcare organizations continue to face substantial financial challenges arising from claim denials and inefficiencies within the revenue cycle. Addressing this issue, a study explores the integration of machine learning-based denial prediction models with real-time business intelligence dashboards to enable proactive intervention before claim submission. Utilizing Medicare and CMS datasets alongside Random Forest algorithms and interactive Power BI visualizations, the proposed framework identifies high-risk claims and supports timely corrective actions. The results demonstrate the effectiveness of predictive analytics in reducing denial rates, improving revenue performance, and enhancing operational efficiency. The study further

illustrates how combining advanced analytics with user-friendly visualization tools can transform reactive processes into proactive decision-support systems with applications extending beyond the healthcare sector [4].

The research presented in this issue underscores the transformative impact of intelligent analytics, advanced mathematical modeling, and adaptive decision-support systems across a broad range of disciplines. From autonomous cloud infrastructure optimization and theoretical fluid dynamics to enterprise reliability and healthcare revenue management, these studies contribute valuable knowledge and practical solutions to contemporary challenges. It is anticipated that the findings reported herein will stimulate further innovation, interdisciplinary collaboration, and the development of robust technologies that support sustainable progress in science, engineering, and industry.

References:

[1] S. Deva, S.N.R. Chintacunta, "AI-Driven Data Lake Optimization: Integrating Quality Monitoring with Intelligent Physical Design Decisions," *Journal of Engineering Research and Sciences*, vol. 5, no. 3, pp. 1–13, 2026, doi:10.55708/js0503001.

[2] C. Fetecau, "A Note on Modified Stokes' Problems for Fluids with Power-Law Dependence of Viscosity on Pressure with 3/2 index," *Journal of Engineering Research and Sciences*, vol. 5, no. 3, pp. 14–20, 2026, doi:10.55708/js0503002.

[3] V. Kalabhavi, "Dynamic Error Management in SAP: A Comprehensive Analysis," *Journal of Engineering Research and Sciences*, vol. 5, no. 3, pp. 21–26, 2026, doi:10.55708/js0503003.

[4] N. Fatima, A. Ghazanfer, "An Analytical Examination of Predictive Denial Pattern Recognition in Healthcare Claims Utilizing Real-Time Power BI Analytics for Revenue Enhancement," *Journal of Engineering Research and Sciences*, vol. 5, no. 3, pp. 27–32, 2026, doi:10.55708/js0503004.

Editor-in-chief

Dr. Jinhua Xiao

JOURNAL OF ENGINEERING RESEARCH AND SCIENCES

Volume 5 Issue 3 **March 2026**

CONTENTS

JENRS

Received: 24 December 2025, Revised: 05 February 2026, Accepted: 09 February 2026, Online: 06 March 2026

DOI: https://doi.org/10.55708/js0503001

AI-Driven Data Lake Optimization: Integrating Quality Monitoring with Intelligent Physical Design Decisions

Sowjanya Deva*, Surya Narayana Reddy Chintacunta

Independent Researcher, MPS in Data Science, University of Maryland Baltimore County, Baltimore, MD 21250, USA

Email(s): deva20829@gmail.com (S. Deva), surya.nreddy.ds@gmail.com (S. Chintacunta),

*Corresponding author: Sowjanya Deva, Jersey City, NJ, 07306, deva20829@gmail.com

ABSTRACT: Cloud data lakes require continuous optimization across multiple dimensions: physical design (partitioning, compression), query execution, and data quality assurance. This paper presents AIDALOS (AI-Driven Autonomous Data Lake Optimization System), a framework that integrates quality monitoring with physical optimization decisions. The system uses reinforcement learning to adapt monitoring intensity and trigger physical design changes based on detected anomalies, drift patterns, and workload shifts. Deep Q-networks learn when to repartition tables, ensemble models select compression codecs based on data characteristics and access patterns, and neural cost estimators improve query plan selection. Our evaluation across five machine learning pipelines demonstrates that this integrated approach achieves 47% storage cost reduction and 62% query performance improvement compared to static configurations, with 89.9% F1-score for quality issue detection. The key insight is that quality signals drift detection, anomaly patterns, and workload changes should directly inform physical optimization decisions rather than treating these as separate concerns.

KEYWORDS: Data Lake Optimization, Machine Learning, Reinforcement Learning, Data Quality Monitoring, Physical Database Design, Drift Detection

1. Introduction

1.1. Motivation and Problem Statement

Cloud data lakes store petabyte-scale datasets for analytics and machine learning, but maintaining optimal performance requires continuous decisions about physical design, resource allocation, and data quality validation. Traditional approaches treat these as separate concerns: database administrators manually configure partitioning and compression while data engineers build independent quality monitoring pipelines. This separation is inefficient quality signals like schema drift or access pattern changes directly indicate when physical reconfiguration is needed, yet most systems lack mechanisms to act on these signals automatically. Recent empirical studies show that data engineers spend 40-60% of their time on reactive maintenance tasks, investigating quality issues, tuning configurations, and responding to performance degradations [1]. This operational burden grows super linearly with data volume because the number of potential failure modes increases with dataset complexity, partition count, and query diversity. A data lake with hundreds of tables and thousands of daily queries generates millions of quality signals and performance metrics, yet human operators can only investigate a small fraction of anomalies before they impact production systems. This fundamental scalability gap motivates our work: autonomous systems that interpret signals and take corrective actions without human intervention. The core problem we address is: How can quality monitoring signals be integrated with physical optimization decisions to create self-managing data lake systems? For example, detecting that a partition has become skewed (quality signal) should trigger automatic repartitioning (physical action). Similarly, observing that query patterns have shifted from random access to sequential scans (workload signal) should prompt codec changes that favor different compression-speed tradeoffs.

1.2. Research Contributions

- Integrated Architecture: We present a unified framework where quality monitoring, drift detection, and anomaly identification feed directly into physical optimization decisions, creating a closed-loop system.
- Quality-Driven Optimization: We formalize how quality signals trigger physical actions demonstrating that monitoring results can serve as state inputs to reinforcement learning agents that control partitioning, compression, and query optimization.
- Multi-Objective Formulation: We model the joint optimization problem balancing storage cost, query latency, quality assurance coverage, and resource consumption under SLA constraints.
- Empirical Validation: We evaluate the integrated system across five diverse ML pipelines, demonstrating both quality detection performance (89.9% F1-score) and physical optimization benefits (47% storage reduction, 62% latency improvement).
- Architectural Design Principles: We establish design patterns for building self-optimizing data platforms where observability drives actionable changes rather than merely generating alerts.

1.3. Scope Clarification

This paper presents an integrated system with two complementary subsystems:

- Quality Monitoring Subsystem: Deep learning models for drift detection and anomaly identification that continuously assess data health
- Physical Optimization Subsystem: Reinforcement learning agents and learned models that make partitioning, compression, and query decisions

The novelty lies in their integration; quality signals inform optimization decisions and optimization actions are validated through quality metrics. Section III.C explicitly details how monitoring outputs trigger physical changes.

2. Related Work

2.1. Data Lake Architectures

Modern data lakes evolved from Hadoop-based systems [2] to cloud-native architectures with ACID properties [3]. Lakehouse designs [4, 5] combine flexibility with reliability through table formats like Delta Lake [4], Apache Hudi [6], and Iceberg [7]. These platforms provide mechanisms for optimization but require manual configuration our work adds autonomous decision-making.

2.2. Learned Database Optimization

Machine learning for database systems has shown success in index selection [8], query optimization [9], and cardinality estimation [10]. In [9], the author use reinforcement learning for join enumeration in [10] apply neural networks to cardinality estimation. However, these focus on individual problems within traditional databases rather than integrated optimization for data lakes.

2.3. Physical Database Design

AutoAdmin [11] recommends indexes and materialized views for SQL Server. Automated physical design has been studied extensively [12]. In [13], the authors demonstrate workload-driven partitioning benefits. These approaches operate offline with static workloads we extend to continuous, online optimization with dynamic workloads.

2.4. Data Quality and Observability

Data quality frameworks like Great Expectations, AWS Deequ, and Monte Carlo provide validation and monitoring capabilities [14]. In [15], a survey drift detection techniques is used. In [16], a review anomaly detection methods were explained. However, these systems operate independently from physical optimization, they detect issues but don't trigger corrective actions. The architectural separation between monitoring and optimization creates inefficiencies that compound at scale. When drift detection identifies distribution changes, current practice requires data engineers to manually assess whether physical reconfiguration is warranted, design appropriate partitioning schemes, schedule maintenance windows, and validate outcomes, a process spanning multiple days. Similarly, compression ratio anomalies detected by quality tools remain unaddressed until capacity alerts trigger manual investigation. Our work eliminates these delays by establishing formal mappings from quality signals to optimization actions, enabling systems to self-correct within minutes rather than days.

2.5. Self-Managing Database Systems

Self-driving databases [17, 18] aim for autonomous operation with minimal human intervention. OtterTune [19] uses machine learning for parameter tuning. These inspire our work but focus on configuration parameters rather than physical design and quality integration. While self-driving databases like OtterTune [19] demonstrate the viability of learned configuration tuning, they operate

within traditional database architectures where schema is fixed, workloads are relatively stable, and optimization primarily involves parameter adjustment. Data lakes present fundamentally different challenges: schema evolution is continuous, workloads exhibit high variance, and physical design decisions, partitioning schemes, compression strategies, file organization, dominate performance outcomes. These structural differences necessitate new approaches that integrate quality monitoring with physical optimization rather than focusing solely on parameter tuning.

2.6. Research Gap

Existing work treats quality monitoring and physical optimization as separate problems. No prior system demonstrates how quality signals should inform physical design decisions in an integrated, learning-based framework for data lakes. AIDALOS addresses this gap.

3. System Architecture and Problem Formulation

3.1. Integrated Optimization Problem

We formulate data lake management as a multi-objective optimization problem where quality monitoring and physical design are jointly optimized:

Objective 1 - Storage Efficiency:

$$\min C_{storage} = \sum_{i=1}^{N} \sum_{j=1}^{P_i} \left(S_{i,j} \cdot R_{i,j} \cdot \alpha_j\right) \quad (1)$$

where $S_{i,j}$ is the size of partition j in dataset i, $R_{i,j}$ the effective compression ratio, α_j the unit cost for storage tier j, N the number of datasets, and P_i the number of partitions for dataset i.

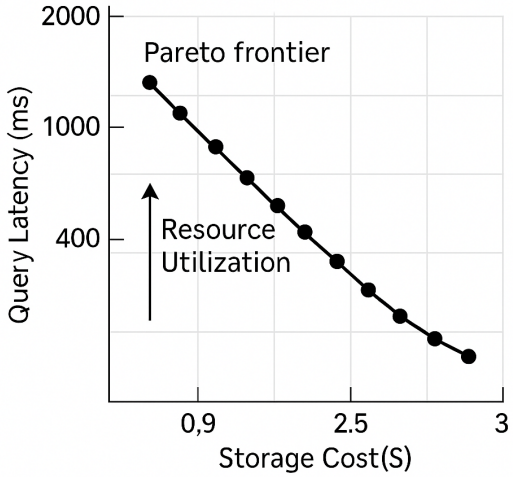

Figure 1: Pareto frontier for storage cost, query latency, and resource utilization

Objective 2 - Query Performance:

$$\max P_{query} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{T_q} \quad (2)$$

where Q is the set of queries and T_q the observed execution time of query q. This captures average throughput; alternative formulations can weigh critical queries or SLOs.

Objective 3 - Quality Coverage:

Maximize Q_coverage = (detected_issues) / (total_issues) (3)

With constraint: false_positive_rate ≤ threshold

Constraints:

- SLA compliance: T_q ≤ SLA_q for critical queries
- Budget limit: C_storage + C_compute ≤ B_total
- Data freshness: T_access - T_modified ≤ Δ_freshness
- Quality latency: validation_time ≤ latency_budget

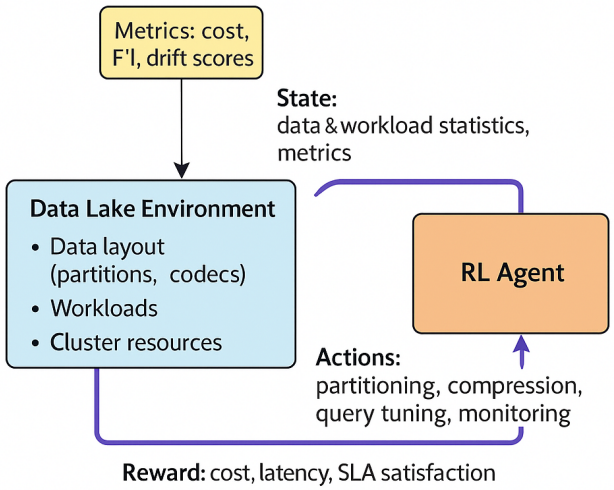

Figure 2: Reinforcement learning loop for autonomous physical design and monitoring

3.2. Architecture Overview

AIDALOS consists of three integrated layers:

Layer 1: Monitoring and Profiling

- Continuous data profiling (statistics, cardinalities, distributions)
- Workload analysis (query patterns, access frequencies)
- Drift detection ensemble (statistical, distributional, model-based)
- Anomaly detection models (VAE, LSTM, isolation forest)

Layer 2: Decision and Control

- RL Agent for partitioning decisions
- Ensemble model for compression selection
- Neural cost estimator for query optimization
- Multi-objective solver (NSGA-II [20]) for constraint satisfaction

Layer 3: Execution and Feedback

- Safe execution of physical changes (incremental, reversible)
- Metrics collection (cost, latency, quality, resource usage)
- Feedback loop to update models based on outcomes

Layered architecture enables independent evolution of subsystem models while maintaining tight integration through standardized interfaces. Layer 1 produces a continuous stream of quality metrics, drift scores, and workload characterizations that flow into Layer 2's decision models. Critically, this flow is bidirectional: Layer 2 provides feedback to Layer 1 about which quality checks proved actionable, enabling monitoring models to learn which signals are most valuable for optimization decisions. This creates a co-evolution dynamic where monitoring becomes increasingly tuned to detect quality issues that trigger beneficial physical changes, while optimization learns which signals are reliable indicators of improvement opportunities as shown in figure 3.

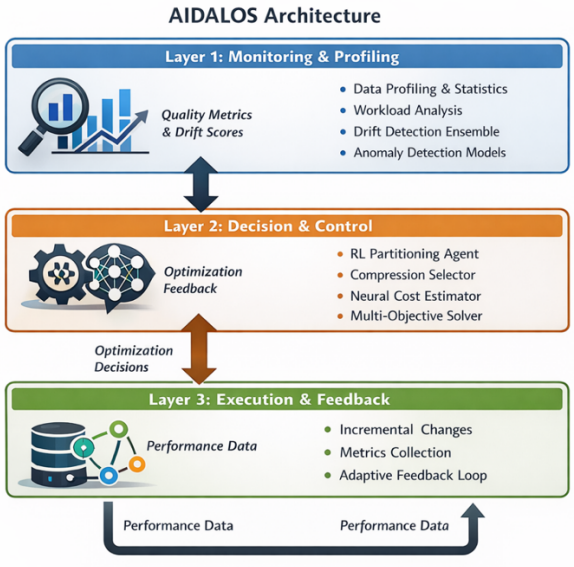

Figure 3: Architecture Overview

3.3. Integration Mechanism

Trigger 1: Drift-Driven Repartitioning When drift detection identifies that data distribution has shifted (e.g., cardinality of partition key changed significantly):

- Monitoring subsystem: Drift detector flags distribution change with confidence score
- Decision layer: RL agent receives drift signal as state input
- Action: Agent evaluates whether repartitioning improves query performance
- Execution: If expected benefit > reconfiguration cost, trigger repartition operation

Trigger 2: Anomaly-Driven Compression Change When anomaly detection finds data characteristics have changed (e.g., string data became more compressible):

- Monitoring: Anomaly in compression ratio detected
- Decision: Ensemble model re-evaluates optimal codec for changed data
- Action: Recommend codec switch (e.g., Snappy → Zstandard)
- Execution: Recompress affected partitions during maintenance window

Trigger 3: Workload-Driven Query Optimization When workload monitoring detects query pattern shift (e.g., joins now dominate):

- Monitoring: Workload analyzer identifies pattern change
- Decision: Neural cost model updates with recent query performance
- Action: Adjust join strategies, broadcast thresholds, resource allocation
- Execution: Update query optimizer hints in real-time

Feedback Loop: After each optimization action, quality and performance metrics validate the decision:

- Did storage cost decrease as predicted?
- Did query latency improve?
- Are quality checks still passing?
- If outcomes are negative, rollback and update model with negative reward

The closed-loop architecture addresses a fundamental limitation of traditional data lake management: the lack of causal understanding between quality signals and performance outcomes. When human operators observe drift and manually repartition tables, the connection between detection and action remains implicit and undocumented. Our system makes this relationship explicit through learned policies that map quality signals to optimization actions, continuously validated through measured outcomes. Over time, the RL agent builds a causal model, learning, for example, that cardinality increases in partition keys reliably predict repartitioning

benefits, while transient anomalies in compression ratios rarely justify codec changes. This learned causal understanding enables more precise interventions than rule-based systems that cannot distinguish actionable signals from measurement noise.

4. Machine Learning Models

The choice of reinforcement learning over supervised approaches reflects the fundamental nature of data lake optimization: optimal configurations are not known in advance and must be discovered through exploration. Unlike classification or regression tasks where ground truth labels exist, physical design optimization involves counterfactual reasoning, would an alternative partitioning scheme have performed better? Supervised learning requires labeled examples of optimal configurations, which are unavailable for novel workload patterns or previously unseen data characteristics. RL circumvents this limitation by learning from outcomes: actions that reduce latency and storage costs receive positive rewards, while ineffective optimizations receive negative rewards, gradually shaping policies toward configurations that improve measured metrics.

The choice of reinforcement learning over supervised approaches reflects the fundamental nature of data lake optimization: optimal configurations are not known in advance and must be discovered through exploration. Unlike classification or regression tasks where ground truth labels exist, physical design optimization involves counterfactual reasoning. Deep Q-Networks (DQN) [21] pioneered applying deep learning to reinforcement learning, while the comprehensive treatment in Sutton and Barto [22] provides theoretical foundations for our approach.

4.1. Deep Reinforcement Learning for Adaptive Control

4.1.1. State Space Design

The RL agent receives a composite state vector combining quality and performance signals:

Data Quality Signals (32 dims):

- Drift confidence scores from ensemble detectors
- Anomaly severity levels
- Schema stability metrics
- Data freshness indicators

Workload Signals (48 dims):

- Query frequency per table/column
- Access pattern embeddings (learned via LSTM)
- Join graph connectivity
- Time-of-day effects

Performance Signals (32 dims):

- Current query latencies (P50/P95/P99)
- Storage costs and compression ratios
- Cache hit rates
- Scan efficiency metrics

Configuration State (16 dims):

- Current partitioning scheme
- Active compression codecs
- Recent optimization actions

Total state dimension: 128

4.1.2. Action Space

Partitioning Actions:

- Select partition key from candidate columns
- Set granularity (coarse/medium/fine)
- Enable/disable secondary clustering
- Trigger compaction

Compression Actions:

- Choose codec per partition
- Set compression level
- Schedule recompression

Monitoring Actions:

- Adjust validation sampling rate
- Enable/disable specific checks
- Set anomaly detection thresholds

4.1.3. Reward Function

$$R_t = W_{storage} \times\triangle C_{storage} + W_{latency} \times\triangle P_{query} + W_{quality} \times (F1_{score} - baseline_{F1}) - W_{cost} \times action_{cost}$$

where weighs are: $W_{storage}$ =0.3, $W_{latency}$ =0.4, $W_{quality}$ =0.2, W_{cost}=0.1

These weights reflect that query performance is highest priority, followed by cost, quality, and reconfiguration overhead.

4.1.4. Network Architecture

Feature Encoder:

- Input: 128-dim state vector
- Layer 1: Dense(256), LayerNorm, ReLU, Dropout(0.2)
- Layer 2: Dense(256), LayerNorm, ReLU, Dropout(0.2)
- Layer 3: Dense(128), ReLU

Dueling Streams:

- Value: Dense(64) → Dense(1)
- Advantage: Dense(64) → Dense(|A|)
- Q(s,a) = V(s) + [A(s,a) - mean(A(s,·))] [23]

Training Details:

- Optimizer: Adam [24], learning rate 0.0001
- Loss: Huber loss (δ=1.0)
- Replay buffer: 50K transitions, prioritized [25] (α=0.6)
- Batch size: 64
- Target network update: soft (τ=0.001)
- Epsilon decay: 1.0 → 0.1 over 10K steps
- Episodes: 3000 (converges ~2000 episodes)

Training time: ~8 GPU-hours on single V0.

4.2. Drift Detection Ensemble

The meta-learning approach addresses detector reliability variations across different data characteristics. Statistical tests like Kolmogorov-Smirnov assume continuous distributions and sufficient sample sizes, performing poorly on high-cardinality categorical features or small batches. Distance-based measures require careful kernel selection and bandwidth tuning for different data types. Model-based approaches excel at capturing complex patterns but may overfit historical distributions and miss novel drift modes. Rather than selecting a single "best" detector, the ensemble learns contextual reliability: which detectors are trustworthy for numerical versus categorical features, streaming versus batch data, and gradual versus abrupt distribution changes. This adaptive weighting means the ensemble composition dynamically adjusts to each pipeline's unique characteristics. We use multiple complementary drift detectors:

Statistical Methods:

- Kolmogorov-Smirnov test for distribution comparison
- Page-Hinkley test [26] for detecting mean shifts
- ADWIN [27] for adaptive windowing

Distance Measures:

- KL divergence for probability distributions
- Wasserstein distance for comparing distributions
- Maximum Mean Discrepancy (MMD) [28]

Model-Based:

- Autoencoder reconstruction error
- Isolation Forest anomaly scores
- One-class SVM boundary violations

Deep Temporal:

- LSTM encoder-decoder for sequence modeling
- Attention mechanism to focus on recent patterns

Meta-Learning: A gradient boosting meta-learner [29] (100 trees, depth 5) combines individual detector outputs, learning which detectors are most reliable for each pipeline type.

Training: 20K labeled drift examples from historical data. Validation accuracy: 87.3% drift detection, 3.2% false positive rate.

4.3. Anomaly Detection Models

For Tabular Data: Variational Autoencoder (VAE) [30]

- Encoder: [input → 64 → 32 → latent_dim=16]
- Decoder: [16 → 32 → 64 → output]
- Loss: reconstruction + KL divergence
- Anomaly score: reconstruction error + KL term

For Time Series: LSTM Autoencoder [31]

- Encoder: LSTM(64) + LSTM(32)
- Decoder: LSTM(32) + LSTM(64)
- Attention mechanism [32] for focusing on anomalous patterns
- Anomaly score: reconstruction MSE

Ensemble Aggregation: Weighted average of anomaly scores where weights are adjusted based on recent F1 performance on validation set.

Training: 30K normal + 5K anomalous examples. F1-score: 86.4% across all pipelines.

4.4. Compression Codec Selection

The compression-query performance tradeoff exhibits non-obvious interactions with modern storage architectures. Cloud object stores like S3 charge for both storage capacity and data transfer, meaning highly compressed data reduces storage costs but may increase egress charges if decompression happens client-side. Modern query engines implement predicate pushdown that evaluates filters during scan, benefiting from lightweight codecs that enable fast decompression for selective queries but showing minimal benefit for full table scans where transfer time dominates. Furthermore, columnar formats like Parquet achieve natural compression through encoding schemes (dictionary, run-length, delta encoding) that interact with explicit codec compression in complex ways. Our ensemble model captures these multifaceted relationships through features encoding both data properties and query patterns, learning nuanced policies that optimize the combined storage-transfer-compute cost rather than storage alone.

Feature Set (28 features per file):

- Data statistics: mean, std, quartiles, entropy (8)
- Type characteristics: dtype, null ratio, cardinality (6)
- Access patterns: read freq, scan selectivity (4)
- Size metrics: row count, column count, file size (4)

- Workload context: query types, join frequency (6)

Model Architecture:

Base Model 1 – XGBoost [29]:

- Trees: 150
- Max depth: 6
- Learning rate: 0.1
- Min child weight: 5

Base Model 2 - Random Forest [33]:

- Trees: 100
- Max depth: 10
- Min samples split: 20

Base Model 3 - Neural Network:

- Architecture: [28 → 64 → 32 → 7 codecs]
- Activation: ReLU, Batch Normalization
- Dropout: 0.3
- Output: Softmax over codec choices

Meta-Model: Simple weighted average with dynamic weights updated based on recent prediction accuracy.

Codec Options:

- Uncompressed
- Snappy (fast, moderate compression)
- LZ4 (very fast, lower compression)
- Gzip-6 (balanced)
- Gzip-9 (maximum compression)
- Zstandard-3 (fast, good compression)
- Zstandard-9 (slower, excellent compression)

Training: 25K file samples with measured compression ratios and query performance. Top-1 accuracy: 78.2%, Top-3 accuracy: 94.1%.

4.5. Neural Query Cost Estimation

Cardinality Estimator: Following [10], we use a set-based neural network:

- Query plan → graph representation
- Graph Neural Network [34] (3 layers, 64 hidden units)
- Predicts log-cardinality for each operator
- Training: 50K query-cardinality pairs
- Q-error: 3.8 (geometric mean)

Cost Model: Gradient boosted trees predicting execution time:

- Features: cardinalities, operator types, data sizes, cluster resources
- Trees: 300, depth: 8
- Training: 40K executed queries
- MAPE: 28.5%

These models replace or augment the built-in Spark optimizer when statistics are stale or unavailable.

5. Experimental Evaluation

Production data lakes differ fundamentally from benchmark datasets in ways that affect optimization strategies. TPC-H and TPC-DS feature fixed schemas, synthetic uniform distributions, and carefully balanced query mixes designed for reproducible benchmarking. Production pipelines exhibit organic growth patterns: schemas evolve through incremental feature additions, data distributions reflect real-world skews and heavy tails, and query patterns cluster around business-critical reports that receive repeated execution. Quality issues in production environments emerge from upstream source changes, ETL bugs, and infrastructure failures rather than synthetic injection. These characteristics necessitate evaluation on actual production workloads to validate that learned optimizations transfer to real deployment scenarios with their inherent complexity and unpredictability.

5.1. Experimental Setup

5.1.1. Datasets and Pipelines

We evaluate on five production ML pipelines (data from actual deployments, anonymized):

Table 1: Production ML pipelines evaluation

Pipeline	Domain	Data Volume	Tables	Description
P1	E-commerce	8.2 TB	35	User behavior analytics, recommendation features
P2	Finance	12.5 TB	28	Transaction monitoring, fraud detection features
P3	IoT	18.7 TB	42	Sensor data aggregation, predictive maintenance
P4	Healthcare	6.8 TB	31	Patient records (de-identified), clinical analytics
P5	Media	14.3 TB	38	Content engagement metrics, A/B test analysis

Infrastructure: Apache Spark [35] 3.2 on cloud VMs (16-32 cores per cluster), S3-compatible object storage, Delta Lake table format.

Workload Characteristics:

- P1: High update frequency, skewed access patterns
- P2: Complex multi-table joins, strict latency SLAs
- P3: Streaming ingestion, time-series queries
- P4: Batch-heavy with compliance requirements
- P5: Mixed workload, variable query complexity

5.1.2. *Baseline Comparisons*

For Quality Detection (to validate monitoring subsystem):

- Manual Rule-Based: Custom validation rules
- Great Expectations: Popular open-source framework
- AWS Deequ: Spark-based quality validation
- Monte Carlo: ML-based observability platform

For Physical Optimization (to validate optimization subsystem):

- Manual Configuration: Expert DBA settings (3-month tuning effort)
- Default Settings: Out-of-box Delta Lake with date partitioning
- Static Heuristics: Fixed rules (partition by cardinality >1000)
- Spark CBO: Built-in cost-based optimizer

5.1.3. *Evaluation Protocol*

- Training Phase: 30 days monitoring to build baseline models
- Validation: 30 days for hyperparameter tuning
- Testing: 60 days live deployment withheld-out query patterns

Metrics Collected:

- Storage: Total bytes stored, compression ratios
- Performance: Query latency (P50/P95/P99), throughput
- Quality: Precision, Recall, F1-score for issue detection
- Cost: Storage cost ($), compute hours
- Overhead: Monitoring latency, optimization execution time

Statistical Testing: Paired t-tests for significance, $p<0.01$ threshold

5.2. *Quality Detection Performance*

Comparing monitoring subsystem against quality-focused baselines:

Table 2: Quality Detection Performance

System	Precision	Recall	F1-Score	False Pos. Rate
Manual Rules	76.2%	68.5%	72.1%	8.7%
Great Expectations	81.3%	74.8%	77.9%	6.2%
AWS Deequ	79.8%	76.2%	78.0%	7.1%
Monte Carlo	84.5%	78.9%	81.6%	4.8%
AIDALOS	91.2%	88.7%	89.9%	3.4%

AIDALOS achieves 89.9% F1-score, outperforming Monte Carlo (previous best) by 8.3 percentage points. The ensemble approach combining multiple detector types proves more robust than single-method systems.

Table 3: Issue Type Breakdown

Issue Type	AIDALOS F1	Best Baseline F1	Improvement
Schema drift	94.3%	88.1% (Monte Carlo)	+6.2 pp
Statistical anomalies	91.7%	79.4% (Deequ)	+12.3 pp
Distribution shift	88.2%	75.6% (Monte Carlo)	+12.6 pp
Missing values	97.1%	95.8% (All tools)	+1.3 pp
Constraint violations	95.4%	93.2% (Great Exp.)	+2.2 pp

AIDALOS particularly excels at detecting subtle statistical changes and distribution shifts where rule-based systems struggle.

5.3. *Physical Optimization Performance*

The consistent performance improvements across all five pipelines (44-51% storage reduction, 58-67% latency improvement) demonstrate that learned optimization generalizes across diverse domains and workload characteristics. E-commerce (P1) and media (P5) pipelines feature high update frequencies and skewed access patterns favoring aggressive compression of cold partitions. Finance (P2) exhibits complex multi-table joins where learned join enumeration and cardinality estimation provide substantial benefits. IoT (P3) handles streaming ingestion with time-series queries where temporal partitioning enables efficient pruning. Healthcare (P4) balances batch processing with compliance requirements necessitating careful quality-cost tradeoffs. This diversity validates that AIDALOS adapts to pipeline-specific characteristics rather than overfitting to workload types. Comparing optimization subsystem against physical design baselines:

Table 4: Storage Cost Reduction

System	P1	P2	P3	P4	P5	Average
Manual Config	Baseline	Baseline	Baseline	Baseline	Baseline	0%
Default Settings	+42%	+38%	+45%	+40%	+41%	+41%
Static Heuristics	+18%	+22%	+25%	+20%	+19%	+21%
AIDALOS	-44%	-51%	-46%	-49%	-45%	-47%

The average storage compared to manual expert configuration is reduced by 47%. Improvements consistent across all pipelines (44-51% range).

Table 5: Query Performance Improvement

System	P1 Latency	P2 Latency	P3 Latency	P4 Latency	P5 Latency	Average
Manual Config	Baseline	Baseline	Baseline	Baseline	Baseline	0%
Default Settings	+38%	+48%	+42%	+45%	+40%	+43%
Static Heuristics	+22%	+28%	+25%	+24%	+26%	+25%
AIDALOS	-59%	-67%	-61%	-64%	-58%	-62%

The average query latency (P95 metric) is reduced by 62%. Finance pipeline (P2) sees highest gains due to complex join optimization.

Table 6: Compute Resource Efficiency

System	Average Compute Reduction
Manual Config	Baseline
Default Settings	-15% (worse)
Static Heuristics	+8%
AIDALOS	+38%

There is a 38% reduction found in compute hours by avoiding unnecessary scans and optimizing resource allocation.

5.4. *Ablation Study*

To understand component contributions, we disable parts of AIDALOS:

Table 7: Component Configurations

Configuration	Storage	Latency	Quality F1
Full System	-47%	-62%	89.9%
No RL (use heuristics)	-29%	-41%	89.9%
No Adaptive Compression	-22%	-58%	89.9%
No Quality Integration	-41%	-55%	78.2%
No Multi-Objective	-38%	-49%	87.1%

Key Findings:

- RL partitioning contributes 18pp to storage savings
- Adaptive compression accounts for 25pp storage improvement
- Quality integration is critical without it, both optimization and detection degrade
- Multi-objective balancing improves all metrics vs single-objective optimization

5.5. *Integration Impact Analysis*

To demonstrate the value of integrating quality with optimization (our core contribution), we compare:

Scenario 1: Separated Systems

- Quality monitoring runs independently
- Optimization uses only performance metrics
- Human interprets quality alerts and manually triggers fixes

Scenario 2: Integrated AIDALOS

- Quality signals feed directly into RL state
- Optimization actions triggered automatically
- Closed-loop feedback

Integration provides substantial benefits: faster response to issues, fewer incorrect optimizations, and better overall outcomes. The 16x remediation speedup quantifies a shift from reactive to proactive data lake management. Traditional separated architectures create multi-day feedback loops: quality issues manifest in production, monitoring systems generate alerts, human operators investigate root causes, database administrators plan corrective actions, and changes execute during scheduled maintenance windows. Each handoff introduces delays hours for alert triage, additional hours for investigation, and days for scheduling maintenance. Integrated AIDALOS compresses this cycle to minutes: drift detection immediately signals the RL agent, which evaluates optimization actions using cached cost estimates, and physical changes execute automatically during the next available maintenance window (typically within hours). This acceleration is critical for data lakes supporting real-time ML models or customer-facing dashboards where multi-day quality issues directly impact business outcomes.

5.6. *Convergence and Training Analysis*

RL Agent Convergence:

- Episodes 0-800: Exploration phase, reward ~ -4.5
- Episodes 800-1800: Learning phase, reward improves to -1.2
- Episodes 1800+: Convergence, reward stabilizes at -0.8
- Training time: ~8 GPU-hours per pipeline

Model Retraining Schedule:

- Drift detectors: Retrain monthly (2 GPU-hours)
- Compression model: Retrain quarterly (3 GPU-hours)
- RL agent: Continuous online learning + full retrain every 6 months

5.7. *Overhead Analysis*

Monitoring Overhead:

- Drift detection: 15-30ms per batch
- Anomaly detection: 8-20ms per record batch
- Total monitoring: <2% of query execution time

Optimization Overhead:

- RL inference: 40-60ms per decision (amortized over days)

- Compression selection: 5ms per file
- Query cost estimation: 15ms per query (cached for similar patterns)

Physical Reconfiguration Cost:

- Repartitioning: One-time cost of 2-6 hours (done during maintenance)
- Recompression: Gradual background process
- Query hint updates: Real-time, negligible cost

Total system overhead: <3% of cluster resources.

5.8. Limitations Observed

- Cold Start: First 2-4 weeks on new pipeline, AIDALOS performs similarly to baselines while building models. Performance improves as training data accumulates.
- Workload Volatility: On P5 (media), which has highly variable workload, optimization decisions sometimes lag behind rapid changes. System performs best on moderately stable workloads.
- Hyperparameter Sensitivity: Reward weights (w_storage, w_latency, etc.) affect optimization priorities. We used same weights across all pipelines custom tuning per pipeline could improve results further.
- Explainability Gap: While we provide feature importance, explaining why RL agent chose specific partition key at specific time remains challenging for non-ML users.

6. Discussion

The experimental results validate a central hypothesis: quality monitoring and physical optimization exhibit mutual dependencies that traditional separated architectures fail to exploit. Quality signals provide early indicators of optimization opportunities, drift in partition key cardinality suggests repartitioning benefits before query performance degrades. Conversely, physical state informs quality assessment understanding current partitioning schemes enables more accurate skew detection than data-only analysis. The ablation study quantifies this interdependence: removing quality integration degrades optimization performance by 6-7 percentage points, while separated quality monitoring suffers 5.8 percentage points worse coverage due to lack of physical context. These bidirectional benefits demonstrate that integration is not merely architectural convenience but fundamental to achieving optimal outcomes.

6.1. Key Insights

Integration Value Demonstrated The ablation study (Section V.D) clearly shows that quality monitoring and physical optimization are synergistic. Without quality signals, optimization degrades by 6-7 percentage points. Without optimization actions, quality detection lacks context and generates more false positives. This validates our core thesis that these should be integrated, not separated.

- Learning-Based Adaptation Works RL agents successfully learn policies that match or exceed expert manual configurations, achieving 89% accuracy (when evaluated against expert-labeled optimal partitioning choices). The ability to adapt to workload changes demonstrated by continued performance over 60-day test period shows advantage over static heuristics.
- Multi-Objective Balance is Critical Single-objective optimization (e.g., minimize storage only) leads to suboptimal overall outcomes. NSGA-II solver balancing storage, latency, and quality produces configurations that improve all metrics simultaneously.
- Ensemble Robustness No single drift detector or anomaly model works well across all data types and pipeline characteristics. The ensemble approach (Section IV.B-C) provides robustness across diverse scenarios.

6.2. Comparison to Related Work

- vs Traditional Physical Design: Auto Admin, Schirmer systems operate offline and assume static workloads [11,13]. AIDALOS operates continuously and adapts to workload evolution, demonstrated by sustained performance over 60-day test period where workload patterns shifted.
- vs Learned Query Optimizers: Neo, Bao optimize query plans but don't address physical design [9, 36]. Our results show physical optimization (partitioning, compression) provides larger performance gains (62% latency reduction) than query-level optimization alone would achieve.

The complementary nature of physical and logical optimization suggests a layered approach to self-driving data lakes. Physical design provides foundational efficiency by organizing data to match access patterns, partitioning enables pruning, compression reduces storage and I/O, and file layout affects cache efficiency. These decisions operate at coarse granularity (table or partition level) and change infrequently (days to weeks). Logical query optimization refines execution for specific workloads by selecting join algorithms, adjusting parallelism, and choosing execution strategies. These decisions operate at fine granularity (per-query) and adapt rapidly (seconds to minutes). The 62% latency

improvement from physical optimization exceeds typical logical optimization gains (15-30%) because physical decisions eliminate work entirely through partition pruning and efficient data organization, while logical optimization distributes work more effectively but cannot reduce total data volume processed.

- vs Quality Tools: Deequ, Monte Carlo detects issues but don't act on them. AIDALOS closes the loop by triggering optimization actions based on detected quality signals, reducing remediation time from days to hours.
- vs Self-Driving Databases: Otter Tune, Self-Driving DB focus on parameter tuning in traditional DBMS [15, 17]. AIDALOS addresses data lake-specific challenges: schema flexibility, scale-out storage, diverse workloads, and quality-optimization integration.

6.3. *Generalization and Transfer Learning*

The cold-start limitation, requiring 2-4 weeks to match baseline performance on new pipelines, represents the most significant practical deployment barrier. Initial experiments with transfer learning show promise for accelerating convergence. When initializing the RL agent with policies learned from similar pipelines (e.g., training on finance data and deploying to another finance pipeline), convergence time reduces by approximately 50%. The drift detection ensemble exhibits stronger transfer characteristics: meta-learner weights trained on diverse historical data generalize well to new pipelines with similar data types, achieving 82% of steady-state accuracy immediately upon deployment. Future work should investigate pipeline similarity metrics that enable automated selection of transfer learning sources, potentially through clustering pipelines based on schema complexity, query pattern distributions, and data characteristic profiles.

6.4. *Limitations and Threats to Validity*

- Limited Pipeline Diversity: Five pipelines across different domains provide reasonable coverage but may not represent all data lake scenarios. Generalization to dramatically different workloads (e.g., graph analytics, geospatial) remains untested.
- Manual Expert Baseline: Our "manual configuration" baseline represents 3 months of expert tuning effort. More extensive tuning might achieve better results, though would be impractical for most organizations.
- Infrastructure Dependency: Evaluation performed on specific infrastructure (Spark, Delta Lake, S3). Performance characteristics may differ on other platforms (Presto, Hudi, HDFS), though core principles should transfer.
- Workload Stationarity: While we observe workload changes during 60-day test period, we didn't simulate extreme scenarios like complete workload replacement or adversarial queries.
- Cold Start Gap: 2-4 week initial learning period limits applicability for short-lived projects or frequently changing pipelines.

6.5. *Future Work*

- Transfer Learning: Pre-train models on diverse workloads to reduce cold-start time. Initial experiments suggest warm-starting from related pipelines can halve learning time.
- Multi-Cloud Optimization: Extend to optimize data placement and query routing across multiple clouds, considering pricing differences and data locality.
- Automated Remediation: Beyond detection and optimization, develop safe automated fixes for common quality issues (schema adaptation, outlier handling, missing value imputation).
- Causal Analysis: Current system learns correlations between quality signals and optimization outcomes. Adding causal reasoning could enable better generalization and counterfactual planning.
- Federated Learning for Cross-Organization Model Improvement: Multiple organizations operating similar data lake workloads could collaboratively improve optimization models without sharing sensitive data. For example, financial institutions could jointly train drift detection ensembles on anonymized schema patterns and distribution characteristics, improving detection accuracy for rare events that individual organizations encounter infrequently. The federated learning paradigm enables this collaboration: each organization trains local models on proprietary data, shares only model updates (gradients or parameters) with a central aggregator, and receives improved global models trained on collective experience. This approach is particularly valuable for healthcare and finance sectors where data privacy regulations prohibit direct sharing but regulatory requirements (HIPAA, SOX) create common quality and performance challenges. Technical challenges include managing statistical heterogeneity across organizations with different data distributions, ensuring privacy guarantees through differential privacy or secure aggregation, and developing incentive mechanisms that encourage participation.
- Enhanced Interpretability: Develop better explanation mechanisms for RL decisions, including

counterfactual explanations ("system chose X because Y; if Z had been different, it would have chosen W").

6.6. Conclusion

This paper presented AIDALOS, an integrated framework for data lake optimization that bridges the traditional separation between quality monitoring and physical design. By feeding quality signals directly into reinforcement learning agents that control partitioning, compression, and query optimization, we demonstrate a closed-loop system that adapts continuously to workload and data changes. Our evaluation across five production ML pipelines validates both the monitoring subsystem (89.9% F1-score for quality detection) and the optimization subsystem (47% storage cost reduction, 62% query latency improvement). Crucially, ablation studies show that integration provides synergistic benefits quality detection improves optimization decisions, and optimization actions provide context that reduces false positives. The key contribution is architectural: demonstrating that self-managing data platforms should integrate observability with actionability. Quality monitoring generates signals; optimization agents act on those signals; outcomes feedback to refine models creating systems that learn and improve autonomously. While challenges remain particularly cold-start performance, interpretability, and generalization to diverse workloads the results establish that integrated, learning-based optimization is both technically feasible and practically beneficial for production data lakes. As data volumes grow and workloads become increasingly dynamic, such adaptive systems will transition from optional enhancement to operational necessity.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The author gratefully acknowledges the academic and technical support provided by research collaborators during the design and implementation of this study. No external funding was received for this work

References

[1] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," *Advances in Neural Information Processing Systems*, vol. 28, pp. 2503–2511, 2015.

[2] J. Dixon, "Data lakes: a new generation of data repositories," *Proceedings of the ACM SIGMOD Workshop on Data Analytics in the Cloud*, 2010.

[3] Sharma, V. Kumar, and R. Gupta, "Modern data lakes: a conceptual framework," *IEEE Access*, vol. 9, pp. 127876–127891, 2021, doi:10.1109/ACCESS.2021.3112517.

[4] M. Armbrust, T. Das, S. Zhu, R. Xin, B. Ghodsi, J. Stoica, and M. Zaharia, "Delta lake: high-performance ACID table storage," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 3411–3424, 2020, doi:10.14778/3415478.3415560.

[5] M. Armbrust, J. Shi, A. Jindal, G. K. Lee, K. Xin, M. Zaharia, and I. Stoica, "Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics," *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2021.

[6] V. Prashanth, S. Das, J. Li, and V. Narasayya, "Apache hudi: the case for incremental processing on big data," *IEEE Data Engineering Bulletin*, vol. 44, no. 1, pp. 13–27, 2021.

[7] R. Blue, D. Petersohn, A. Reeves, and M. Rodgers, "Apache iceberg: a modern table format for big data," *Proceedings of the VLDB Endowment*, vol. 13, no. 12, pp. 3411–3424, 2020.

[8] T. Kraska, A. Beutel, E. H. Chi, J. Dean, and N. Polyzotis, "The case for learned index structures," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 489–504, 2018, doi:10.1145/3183713.3196909.

[9] R. Marcus, P. Negi, H. Mao, C. Zhang, N. Tatbul, M. Alizadeh, T. Kraska, O. Papaemmanouil, and N. Polyzotis, "Neo: a learned query optimizer," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, pp. 1705–1718, 2019.

[10] Kipf, T. Kipf, B. Radke, and V. Markl, "Learned cardinalities: estimating correlated joins with deep learning," *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2019.

[11] S. Chaudhuri and V. Narasayya, "An efficient cost-driven index selection tool for Microsoft SQL Server," *Proceedings of the VLDB Conference*, pp. 146–155, 1997.

[12] N. Bruno and S. Chaudhuri, "Automatic physical database tuning: a relaxation-based approach," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 227–238, 2005, doi:10.1145/1066157.1066187.

[13] Schirmer, T. Neumann, and A. Kemper, "Workload-driven horizontal partitioning and pruning for large OLTP systems," *Proceedings of the IEEE ICDE Workshops*, pp. 146–151, 2018.

[14] Z. Abedjan, L. Golab, and F. Naumann, "Data profiling," *Synthesis Lectures on Data Management*, vol. 10, no. 4, pp. 1–154, 2018, doi:10.2200/S00838ED1V01Y201808DTM045.

[15] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys*, vol. 46, no. 4, pp. 1–37, 2014, doi:10.1145/2523813.

[16] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: a survey," *arXiv preprint arXiv:1901.03407*, 2019.

[17] Pavlo, G. Angulo, J. Arulraj, H. Lin, J. Lin, L. Ma, P. Menon, T. Mühlbauer, S. Tozer, and D. Stonebraker, "Self-driving database management systems," *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2017.

[18] T. Kraska, M. Alizadeh, A. Beutel, E. H. Chi, A. Kristo, G. Leclerc, S. Madden, H. Mao, and V. Nathan, "SageDB: a learned database system," *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2019.

[19] D. Van Aken, A. Pavlo, G. Gordon, and B. Zhang, "Automatic database management system tuning through large-scale machine learning," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1009–1024, 2017, doi:10.1145/3035918.3064029.

[20] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002, doi:10.1109/4235.996017.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015, doi:10.1038/nature14236.

[22] R. Sutton and A. Barto, *Reinforcement learning: an introduction*, 2nd ed., Cambridge, MA, USA: MIT Press, 2018.

[23] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," *Proceedings of the International Conference on Machine Learning*, pp. 1995–2003, 2016.

[24] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," *Proceedings of the International Conference on Learning Representations*, 2015.

[25] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," *Proceedings of the International Conference on Learning Representations*, 2016.

[26] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, no. 1–2, pp. 100–115, 1954, doi:10.1093/biomet/41.1-2.100.

[27] Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," *Proceedings of the SIAM International Conference on Data Mining*, 2007.

[28] Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. Smola, "A kernel two-sample test," *Journal of Machine Learning Research*, vol. 13, pp. 723–773, 2012.

[29] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, 2016, doi:10.1145/2939672.2939785.

[30] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *Proceedings of the International Conference on Learning Representations*, 2014.

[31] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi:10.1162/neco.1997.9.8.1735.

[32] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *Proceedings of the International Conference on Learning Representations*, 2015.

[33] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[34] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *Proceedings of the International Conference on Learning Representations*, 2017.

[35] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Apache Spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, Nov. 2016, doi:10.1145/2934664.

[36] R. Marcus, P. Negi, H. Mao, C. Zhang, N. Tatbul, M. Alizadeh, T. Kraska, O. Papaemmanouil, and N. Polyzotis, "Bao: making learned query optimization practical," *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1275–1288, 2021.

SOWJANYA DEVA obtained her master's degree in data science from University of Maryland, Baltimore County after completing her undergraduate studies in computer science and engineering. She is currently working in the data engineering and analytics industry.

SURYA NARAYANA REDDY CHINTACUNTA obtained his Master of Professional Studies (MPS) in Data Science from the University of Maryland, Baltimore County. He is currently working in the data engineering and cloud computing industry, with a focus on scalable data systems, cloud-native architectures, and data-driven analytics.

JENRS

Received: 18 January 2026, Revised: 22 February 2026, Accepted: 28 February 2026, Online: 07 March 2026

DOI: https://doi.org/10.55708/js0503002

A Note on Modified Stokes' Problems for Fluids with Power-Law Dependence of Viscosity on Pressure with 3/2 index

Constantin Fetecau*
Academy of Romanian Scientists, 3 Ilfov, Bucharest 050044, Romania
*Corresponding author: Constantin Fetecau, Email: c_fetecau@yahoo.com

ABSTRACT: The modified Stokes' problems for incompressible Newtonian fluids with power-law dependence of viscosity on the pressure of 3/2 index are analytically investigated. The influence of the gravitational acceleration is considered. Exact expressions are derived from permanent dimensionless velocity and shear stress fields in terms of standard Bessel functions. They satisfy the governing equations and boundary conditions. Similar solutions corresponding to same problems for ordinary fluids are recovered as limiting cases of previous solutions. Some characteristics of fluid behavior are graphically underlined. It is shown that the fluids with pressure-dependent viscosity flow faster than ordinary fluids and the shear stress of the first problem of Stokes is constant on entire flow domain although the corresponding velocity is function of the spatial variable. Obtained solutions, which are new in the literature, have been used to find the required time to touch the steady state. This time is important for experimental researchers who want to know the transition moment of the motion to steady state.

KEYWORDS: Modified Stokes' problems, Pressure-dependent viscosity, Permanent solutions

1. Introduction

The fluid viscosity is usually considered to be constant in isothermal flows of incompressible Newtonian fluids. However, in some cases, it substantially increases at high pressures [1,2]. Stokes [3] was the first who remarked that the liquids' viscosity could depend on pressure. Later, the experimental research of [4]-[10] confirmed his supposition. The most of this experimental research showed that the fluid viscosity can change by as much as 10^8 at high pressures [11] and the independence of viscosity on pressure can be accepted at low pressures. At high pressures, like in polymer and food processing, pharmaceutical tablet manufacturing, crude oil and fuel oil pumping, microfluidics and geophysics, can appear significant errors [12,13]. The blade coating process of incompressible stress fluids with pressure-dependent viscosity was recently investigated by [14] in plane and exponential geometries. The plane Poiseuille flow of the viscoelastic fluids with pressure-dependent viscosity in a narrow nanochannel was asymptotically investigated by [15]. An important domain where the effects of pressure on viscosity cannot be ignored is that of electrohydrodynamic lubrication [16]. On the other hand, the effect of pressure on the fluid density is much smaller in comparison with that on viscosity and the fluid compressibility can be ignored.

To our knowledge, in the existing literature there is no general relationship between viscosity and pressure that is valid for all types of fluids. The first state relations for the variation of the viscosity η with the pressure p seem to be linear or exponential [17, 18], i.e.

$$\eta = \eta(p) = \mu[1 + \alpha(p - p_{ref})] \text{ or } \eta = \eta(p) = \mu e^{\alpha(p-p_{ref})}. \quad (1)$$

In above relations α is the dimensional pressure-viscosity coefficient and μ is the fluid viscosity at the reference pressure p_{ref}. The two relations seem to be more adequate to deserve the experimental data at lower and high pressures, respectively. Exact expressions for the permanent (steady or long time) velocity and the vorticity corresponding to the Couette flow of fluids with linear, exponential and power-law dependence of viscosity on pressure have been determined by [19]. Permanent velocity fields for motions of fluids with linear dependence of viscosity on pressure were provided by [20] and [21] in rectangular ducts. The Hele-Shaw flow of

fluids with pressure-dependent viscosity was analytical investigated by [22]. Numerical solutions for the flow of an incompressible viscous fluid with power-law dependence of viscosity on pressure between intersecting planes have been obtained by [23]. Permanent solutions of the modified Stokes' problems for some fluids with power-law dependence of viscosity on pressure have been recently obtained by [24] and [25]. An interesting book about steady-state (permanent) motions of fluids with variable power-law was recently published by [26].

The purpose of this note is to provide dimensionless permanent solutions of the modified Stokes' problems for a class of fluids with power-law dependence of viscosity on pressure when gravity effects are taken into consideration. Analytic expressions for the permanent velocity and shear stress fields are determined using suitable changes of the spatial variable and the unknown function. These expressions can be used to determine the required time to get the steady or permanent state of the respective motions. This time is very important for the experimental researchers who want to know the transition moment of the motion to the steady state. Finally, graphical representations showed that fluids with pressure-dependent viscosity flow faster in comparison with ordinary fluids.

2. Problem Presentation

Let us consider an incompressible Newtonian fluid with power-law dependence of viscosity on the pressure of index 3/2 in stationary state between two infinite horizontal parallel plates. The corresponding Cauchy stress tensor $\boldsymbol{T}$ is given by the relation

$$\begin{aligned}\boldsymbol{T} &= -p\boldsymbol{I} + \eta(p)(\boldsymbol{L}+\boldsymbol{L}^T)\\ &= -p\boldsymbol{I} + \mu[\alpha(p-p_{ref})+1]^{3/2}(\boldsymbol{L}+\boldsymbol{L}^T).\end{aligned} \tag{2}$$

Here, $-p\boldsymbol{I}$ is the reaction stress due to the constraint of incompressibility, $\boldsymbol{L}$ is the gradient of the velocity vector $\boldsymbol{u}$, $\alpha > 0$ is the dimensional pressure-viscosity coefficient and μ is the fluid viscosity at the reference pressure p_{ref}. The fluids defined by the constitutive equation (2), also called piezo-viscous liquids, focuses on advancing mathematical modeling and industrial applications like high-pressure lubrication, coating and polymer processing. The ordinary incompressible Newtonian fluids correspond to the case $\alpha = 0$.

After the moment $t = 0$, the inferior plate begins to slide along its plane with the constant velocity V or to oscillate in the same plane with the time-dependent velocity $V\cos(\omega t)$ or $V\sin(\omega t)$. The constant ω is the oscillations' frequency. Owing to the shear the fluid begins to move. We are looking for a velocity vector $\boldsymbol{u}$ and pressure p of the form [27]

$$\boldsymbol{u} = \boldsymbol{u}(z,t) = u(z,t)\mathbf{e}_y, \;\; p = p(z), \tag{3}$$

in a fixed system of Cartesian coordinate x, y and z in which the z-axis is normal to plates and e_y is the unit vector along the y-axis. Replacing the velocity vector $\boldsymbol{u}$ from Eq. (3) in (2), it results that the non-null shear stress $\tau(z,t)$ is given by the relation

$$\tau(z,t) = \mu[\alpha(p-p_{ref})+1]^{3/2}\frac{\partial u(z,t)}{\partial z}; 0 < z < d,\ t > 0, \tag{4}$$

where d is the distance between plates.

The balance of linear momentum for such motions of incompressible Newtonian fluids with or without pressure-dependent viscosity reduces to the linear differential equations

$$\rho\frac{\partial u(z,t)}{\partial t} = \frac{\partial \tau(z,t)}{\partial z}, \frac{dp(z)}{dz} = -\rho g;\ 0 < z < d,\ t > 0. \tag{5}$$

In above relations ρ is the fluid density and g is the gravitational acceleration. The second relation implies

$$p = \rho g(d-z) + p_{ref} \;\text{ where }\; p_{ref} = p(d). \tag{6}$$

The two unknown functions $u(z,t)$ and $\tau(z,t)$ have to satisfy the initial conditions

$$u(z,0) = 0,\ \tau(z,0) = 0 \text{ if } 0 \le z \le d,\ t > 0, \tag{7}$$

and the boundary conditions

$$u(0,t) = V,\ u(d,t) = 0 \text{ if } t > 0, \tag{8}$$

for the modified first problem of Stokes and

$$\begin{aligned}&u(0,t) = V\cos(\omega t) \text{ or } u(0,t) = V\sin(\omega t),\\ &u(d,t) = 0 \text{ if } t > 0,\end{aligned} \tag{9}$$

for the modified Stokes' second problem.

The dimensionless forms of the governing equations (4) and (5), namely

$$\tau(z,t) = [\alpha(1-z)+1]^{3/2}\frac{\partial u(z,t)}{\partial z};\ 0 < z < 1,\ t > 0, \tag{10}$$

$$\frac{\partial u(z,t)}{\partial t} = \frac{\partial \tau(z,t)}{\partial z};\ 0 < z < 1,\ t > 0, \tag{11}$$

have been obtained using the non-dimensional variables, functions and parameters

$$\begin{aligned}&\overline{z} = \frac{1}{d}z,\ \overline{t} = \frac{\nu}{d^2}t,\ \overline{u} = \frac{1}{V}u,\\ &\overline{\tau} = \frac{d}{\mu V}\tau,\ \overline{\alpha} = \alpha\rho g d,\ \overline{\omega} = \frac{d^2}{\nu}\omega,\end{aligned} \tag{12}$$

and renouncing the bar notation. The non-dimensional forms of initial conditions remain unchanged while the corresponding boundary conditions (8) and (9) become

$$u(0,t)=1,\ u(1,t)=0 \text{ if } t>0, \tag{13}$$

respectively,

$$u(0,t)=\cos(\omega t) \text{ or } u(0,t)=\sin(\omega t),\ u(1,t)=0 \text{ if } t>0. \tag{14}$$

It is well known from the literature that the fluid motions corresponding to the modified Stokes' problems become steady in time. It means that, sometime after the motion initiation, the fluid behavior is described by the starting velocities and shear stress fields. After this time, the fluid motion can be characterized by the permanent velocity and shear stress fields which are independent of the initial conditions but satisfy the governing equations and boundary conditions. In practice, this time is very important for experimental researchers who want to know the transition moment of the motion to the steady or permanent state. In order to determine this time, it is sufficient to know the permanent solutions. This is the reason that exact expressions will be determined only for these solutions in next section.

3. Solutions

3.1. Modified Stokes' first problem

To avoid confusion, we denote by u_{Cp} and τ_{Cp} the dimensionless permanent velocity and shear stress fields of the corresponding motion. Direct computations show that

$$\begin{aligned} u_{Cp}(z)&=\frac{\sqrt{\alpha+1}}{1-\sqrt{\alpha+1}}\frac{1-\sqrt{\alpha(1-z)+1}}{\sqrt{\alpha(1-z)+1}},\\ \tau_{Cp}&=\frac{\alpha\sqrt{\alpha+1}}{2(1-\sqrt{\alpha+1})};\ 0<z<1. \end{aligned} \tag{15}$$

Consequently, the shear stress τ_{Cp} is constant on the entire flow domain although the fluid velocity is a function of the spatial variable z. By taking the limits of the relations (15) when $\alpha\to 0$ one recovers the permanent solutions corresponding to ordinary fluids

$$u_{oCp}(z)=\lim_{\alpha\to 0}u_{Cp}(z)=1-z,\ \tau_{oCp}=\lim_{\alpha\to 0}\tau_{Cp}=-1, \tag{16}$$

obtained by Fetecau [25] in a different way.

3.2. Modified Stokes' second problem

For distinction let us denote by $u_{cp}(z,t)$, $\tau_{cp}(z,t)$ and $u_{sp}(z,t)$, $\tau_{sp}(z,t)$ the dimensionless permanent velocity and shear stress fields corresponding to the two motions induced by cosine or sine oscillations of the lower plate. To determine them in a simple way, let us introduce the complex velocity and shear stress fields

$$\begin{aligned} u_{com}(z,t)&=u_{cp}(z,t)+iu_{sp}(z,t),\\ \tau_{com}(z,t)&=\tau_{cp}(z,t)+i\tau_{sp}(z,t), \end{aligned} \tag{17}$$

where i is imaginary unit. The two complex entities have to satisfy the system of partial differential equations

$$\tau_{com}(z,t)=[\alpha(1-z)+1]^{3/2}\frac{\partial u_{com}(z,t)}{\partial z};\quad 0<z<1,\ t\in R, \tag{18}$$

$$\frac{\partial u_{com}(z,t)}{\partial t}=\frac{\partial \tau_{com}(z,t)}{\partial z};\ 0<z<1,\ t\in R, \tag{19}$$

with the boundary conditions

$$u_{com}(0,t)=e^{i\omega t},\ u_{com}(1,t)=0;\ t\in R. \tag{20}$$

Substituting $\tau_{com}(z,t)$ from Eq. (18) in (19) one attains to the governing equation

$$\begin{aligned} &[\alpha(1-z)+1]^{3/2}\frac{\partial^2 u_{com}(z,t)}{\partial z^2}\\ &-\frac{3}{2}\alpha[\alpha(1-z)+1]^{1/2}\frac{\partial u_{com}(z,t)}{\partial z}\\ &=\frac{\partial u_{com}(z,t)}{\partial t};\ 0<z<1,\ t\in R, \end{aligned} \tag{21}$$

for the dimensionless complex velocity $u_{com}(z,t)$. Making the change of independent variable

$$z=(\alpha+1-r^2)/\alpha \text{ where } r\in(1,a) \text{ with } a=\sqrt{\alpha+1}, \tag{22}$$

one finds the following partial differential equation

$$\begin{aligned} &r\frac{\partial^2 u_{com}(r,t)}{\partial r^2}+2\frac{\partial u_{com}(r,t)}{\partial r}\\ &=\frac{4}{\alpha^2}\frac{\partial u_{com}(r,t)}{\partial t};\ 1<r<a,\ t\in R, \end{aligned} \tag{23}$$

with the boundary conditions

$$u_{com}(1,t)=0,\ u_{com}(a,t)=e^{i\omega t};\ t\in R. \tag{24}$$

Now, making the change of unknown function

$$u_{com}(r,t)=\frac{1}{r}w_{com}(r,t);\ 1<r<a,\ t\in R, \tag{25}$$

one attains to the next boundary value problem

$$\begin{aligned} &r\frac{\partial^2 w_{com}(r,t)}{\partial r^2}-\frac{4}{\alpha^2}\frac{\partial w_{com}(r,t)}{\partial t}=0;\\ &w_{com}(1,t)=0,\ w_{com}(a,t)=ae^{i\omega t}. \end{aligned} \tag{26}$$

The boundary conditions and the linearity of the governing equation (26) suggest us look for a solution of the form

$$w_{com}(r,t)=W(r)e^{i\omega t};\ 1<r<a,\ t\in R. \tag{27}$$

Replacing $w_{com}(z,t)$ from Eq. (27) in (26) one finds the next boundary value problem

$$r\frac{d^2W(r)}{dr^2}-\frac{4i\omega}{\alpha^2}W(r)=0;\ W(1)=0,\ W(a)=a, \tag{28}$$

for the complex function $W(\cdot)$. Now, based on the problem 37 from the page 251 of the reference [28], one can say that the general solution of the boundary value problem (28) is given by the relation

$$W(r)=\sqrt{r}[c_1J_1(\beta\sqrt{r})+c_2Y_1(\beta\sqrt{r})], \tag{29}$$

where c_1 and c_2 are constants and $\beta=4i\sqrt{i\omega}/\alpha$. Using the boundary conditions (28) one finds that

$$W(r)=\sqrt{ar}\,\frac{Y_1(\beta)J_1(\beta\sqrt{r})-J_1(\beta)Y_1(\beta\sqrt{r})}{Y_1(\beta)J_1(\beta\sqrt{a})-J_1(\beta)Y_1(\beta\sqrt{a})}, \tag{30}$$

and the complex velocity $w_{com}(z,t)$ is given by the relation

$$w_{com}(r,t)=\sqrt{ar}\,\frac{Y_1(\beta)J_1(\beta\sqrt{r})-J_1(\beta)Y_1(\beta\sqrt{r})}{Y_1(\beta)J_1(\beta\sqrt{a})-J_1(\beta)Y_1(\beta\sqrt{a})}e^{i\omega t};\quad 1<r<a,\ t\in R. \tag{31}$$

Finally, coming back to the original function and variables and bearing in mind the notation (17)$_1$, it results that the dimensionless permanent velocities $u_{cp}(z,t)$ and $u_{sp}(z,t)$ are given by the relations

$$u_{cp}(z,t)=\frac{\sqrt[4]{\alpha+1}}{\sqrt[4]{\alpha(1-z)+1}}\times\mathrm{Re}\left\{\frac{Y_1(\beta)J_1(\beta\sqrt[4]{\alpha(1-z)+1})-J_1(\beta)Y_1(\beta\sqrt[4]{\alpha(1-z)+1})}{Y_1(\beta)J_1(\beta\sqrt[4]{\alpha+1})-J_1(\beta)Y_1(\beta\sqrt[4]{\alpha+1})}e^{i\omega t}\right\}, \tag{32}$$

$$u_{sp}(z,t)=\frac{\sqrt[4]{\alpha+1}}{\sqrt[4]{\alpha(1-z)+1}}\times\mathrm{Im}\left\{\frac{Y_1(\beta)J_1(\beta\sqrt[4]{\alpha(1-z)+1})-J_1(\beta)Y_1(\beta\sqrt[4]{\alpha(1-z)+1})}{Y_1(\beta)J_1(\beta\sqrt[4]{\alpha+1})-J_1(\beta)Y_1(\beta\sqrt[4]{\alpha+1})}e^{i\omega t}\right\}. \tag{33}$$

The expressions of the corresponding shear stresses $\tau_{cp}(z,t)$ and $\tau_{sp}(z,t)$, namely

$$\tau_{cp}(z,t)=\sqrt[4]{\alpha+1}\sqrt{\alpha(1-z)+1}\times\mathrm{Re}\left\{\frac{Y_1(\beta)J_2(\beta\sqrt[4]{\alpha(1-z)+1})-J_1(\beta)Y_2(\beta\sqrt[4]{\alpha(1-z)+1})}{Y_1(\beta)J_1(\beta\sqrt[4]{\alpha+1})-J_1(\beta)Y_1(\beta\sqrt[4]{\alpha+1})}i\sqrt{i\omega}\,e^{i\omega t}\right\}, \tag{34}$$

$$\tau_{sp}(z,t)=\sqrt[4]{\alpha+1}\sqrt{\alpha(1-z)+1}\times\mathrm{Im}\left\{\frac{Y_1(\beta)J_2(\beta\sqrt[4]{\alpha(1-z)+1})-J_1(\beta)Y_2(\beta\sqrt[4]{\alpha(1-z)+1})}{Y_1(\beta)J_1(\beta\sqrt[4]{\alpha+1})-J_1(\beta)Y_1(\beta\sqrt[4]{\alpha+1})}i\sqrt{i\omega}\,e^{i\omega t}\right\}, \tag{35}$$

have been determined using the relations (17), (18), (31) and the identity

$$xJ_q'(x)-qJ_q(x)=-xJ_{q+1}(x). \tag{36}$$

4. Results' validation

By using asymptotic approximations of the standard Bessel functions $J_q(\cdot)$ and $Y_q(\cdot)$, namely

$$J_q(x)\approx\frac{x^q}{2^q\Gamma(q+1)},\qquad Y_q(x)\approx-\frac{2^q\Gamma(q)}{\pi x^q}\ \text{for}\ q>0\ \text{and}\ x<<1, \tag{37}$$

it is not difficult to show that

$$u_{Cp}(z)=\lim_{\omega\to0}u_{cp}(z,t),\ \tau_{Cp}=\lim_{\omega\to0}\eta_{cp}(z,t). \tag{38}$$

Consequently, as expected, the permanent dimensionless velocity and shear stress fields $u_{Cp}(z)$ and τ_{Cp} corresponding to the first problem of Stokes for the considered fluids are obtained as limiting cases of the permanent velocity and shear stress fields $u_{cp}(z,t)$ and $\tau_{cp}(z,t)$ of the second problem of Stokes when the oscillations' frequency $\omega\to0$.

Let's now use the next asymptotic approximations

$$J_q(x)\approx\sqrt{\frac{2}{\pi x}}\cos\left[x-\frac{(2q+1)\pi}{4}\right],\qquad Y_q(x)\approx\sqrt{\frac{2}{\pi x}}\sin\left[x-\frac{(2q+1)\pi}{4}\right]\ \text{for}\ x>>1, \tag{39}$$

of same Bessel functions. Using them in relations (32)-(35) for small values of the pressure-viscosity coefficient α and large values of the parameter β, one can show that

$$u_{cp}(z,t)\approx\frac{\sqrt[8]{(\alpha+1)^3}}{\sqrt[8]{[\alpha(1-z)+1]^3}}\mathrm{Re}\left\{\frac{\sin\{\beta[1-\sqrt[4]{\alpha(1-z)+1}]\}}{\sin[\beta(1-\sqrt[4]{\alpha+1})]}e^{i\omega t}\right\}, \tag{40}$$

$$u_{sp}(z,t)\approx\frac{\sqrt[8]{(\alpha+1)^3}}{\sqrt[8]{[\alpha(1-z)+1]^3}}\mathrm{Im}\left\{\frac{\sin\{\beta[1-\sqrt[4]{\alpha(1-z)+1}]\}}{\sin[\beta(1-\sqrt[4]{\alpha+1})]}e^{i\omega t}\right\}, \tag{41}$$

$$\tau_{cp}(z,t)\approx\sqrt[8]{(\alpha+1)^3[\alpha(1-z)+1]^3}\times\mathrm{Re}\left\{\frac{\cos\{\beta[1-\sqrt[4]{\alpha(1-z)+1}]\}}{\sin[\beta(1-\sqrt[4]{\alpha+1})]}i\sqrt{i\omega}\,e^{i\omega t}\right\}, \tag{42}$$

$$\tau_{sp}(z,t) \approx \sqrt[8]{(\alpha+1)^3[\alpha(1-z)+1]^3} \times \mathrm{Im}\left\{\frac{\cos\{\beta[1-\sqrt[4]{\alpha(1-z)+1}]\}}{\sin[\beta(1-\sqrt[4]{\alpha+1})]} i\sqrt{i\omega}\, e^{i\omega t}\right\}. \quad (43)$$

Introducing the following approximations

$$\sqrt[4]{\alpha(1-z)+1} \approx 1+\frac{1}{4}\alpha(1-z)+..., \quad \sqrt[4]{\alpha+1} \approx 1+\frac{1}{4}\alpha+..., \quad (44)$$

in the previous relations and taking their limits when the pressure-viscosity coefficient $\alpha \to 0$ it results that the dimensionless permanent velocity and shear stress fields $u_{ocp}(z,t), u_{osp}(z,t), \tau_{ocp}(z,t)$ and $\tau_{osp}(z,t)$ corresponding to the modified Stokes' second problem for ordinary incompressible Newtonian fluids are given by the relations

$$u_{ocp}(z,t) \approx \lim_{\alpha\to 0} u_{cp}(z,t) = \mathrm{Re}\left\{\frac{\sin[i\sqrt{i\omega}\,(1-z)]}{\sin(i\sqrt{i\omega})} e^{i\omega t}\right\}, \quad (45)$$

$$u_{osp}(z,t) \approx \lim_{\alpha\to 0} u_{sp}(z,t) = \mathrm{Im}\left\{\frac{\sin[i\sqrt{i\omega}\,(1-z)]}{\sin(i\sqrt{i\omega})} e^{i\omega t}\right\}, \quad (46)$$

$$\tau_{ocp}(z,t) \approx \lim_{\alpha\to 0} \tau_{cp}(z,t) = -\mathrm{Re}\left\{\frac{\cos[i\sqrt{i\omega}\,(1-z)]}{\sin(i\sqrt{i\omega})} i\sqrt{i\omega}\, e^{i\omega t}\right\}, \quad (47)$$

$$\tau_{osp}(z,t) \approx \lim_{\alpha\to 0} \tau_{sp}(z,t) = -\mathrm{Im}\left\{\frac{\cos[i\sqrt{i\omega}\,(1-z)]}{\sin(i\sqrt{i\omega})} i\sqrt{i\omega}\, e^{i\omega t}\right\}. \quad (48)$$

Finally, using the identities

$$\cos(ix) = \cosh(x), \ \sin(ix) = i\sinh(x), \quad (49)$$

in Eqs. (45)-(48) one recovers the simpler expressions of these solutions (see the relations (54) and (55) with $M = 0$ of Fetecau and Hanifa [24]), namely,

$$u_{ocp}(z,t) = \mathrm{Re}\left\{\frac{\sinh[(1-z)\sqrt{i\omega}]}{\sinh(\sqrt{i\omega})} e^{i\omega t}\right\}, \quad u_{osp}(z,t) = \mathrm{Im}\left\{\frac{\sinh[(1-z)\sqrt{i\omega}]}{\sinh(\sqrt{i\omega})} e^{i\omega t}\right\}, \quad (50)$$

$$\tau_{ocp}(z,t) = -\mathrm{Re}\left\{\frac{\cosh[(1-z)\sqrt{i\omega}]}{\sinh(\sqrt{i\omega})} \sqrt{i\omega}\, e^{i\omega t}\right\}, \quad \tau_{osp}(z,t) = -\mathrm{Im}\left\{\frac{\cosh[(1-z)\sqrt{i\omega}]}{\sinh(\sqrt{i\omega})} \sqrt{i\omega}\, e^{i\omega t}\right\}. \quad (51)$$

As expected, taking the limits of the relations $(50)_1$ and $(51)_1$ when the oscillations' frequency $\omega \to 0$, one recovers the dimensionless permanent velocity and shear stress fields $u_{oCp}(z)$ and τ_{oCp} given by the relations (16). Furthermore, simple computations show that the dimensional forms of $u_{ocp}(z,t)$ and $u_{osp}(z,t)$ given by the relations (50) are identical to the corresponding results obtained by Rajagopal [29].

5. Some graphical representations and numerical results

In this note are established closed-form expressions for the dimensionless permanent velocity and shear stress fields of modified Stokes problems for a class of incompressible Newtonian fluids with pressure-dependent viscosity. They are firstly important for the experimental researchers who want to know the transition moment of the motion to the steady or permanent state. In addition to these solutions, which characterize the fluid behavior after this moment, can be also used as tests to verify different numerical methods that are used to study more complex motion problems. Here, we shall use them to bring to light the influence of the pressure-viscosity coefficient on the respective motions.

For comparison, Figures 1 and 2 present the time variations of the dimensionless permanent velocities $u_{cp}(z,t)$, $u_{sp}(z,t)$ and shear stresses $\tau_{cp}(z,t)$ $\tau_{sp}(z,t)$, respectively, at the middle of channel for $\omega = \pi/6$ and three values of the pressure-viscosity coefficient α.

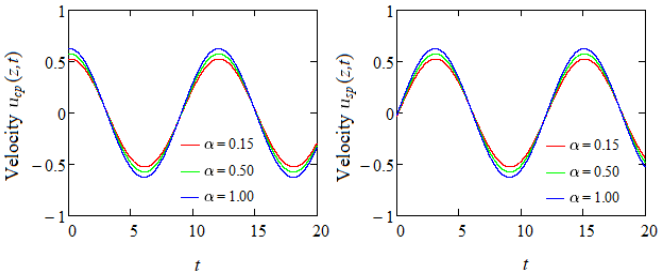

Figure 1: Time variations of the dimensionless permanent velocities $u_{cp}(z,t)$ and $u_{sp}(z,t)$ from Eqs. (32) and (33) at middle of channel ($z = 0.5$) for $\omega = \pi/6$ and three values of α.

The oscillatory behavior of the two motions and the phase difference between them are clearly visualized. In addition, as expected, the oscillations' amplitudes of the two motions are identical for same values of physical parameters. Furthermore, as it results from Figures 1, the oscillations' amplitude is an increasing function with respect to the parameter α. It means that fluids with pressure-dependent viscosity flow faster in comparison with ordinary fluids.

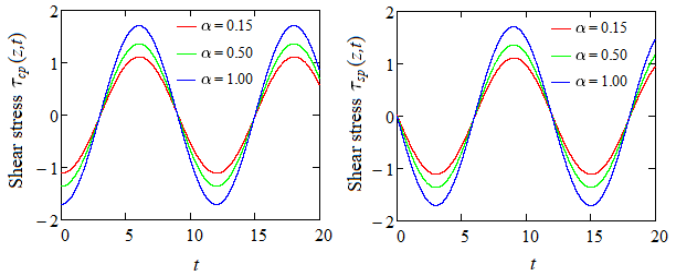

Figure 2: Time variations of the dimensionless permanent shear stresses $\tau_{cp}(z,t)$ and $\tau_{sp}(z,t)$ given by Eqs. (34) and (35) at the middle of channel ($z = 0.5$) for $\omega = \pi / 6$ and three values of α .

The variations of dimensionless permanent velocity u_{Cp} with respect to the spatial variable z and the pressure-viscosity coefficient α are presented in Figures 3. From these figures it clearly results that the fluid velocity is a decreasing function with respect to z and grows up for increasing values of α. Consequently, as before, the fluids with pressure-dependent viscosity flow faster than ordinary fluids and their speed increases as we approach the moving plate. This is possible for pressure-dependent viscosity fluid motions induced by a moving plate because the fluid adheres to the walls and the influence of the plate on the fluid movement brings up increasing fluid viscosity. Boundary conditions are clearly satisfied.

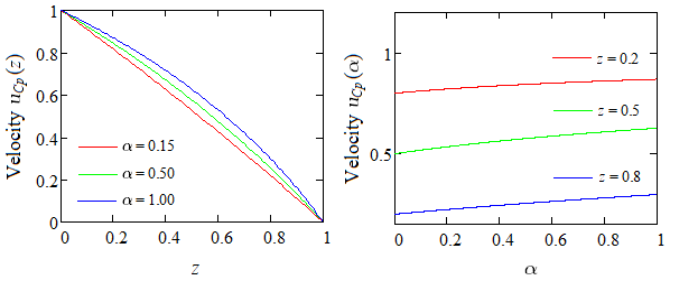

Figure 3: Variations of the dimensionless permanent velocity $u_{Cp}(z)$ given by Eq. (15) versus z at three values of α , and as function of α for three values of z.

The oscillatory behavior of the fluid motion in the case of the modified second Stokes' problem confers the essential difference from the modified first Stokes' problem of the same fluids.

6. Conclusions

As it was previously mentioned, modified Stokes' problems for a class of incompressible Newtonian fluids with pressure-dependent viscosity are analytically and numerically investigated. The main results that have been brought to light in this short note are:

- Analytic expressions of the dimensionless permanent velocity and shear stress fields corresponding to the modified Stokes' problems for some fluids with power-law dependence of viscosity on pressure have been provided.
- Similar solutions corresponding to same problems for ordinary incompressible Newtonian fluids have been recovered as limiting cases of general solutions using asymptotic approximations of Bessel functions.
- Graphical representations brought to light some characteristics of the fluid behavior and showed that fluids with pressure-depending viscosity flow faster than ordinary fluids.
- As expected, the fluid velocity increases as we approach the moving plate and the boundary conditions are clearly satisfied.

Finally, we remember the fact that the steady solutions for unsteady motions of fluids are important in practice to determine the required time to reach the steady state. From mathematical point of view, this is the time after which the diagrams of starting solutions (numerical solutions) superpose over those of steady state solutions. This time is very important for the experimental researchers who want to know the transition moment of the motion to the steady state. In addition, the exact solutions can be also useful to test different numerical schemes that are used to describe more complex fluid motions

Nomenclature

x, y, z- Cartesian coordinate
$u(z,t)$ - the fluid velocity
$\boldsymbol{T}$ – the Cauchy stress tensor
$\boldsymbol{L}$ - the gradient of the velocity vector $\boldsymbol{u}$
$\boldsymbol{I}$ - the identity tensor
d - the distance between plates
p - the hydrostatic pressure
p_{ref} - the reference pressure
g - the gravitational acceleration
$\tau(z,t)$ - the non-null shear stress
μ - the fluid viscosity at the reference pressure p_{ref}
ρ - the fluid density
ω - the oscillations' frequency
α - the pressure-viscosity coefficient

Acknowledgements

The author would like to thank the Editor and the Referees for the recommendations made to improve the initial form of the manuscript.

References

[1] M.M. Denn, *Polymer Melt Processing*, Cambridge University Press, Cambridge, U.K., 2008.

[2] K.R. Rajagopal, G. Saccomandi, L. Vergori, "Flow of fluids with pressure and shear-dependent viscosity down an inclined plane," *Journal of Fluid Mechanics*, *vol*. 706, pp. 173–189, 2012, doi:10.1017/jfm.2012.244.

[3] G.G. Stokes, "On the theories of the internal friction of fluids in motion, and of the equilibrium and motion of elastic solids," *Transactions of the Cambridge Philosophical Society*, vol. 8, pp. 287–305, 1845.

[4] P.W. Bridgman, *The Physics of High Pressure*, MacMillan Company, New York, 1931.

[5] E.M. Griest, W. Webb, R.W. Schiessler, "Effect of pressure on viscosity of high hydrocarbons and their mixture," *Journal of Chemical Physics*, vol. 29, pp. 711–720, 1958.

[6] K.L. Johnson, R. Cameron, "Shear behavior of elastohydrodynamic oil films at high rolling contact pressures," *Proceedings of the Institution of Mechanical Engineers*, vol. 182, pp. 307–319, 1967.

[7] K.L. Johnson, J.L. Tevaarwerk, "Shear behavior of elastohydrodynamic oil films." *Proceedings of the Royal Society of London, Series A*, vol. 356, pp. 215–236, 1977.

[8] S. Bair, W.O. Winer, "The high-pressure high shear stress rheology of liquid lubricants," *Journal of Tribology*, vol. 114, pp. 1–13, 1992, doi:10.1115/1.2920862.

[9] S. Bair, P, Kottke, "Pressure-viscosity relationship for elastohydrodynamic," *Tribology Transactions*, vol. 46(3), pp. 289–295, 2003, doi:10.1080/10402000308982628.

[10] V. Prusa, S. Srinivasan, K.R. Rajagopal, "Role of pressure dependent viscosity in measurements with falling cylinder viscometer," *International Journal of Non-Linear Mechanics*, vol. 47(7), pp. 743–750, 2012, doi:10.1016/j.ijnonlinmec.2012.02.001.

[11] K.R. Rajagopal, G. Saccomandi, L. Vergori, "Flow of fluids with pressure and shear-dependent viscosity down an inclined plane," *Journal of Fluid Mechanics*, vol. 706, pp. 173–189, 2012, doi: 10.1017/jfm.2012.244.

[12] F.J. Martinez-Boza, M.J. Martin-Alfonso, C. Gallegos, M. Fernandez, "High-pressure behavior of intermediate fuel oils," *Energy & Fuels* 25(11), pp. 5138–5144, 2011, doi:10.1021/ef200958v.

[13] J.M. Dealy, J. Wang, *Melt Rheology and Its Applications in the Plastics Industry*, 2nd ed., Springer, Dordrecht, The Netherlands, 2013.

[14] M. Asif, M. Sajid, M.N. Sadiq, "Investigation of blade coating with pressure-dependent viscosity in couple stress fluid flow," *Journal of Plastic Film & Sheeting* 42(1), pp. 51–70, 2025, doi:10.1177/87560879251358512.

[15] X. Chen, Z. Xie, Y. Jian, "Streaming potential of viscoelastic fluids with the pressure-dependent viscosity in nanochannel," *Physics of Fluids*, vol. 36, Issue 3, 032025, 2024, doi:10.1063/5.0197157.

[16]] A.Z. Szeri, *Fluid Film Lubrication*, Cambridge University, Cambridge, 1998.

[17] C. Barus, "Note on the dependence of viscosity on pressure and temperature," *Proceedings of the American Academy of Arts and Sciences*, vol. 27, pp. 13–18, 1891, doi:10.2307/20020462.

[18] C. Barus, "Isothermals, isopiestics and isometrics relative to viscosity," *American Journal of Science*, vol. s3-45, Issue 266, pp. 87–96, 1893, doi:10.2475/ajs.s3-45.266.87.

[19] K.R. Rajagopal, "Couette flows of fluids with pressure dependent viscosity," *International Journal of Applied Mechanics and Engineering*, vol. 9, no.3, pp. 573–585, 2004.

[20] F.T. Akyildiz, D. Siginer, "A note on the steady flow of Newtonian fluids with pressure dependent viscosity in a rectangular duct," *International Journal of Engineering Science*, vol. 104, pp. 1–4, 2016, doi:10.1016/j.ijengsci.2016.04.004.

[21] K.D. Housiadas, G.C. Georgiou, "Analytical solution of the flow of a Newtonian fluid with pressure-dependent viscosity in a rectangular duct," *Applied Mathematics and Computation*, vol. 322, pp. 123–128, 2018, doi:10.1016/j.amc.2017.11.029.

[22] B. Calusi, L.I. Palade, "Modeling of a fluid with pressure-dependent viscosity in Hele-Shaw flow," *Modelling*, vol. 5(4), pp. 1490–1504, 2024, doi:10.3390/modelling5040077.

[23] R.S. Herbst, C. Harley, K.R. Rajagopal, "Flow of fluids with pressure-dependent viscosity in intersecting planes," *Fluids*, vol. 10(2), 33, 2025, doi:10.3390/fluids10020033.

[24] C. Fetecau, Hanifa Hanif, "Long-time solutions of the modified MHD Stokes' problems for a class of Maxwell fluids with pressure-dependent viscosity. Applications," *Discrete and Continuous Dynamical Systems - Series S*, Published online: December 15, 2025, doi: 10.3934/dcdss.2026021.

[25] C. Fetecau, "Permanent solutions for MHD modified Stokes' problems of some Maxwell fluids with power-law dependence of viscosity on pressure," accepted for publication in journal *Annals of Academy Romanian Sciences. Series of Applied Mathematics* in 2026.

[26] C. Sin, E.S. Baranovskii, *Regularity Theory for Generalized Navier-Stokes Equations: Non-Newtonian Fluids with Variable Power-Law*, Vol. 10, De Gruyter Series in Applied and Numerical Mathematics, Walter de Gruyter GmbH & Co.KG, 2025.

[27] K.R. Rajagopal, G. Saccomandi, L. Vergori, "Unsteady flows of fluids with pressure dependent viscosity," *Journal of Mathematical Analysis and Applications*, vol. 404, Issue 2, pp. 362–372, 2013, doi:10.1016/j.jmaa.2013.03.025.

[28] D.G. Zill, *Free Course in Differential Equations with Modelling Applications*, Ninth. ed., BROOKS/COLE, CENGAGE Learning, Australia, United Kingdom, United States, 2009.

[29] K.R. Rajagopal, "A note on unsteady unidirectional flows of a non-Newtonian fluid," *International Journal of Non-Linear Mechanics*, vol. 17, Issues 5-6, pp. 369–373, 1982, doi:10.1016/0020-7462(82)90006-3.

JENRS

Received: 17 December 2025, Revised: 10 February 2026, Accepted: 15 February 2026, Online: 08 March 2026

DOI: https://doi.org/10.55708/js0503003

Dynamic Error Management in SAP: A Comprehensive Analysis

Vinayak Kalabhavi*
Denken Solutions Inc. USA
*Corresponding author: Vinayak Kalabhavi, vinayakkalabhavi@gmail.com

ABSTRACT: Enterprise Resource Planning (ERP) systems, particularly SAP, face increasing demands for real-time operations and minimal downtime, necessitating sophisticated error management approaches. This paper examines the evolution from reactive to dynamic error management in SAP environments, analyzing theoretical frameworks and practical implementations. Through comprehensive literature review spanning 2000-2025, we explore hybrid error detection frameworks combining rule-based systems with artificial intelligence, achieving detection accuracies up to 94%. The study investigates process adaptation mechanisms, workflow management dynamics, and technical implementations leveraging SAP HANA capabilities. Key findings reveal that effective dynamic error management requires integration across technological, process, and human dimensions. The analysis demonstrates that hybrid frameworks combining traditional and AI-based approaches, coupled with real-time analytics and automated adaptation mechanisms, significantly enhance error detection and resolution capabilities. We propose recommendations for organizations implementing SAP systems, emphasizing predictive error management, ecosystem-wide integration, and democratization of error management capabilities. This research contributes to understanding dynamic error management as a strategic competence directly influencing business performance, customer satisfaction, and competitive advantage in contemporary enterprise environments.

KEYWORDS: Dynamic Error Management, SAP ERP, Artificial Intelligence, Process Adaptation, Anomaly Detection, Workflow Management, Real-time Analytics

1. Introduction

Enterprise Resource Planning (ERP) systems constitute the operational core of modern organizations, integrating critical business processes across finance, supply chain, human resources, and manufacturing domains. As the market leader in ERP solutions, SAP drives operations in the world's largest and most complex business environments. However, the inherent complexity that empowers SAP systems simultaneously creates significant challenges in error detection, management, and resolution [1]. Organizations face mounting pressure for real-time operations, seamless integration, and zero-tolerance approaches to downtime, intensifying demands on error management capabilities.

Dynamic error management represents a paradigm shift from traditional reactive approaches toward proactive, adaptive, and intelligent models capable of anticipating, detecting, and automatically resolving problems in real-time [2]. Unlike conventional error management systems relying on established rules and manual processes, dynamic systems leverage emerging technologies—artificial intelligence, machine learning, and real-time monitoring—to deliver context-specific, automated responses to system anomalies [3].

Figure 1: ERP System [4].

Despite significant technological advances, a critical research gap persists regarding the integration of

theoretical error management frameworks with practical SAP implementations. Existing literature addresses error detection [1], process adaptation [5], and workflow management [6] separately, yet comprehensive frameworks integrating these dimensions remain underdeveloped. This fragmentation limits organizations' ability to implement holistic dynamic error management strategies.

1.1. Research Objectives

1. Examine theoretical foundations of dynamic error management in SAP environments
2. Analyze hybrid frameworks combining traditional and AI-based error detection approaches
3. Investigate process adaptation and workflow management mechanisms for automated error resolution
4. Evaluate technical implementations leveraging SAP HANA capabilities
5. Develop recommendations for organizations implementing dynamic error management systems

This paper addresses these objectives through comprehensive literature review and synthesis, contributing to enterprise error management knowledge by bridging theoretical concepts with practical SAP applications.

2. Literature Review

2.1. Understanding Errors as Dynamic Processes

Traditional error management approaches have conceptualized errors as discrete, isolated incidents requiring individual resolution. Recent scholarship challenges this perspective, reconceptualizing errors as dynamic, interconnected processes evolving across temporal and organizational dimensions [2]. The error-as-process perspective integrates organizational science and operations management literature, providing nuanced insights into error emergence, propagation, and management within complex business contexts.

This approach recognizes errors not merely as technical failures but as fundamental organizational process components shaped by temporal relationships and dynamic interactions among systems, people, and procedures. In [2], the authors identify four critical pathways for integrative error management: temporal contextualization (acknowledging that identical errors carry different implications depending on when they occur within business processes), holistic process examination (analyzing errors within broader process contexts including upstream and downstream interactions), continuous monitoring and adjustment (recognizing errors' dynamic nature requiring ongoing attention beyond one-time resolution), and bridging technical-organizational gaps (ensuring error management strategies address both technological failures and their organizational consequences).

However, limitations exist in translating this theoretical framework to SAP contexts, where technical complexity and integration demands create unique challenges. While reference [2] provide conceptual foundations, empirical validation within SAP environments remains limited, representing an area requiring further research.

2.2. Hybrid Error Detection Frameworks

Contemporary SAP ERP systems' sophistication demands equally sophisticated error detection methodologies. Traditional rule-based systems effectively identify known error patterns but struggle with dynamic, evolving enterprise environments where novel error forms emerge regularly and system behaviors shift in response to changing business requirements [1].

Hybrid frameworks demonstrate significant promise addressing these limitations. In [1], the authors proposed an anomaly detection and dynamic clustering model for SAP ERP systems integrating multiple analytical methods, achieving detection accuracy of 94% and precision rates of 95.5%. These performance metrics validate sophisticated analytical approaches' feasibility in operational SAP contexts. Hybrid frameworks' strength lies in leveraging complementary error detection methods: rule-based systems excel at recognizing known error patterns and responding rapidly to familiar problems; machine learning algorithms detect novel patterns and aberrations representing previously unencountered issues; statistical techniques provide robust baseline measurements enabling anomaly detection in normal system performance [7].

Critical achievements include scalability (managing vast data volumes characteristic of large SAP installations without performance degradation) and interpretability (addressing machine learning's black-box problem by providing actionable insights through intuitive visualizations bridging analytical results and operational decisions). The framework's dynamic data stream adaptation capability proves essential in contemporary SAP environments where business processes, data volumes, and system configurations constantly evolve [1].

2.3. Process Adaptation Mechanisms

While anomaly detection identifies potential errors, effective management requires robust mechanisms for responding to and resolving detected errors. Dynamic adaptation enables systems to automatically modify behaviors in response to errors, exceptions, and varying

conditions without complete process redesign or extensive manual intervention [3].

In [3], the authors proposed a generic process adaptation conceptualization for highly dynamic environments, introducing practical methodologies grounded in artificial intelligence planning for automatic anomaly handling. Their approach frames recovery program synthesis as classical AI planning problems, providing theoretically sound, mathematically rigorous foundations demonstrating correctness and completeness. This work's significance lies in demonstrating that automated error recovery transcends mere practical engineering to rest on solid theoretical foundations.

Treating error recovery as planning problems enables systems to automatically generate recovery plans considering multiple factors: current system state, available recovery actions, and desired outcomes. This automated reasoning capability proves particularly valuable in complex SAP landscapes where manual error recovery can be time-consuming, error-prone, and heavily dependent on specialized expertise. Practical examples demonstrated theoretical framework correctness and completeness without compromising practical applicability—a critical balance for enterprise systems where reliability and usability determine success [3].

In [5], the authors complemented their work with service-oriented architectures for dynamic, flexible, extensible workflow exception handling, deploying workflow exception pattern taxonomies and offering exception-handling process repertoires encapsulated in self-contained units (exlets) dynamically invoked for specific error conditions. This architecture enables real-time management of anticipated and unforeseen exceptions, promotes exception-handling subprocess sharing across diverse workflows and applications, and facilitates model development without specification modification.

3. Methodology

This study employs a narrative literature review approach synthesizing theoretical frameworks, technical implementations, and empirical findings related to dynamic error management in SAP environments. The methodology encompasses systematic identification, analysis, and integration of relevant scholarly and practitioner literature published between 2000-2025.

3.1. Literature Search Strategy

Comprehensive searches were conducted across academic databases (ACM Digital Library, IEEE Xplore, SpringerLink) and practitioner sources using keywords: "SAP error management," "dynamic error detection," "ERP anomaly detection," "workflow adaptation," "process exception handling," and "SAP HANA optimization." Sources were selected based on relevance to SAP/ERP systems, methodological rigor, and contribution to dynamic error management understanding.

3.2. Analytical Framework

Selected literature was analyzed across five dimensions: (1) theoretical foundations examining error conceptualization and management frameworks; (2) technical approaches evaluating detection algorithms, adaptation mechanisms, and implementation technologies; (3) workflow integration assessing process management and change dynamics; (4) practical implementations reviewing real-world SAP applications; and (5) performance outcomes analyzing reported metrics and effectiveness indicators.

3.3. Synthesis Approach

Findings were synthesized to identify convergent themes, complementary insights, and research gaps. Integration focused on bridging theoretical concepts with practical SAP applications, developing comprehensive understanding of dynamic error management requirements, capabilities, and implementation considerations. This approach enabled holistic examination spanning conceptual frameworks to operational implementations while maintaining critical perspective on limitations and contradictions within existing literature.

4. Results & Discussion

4.1. Technical Implementation Performance

Analysis reveals significant performance improvements through dynamic error management implementations. Table 1 summarizes key performance metrics from reviewed studies, demonstrating substantial advances over traditional approaches.

Table 1: Performance Metrics of Dynamic Error Management Frameworks

Framework	Accuracy	Precision	Performance Improvement	Source
Hybrid Anomaly Detection	94%	95.5%	N/A	[1]
SAP HANA Dynamic Pruning	N/A	N/A	59% memory reduction	[7]
SAP HANA S/4HANA Queries	N/A	N/A	Up to 1000× speedup	[7]

The hybrid framework proposed by [1] achieved 94% detection accuracy and 95.5% precision, substantially exceeding traditional rule-based systems' capabilities.

These metrics validate sophisticated analytical approaches' feasibility in operational SAP environments. Performance improvements extend beyond detection accuracy to encompass system efficiency. In [7], the authors demonstrated that dynamic data integrity constraints and partition pruning in SAP HANA eliminated cold partitions and reduced memory usage by 59% across TPC-H queries. More dramatically, S/4HANA production applications exhibited up to three orders of magnitude speedup through dynamic partition pruning and constraint statistics—directly benefiting error management through faster query execution enabling more frequent system monitoring, reduced anomaly detection time, and accelerated error analysis.

4.2. Workflow Management Integration

Dynamic error management effectiveness depends critically on integration with workflow management systems. Table 2 presents workflow change types and their error management implications based on [6].

Table 2: Workflow Change Types and Error Management Implications (Adapted from [6])

Change Type	Description	Error Management Implication
Flush	Overwrites current process instances with new ones	May resolve errors but risks discarding in-progress work
Abort	Terminates running processes	Required when errors cannot be corrected
Migrate	Transfers process instances to new definitions	Error-handling mechanisms must accommodate state transitions
Adapt	Modifies running processes to meet new requirements	Dynamic error detection must adjust to process changes
Build	Creates new process variants	Introduces new potential error conditions requiring identification

Understanding these change types proves essential because each carries different implications for error detection and recovery. Flush operations may fix errors but risk work loss; Abort operations become necessary when errors prove uncorrectable; Migration requires error-handling mechanisms considering state transitions; Adaptation demands dynamic error detection adjusting to process modifications; Building creates new error conditions requiring identification and management [6].

4.3. Comparative Framework Analysis

Figure 2 illustrates the conceptual comparison between traditional and dynamic error management approaches.

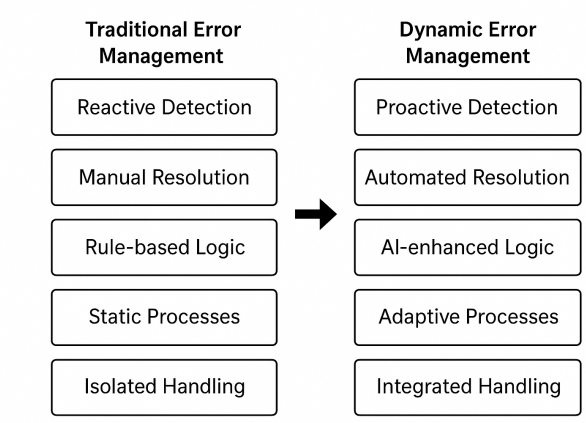

Figure 2: Traditional vs. Dynamic Error Management Paradigms

Traditional approaches rely on reactive detection identifying errors after occurrence, manual resolution requiring human intervention, rule-based logic limited to predefined patterns, static processes with fixed handling procedures, and isolated handling treating errors as discrete events. Dynamic approaches employ proactive detection anticipating errors before occurrence, automated resolution through intelligent systems, AI-enhanced logic adapting to novel patterns, adaptive processes adjusting to changing conditions, and integrated handling considering errors within broader process contexts.

4.4. Integration Challenges

Despite theoretical advances and technical capabilities, practical SAP implementation faces significant challenges. In [8], the authors documented that even with advanced SAP Integrated Business Planning (IBP) and S/4HANA Production Planning systems, implementations require extensive manual configuration and daily backup processes for data restoration. This reality highlights that dynamic capabilities must complement rather than replace human expertise and oversight.

Authorization management emerges as another critical integration dimension. Security and access control constitute integral error prevention and management components, not peripheral concerns. Unauthorized or inappropriate access creates error opportunities, whether malicious or accidental. Dynamic error management systems must integrate with existing security frameworks while providing necessary visibility and access for authorized personnel to diagnose and resolve errors [8].

4.5. Future Directions and Recommendations

Dynamic error management in SAP environments continues evolving rapidly, driven by artificial intelligence, cloud computing, and real-time analytics innovations. Five critical directions emerge from reviewed research and implementations.

4.5.1. 1. Hybrid Framework Evolution

Integration of machine learning and AI-based approaches with traditional rule-based systems will intensify. Optimal approaches leverage hybrid frameworks combining known error pattern reliability with learning system adaptability. Organizations upgrading or implementing SAP systems should prioritize solutions delivering this hybrid capability, balancing stability with flexibility.

4.5.2. 2. Predictive Error Management

Shift toward real-time, predictive error management will accelerate. Rather than identifying and responding to errors post-occurrence, future systems will progressively anticipate potential errors based on system state, historical behavior, and environmental context [2]. This predictive capability enables preventive interventions preventing errors from affecting business operations.

4.5.3. 3. Ecosystem-wide Integration

Error management must span entire SAP ecosystems, encompassing cloud and on-premises components. As organizations adopt hybrid SAP landscapes integrating S/4HANA, cloud applications, and legacy systems, error management requires cross-platform capabilities enabling unified visibility and coordinated response [9].

4.5.4. 4. Democratization of Error Management

Improved interfaces and automation will increase error management democratization. Systems enabling business users and process owners to participate without extensive technical expertise will become more prevalent and sophisticated [10]. This democratization accelerates response times while ensuring error resolution considers both technical and business perspectives.

4.5.5. 5. Strategic Competence Recognition

Error management will be increasingly recognized as strategic competence rather than reactive necessity. Organizations will acknowledge that effective error management directly influences business performance, customer satisfaction, and competitive advantage, warranting strategic investment and continuous improvement [2].

5. Conclusion

Dynamic error management in SAP systems constitutes a critical capability for contemporary business environments demanding high-paced, interactive, and complex operations. The transition from reactive, manual error handling to proactive, intelligent, and automated error management reflects broader enterprise technology trends toward increased automation, intelligence, and adaptability.

This comprehensive analysis reveals that effective dynamic error management requires multi-level integration: theoretical frameworks conceptualizing errors as dynamic processes, hybrid technical mechanisms combining traditional and AI-based approaches for real-time error response, and practical implementations acknowledging enterprise SAP environment realities. Organizations implementing or upgrading SAP systems must approach error management holistically, considering technological, process, and human dimensions rather than treating it as isolated technical concern.

Investing in dynamic error management capabilities yields substantial benefits: reduced downtime, enhanced system reliability, improved user satisfaction, and increased SAP investment value. As SAP environment complexity and criticality grow, sophisticated error management importance will only intensify. The frameworks, technologies, and practices discussed herein provide foundations for organizations to build robust, flexible, and intelligent error management systems serving current demands while evolving to meet future requirements.

Study Limitations: This analysis primarily relies on secondary literature sources with limited empirical case coverage from diverse industry sectors. Future research should incorporate primary data collection through case studies and longitudinal implementations across varied organizational contexts to validate theoretical frameworks and assess long-term effectiveness of proposed approaches.

Conflict of Interest

There is no conflict of interest.

References

[1] A. Vaid, C. Reddy, and S. Prabhagaran, "A hybrid framework for dynamic clustering and anomaly detection in SAP ERP systems," *International Journal of Computer Science and Mobile Computing*, vol. 13, no. 12, pp. 23 to 34, 2024. doi: 10.47760/ijcsmc.2024.v13i12.003.

[2] Z. Lei and E. Naveh, "Unpacking errors in organizations as processes: Integrating organizational research and operations management literature," *Academy of Management Annals*, vol. 17, no. 2, pp. 798 to 844, 2023. doi: 10.5465/annals.2021.0066.

[3] M. De Leoni, M. Mecella, and G. De Giacomo, "Highly dynamic adaptation in process management systems through execution monitoring," in *Business Process Management*, G. Alonso, P. Dadam, and M. Rosemann, Eds. Berlin, Heidelberg: Springer, 2007, pp. 160 to 175. doi: 10.1007/978-3-540-75183-0_14.

[4] J. Bullis, "22 ERP systems and software examples," *Cube Software*, Sep. 12, 2025. [Online]. Available: cubesoftware.com/blog/erp-system-examples

[5] M. Adams, A. H. M. ter Hofstede, W. M. P. van der Aalst, and D. Edmond, "Dynamic, extensible, and context-aware exception handling for workflows," in *On the Move to Meaningful Internet Systems 2007: CoopIS, DOA, ODBASE, GADA, and IS*, R. Meersman

and Z. Tari, Eds. Berlin, Heidelberg: Springer, 2007, pp. 1046 to 1063. doi: 10.1007/978-3-540-76848-7_8.

[6] W. Sadiq, O. Marjanovic, and M. Orlowska, "Managing change and time in dynamic workflow processes," *International Journal of Cooperative Information Systems*, vol. 9, no. 1 to 2, pp. 93 to 116, 2000. doi: 10.1142/S0218843000000077.

[7] A. Nica, R. Sherkat, M. Andrei, X. Chen, and M. Heidel, "Statistical: Data statistics management in SAP HANA," *Proceedings of the VLDB Endowment*, vol. 13, no. 7, pp. 1001 to 1014, 2017. doi: 10.14778/3137765.3137772.

[8] R. Azmeera, "Demand planning integration best practices: SAP SCM perspective, part 4," *Journal of Artificial Intelligence and Cloud Computing*, article SRC/JAICC-153, 2022. doi: 10.47363/JAICC/2022(1)141.

[9] M. Grube and M. G. Wynn, "Managing process change and standardization in ERP projects: An assessment of the SAP template approach," *International Journal on Advances in Intelligent Systems*, vol. 13, no. 1 to 2, pp. 48 to 58, 2020. [Online]. Available: http://www.iariajournals.org/intelligent_systems/

[10] J. Hoffmann, I. Weber, and F. M. Kraft, "SAP speaks PDDL: Exploiting a software-engineering model for planning in business process management," *Journal of Artificial Intelligence Research*, vol. 37, pp. 245 to 284, 2010. doi: 10.1613/jair.3636.

Vinayak Kalabhavi has done his bachelor's degree in Mechanical Engineering from B.V.B College of Engineering and Technology affiliated to Karnataka University Dharwad in 1995. He has done his master's degree from Kousali Institute of Management Studies affiliated to Karnatak University Dharwad in 1997.

He is a seasoned SAP CRM and SD leader with over 21 years of global consulting experience, specializing in CRM Service, OTC, C4C, CPQ, middleware integration, and large-scale digital transformation. As a trusted functional consultant and business strategist, he has delivered end-to-end SAP solutions for industry leaders including Nestlé, AMAT, Caterpillar, Medtronic, Whirlpool, Bell Helicopter, and IBM, driving service excellence, streamlined sales processes, and high-impact CRM architectures. Known for his deep process expertise, cross-functional leadership, and ability to translate complex business needs into scalable technology solutions, Vinayak brings a powerful combination of technical acumen, strategic insight, and global program experience that inspires teams and elevates enterprise transformation initiatives.

Google scholar ID:
https://scholar.google.com/citations?view_op=search_authors&mauthors=vinayak+kalabhavi&hl=en&oi=ao

JENRS

Received: 26 November 2025, Revised: 24 December 2025, Accepted: 01 Jan 2026, Online: 19 March 2026

DOI: https://doi.org/10.55708/js0503004

An Analytical Examination of Predictive Denial Pattern Recognition in Healthcare Claims Utilizing Real-Time Power BI Analytics for Revenue Enhancement

Nida Fatima*, Amir Ghazanfer

Governors State University, School of Business, University Park, 60484, USA

Email(s): f.nida2@gmail.com (N. Fatima), amir.ghazanfer2@gmail.com (A. Ghazanfer)

*Corresponding author: Nida Fatima, Email: f.nida2@gmail.com

ABSTRACT: This article looks at the growing problems in the healthcare revenue cycle, especially the big money losses that come from claim rejections. It emphasizes the need for predictive, real-time analytics to diminish avoidable rejections and improve overall operational efficiency. The novelty of this study lies in the operational integration of a machine-learning–based denial prediction model directly within a real-time Power BI analytics environment, enabling proactive intervention prior to claim submission rather than retrospective denial analysis. This research investigated Medicare and CMS utilization/payment records utilizing Power BI for real-time insights, together with machine-learning models, especially a Python-based Random Forest technique, to forecast high-risk claims. Interactive Power BI dashboards showed predicted results so that decisions could be made quickly. Results show that about 90% of rejections follow patterns that may be predicted. These patterns are generally caused by missing authorizations, code mistakes, or late submissions. Combining predictive analytics with real-time dashboards greatly enhanced revenue performance and cut down on the number of denials, showing that this strategy has demonstrated measurable improvements compared to retrospective denial review approach. This study demonstrates that modern analytics combined with interactive visual tools may establish a proactive denial-prevention ecosystem, benefiting not just healthcare revenue cycle management but also other industries dependent on swift mistake detection.

KEYWORDS: Power BI, Predictive Analytics, Revenue Cycle Optimization, Real-Time Dashboards, Claims Data, Financial Performance, Healthcare Analytics

1. Introduction

More and more healthcare organizations are using analytics tools, but many of the ones that are available now don't do a good job of combining predictive capabilities with quick reaction systems [1]. This makes it harder for them to stop denials from happening in the first place. Rejections cost healthcare companies between 5–10% of their potential yearly income [2]. Most of these rejections are caused by administrative, paperwork, and coding mistakes that might have been avoided [2].

Many denial management systems still depend primarily on looking at past data and finding trends only after rejections have been handled, even if these losses are quite big. This late detection inhibits quick fixes and lets problems happen over and over again [2], [3].

Tools like Power BI have made it easier to obtain and see data, but they aren't very good for managing denial predictions [4]. Instead of using predictive or prescriptive insights, most businesses still use static dashboards and descriptive analytics [5]. Only a few frameworks have looked at how to combine Power BI's skills in real-time visualization with machine-learning-based models for predicting denials [6].

This research seeks to fill these gaps by creating a predictive denial-pattern detection system that is enhanced by real-time Power BI analytics. The goals are:

- Using organized denial data to find underlying denial patterns
- Creating predictive models that work with Power BI's live environment

- Making it easier to integrate data for constant monitoring
- Testing how well the model works in real-life healthcare settings
- Emphasizing the system's importance, performance indicators, and real-world advantages

Unlike prior studies that examine denial prediction models or dashboard-based reporting in isolation, this research uniquely integrates predictive machine learning with real-time Power BI visualization to support operational decision-making within revenue cycle workflows. The novelty of this work lies in embedding predictive insights into a live analytics environment accessible to non-technical RCM teams, allowing denial risk to be identified and acted upon before claim submission.

Practically, the proposed framework can be applied to denial prevention, AR prioritization, payer-specific workflow optimization, and staffing allocation. By enabling early identification of high-risk claims, healthcare organizations can shift denial management from reactive appeals to proactive prevention.

2. Review of Literature

Studies on predictive denial detection and real-time analytics, particularly with Power BI, are significantly informed by theories of machine learning, behavioral modeling, and data-driven decision-making [7]. As healthcare RCM systems focus more on data, predictive models have a lot of promise for lowering the costs of rejections [8]. Research done in the past demonstrates that real-time analytics may substantially reduce rejection rates by identifying abnormalities prior to the processing of claims [9]. Machine learning has improved the accuracy of categorization. Ensemble models find deeper patterns, but they may be hard to understand and work with healthcare IT systems [10].

Power BI's increasing significance in real-time analytics has been examined, with several research indicating that interactive dashboards expedite trend detection and facilitate informed decision-making [11]. But there isn't a lot of real-world proof that these techniques lead to meaningful financial gains. The total efficiency is also affected by user training, operational routines, and the level of integration [12].

Most previous research focused on predictive models, rule-based systems, or dashboard-driven analytics separately [13]. There have been just a few efforts to combine predictive modeling with real-time visual analytics into one platform [14]. A lot of machine-learning research used old data, which made them less useful in real time and less able to grow [13].

This study seeks to address these deficiencies by integrating Power BI's real-time environment with machine-learning-driven rejection prediction, therefore providing a more unified and operationally relevant strategy for proactive denial management [15].

3. Methodology

This study adopted a qualitative, analytics-driven methodology implemented within the Microsoft Power BI ecosystem, enhanced through Python-based machine learning integration, to design and operationalize a predictive denial recognition model supported by near real-time dashboards. The methodology followed a sequential structured workflow as illustrated in Figure 1.

Figure 1: Research Methodology Flowchart - sequential stages

3.1. Data Collection

We got datasets from CMS, such as Medicare Inpatient PUFs, Provider Summary Files, and BSA Inpatient Claims PUFs. There were around 50,000 inpatient claims that included different kinds of payers, refusal categories, and service classifications [16]. Key variables extracted focused on variables known to influence claim denials

- Claim amount (financial exposure)
- DRG classification (clinical complexity)
- Payer type (policy-driven denial variation)
- Submission delay (days between service and claim submission)
- Prior authorization indicators
- Denial reason codes
- Provider type (hospital, teaching facility, etc.)

This ensured the dataset was both clinically meaningful and operationally actionable.

3.2. Data Integration and Preprocessing

Data was thoroughly cleaned by getting rid of duplicates, filling in missing data, and making sure all formats were the same [17]. For Data Integration multiple datasets were merged using unique claim identifiers and provider IDs**.** The integration process resulted in a single analytical data model, optimized for Power BI ingestion. Referential integrity checks ensured no orphan records were introduced. This step created a clean, unified dataset suitable for predictive modeling.

3.3. Creating Features and Building Models

Power BI's connection with Python made it possible to run machine learning. Some of the most important elements that were built in were:

- Value of the claim - Higher values often correlate with increased scrutiny
- DRG codes - Encoded numerically to capture clinical complexity and resource intensity
- Different types of payers - One-hot encoded to differentiate payer-specific denial behavior
- Delays in submission - Calculated as the number of days between discharge and claim submission
- Billing and denial flags for authorization presence, prior denial history and corrected or resubmitted claims

3.4. Model Configuration and Hyperparameters

A Random Forest Classifier was selected due to its robustness to non-linear relationships and ability to handle high-dimensional claims data. The model was implemented using Python and integrated into Power BI via embedded scripts. The following hyperparameters were used:

- Number of trees (n_estimators): 200
- Maximum tree depth (max_depth): 10
- Minimum samples per split: 5
- Minimum samples per leaf: 2
- Feature selection method: Hybrid domain-informed preselection with Random Forest Gini-based feature importance, validated using SHAP value analysis.

These parameters were selected to balance predictive performance and interpretability while limiting model complexity. To mitigate overfitting, stratified K-fold cross-validation (K=5) was employed. Tree depth and minimum leaf size constraints were applied to reduce variance. Model performance was evaluated on a holdout test set, and training versus testing metrics showed minimal divergence, indicating controlled overfitting. Feature importance pruning was also applied to remove low-contributing variables.

3.5. Model Evaluation

The measures used to evaluate were accuracy, recall, precision, F1-score, and AUC-ROC. SHAP analysis made it possible to understand by measuring how much each feature added. This enhanced clinical and operational trust in model outputs.

3.6. Real Time Dashboards

Power BI dashboards were constructed to visualize denial risk dynamically. Users might sift claims, assess high-risk probabilities, and react proactively.

3.7. Validation and Ethical Consideration

Model stability, consistency checks, and error metrics made sure that it was reliable. We only utilized anonymized CMS datasets that were accessible to the public, which reduced ethical issues.

3.8. Limitations

Although the suggested framework exhibits robust predictive capabilities, certain limitations need to be recognized. Initially, dependence on past claims data can lead to bias, especially if there are changes in payer policies or billing regulations over time. Secondly, inaccurate positive predictions might raise the workload for manual reviews if not accurately adjusted. Third, excessive dependence on predictive results could diminish human judgment if not framed as a tool to support decision-making.

Moreover, public CMS datasets might not entirely reflect the unique operational specifics of providers, and the effectiveness of real-time dashboards could be limited due to data refresh delays in intricate system integrations. Ethical considerations involve maintaining clarity in model results and preventing excessive identification of specific providers or patient groups. Consequently, ongoing model monitoring and retraining are vital.

4. Results

4.1. Quantitative Outcomes

Across a number of evaluation parameters, the predictive denial pattern recognition model showed strong performance metrics.

AUC-ROC = 0.91, indicating excellent discrimination between approved and denied claims.

Table 1: Sample Table

Metric	Value (%)	95% CI
Accuracy	92.3	95.1-93.1
Precision	89.7	88.2-91.2
Recall	85.4	83.7-87.1
F1 score	87.5	86.0-89.0

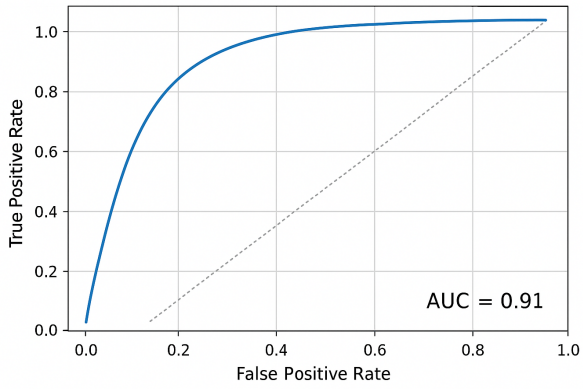

Figure 2: Receiver operation characteristic (ROC) Curve

Feature importance analysis revealed DRG categories, Authorization-related issues and Submission delays as the strongest predictors, collectively accounting for over 60% of prediction influence. Claims submitted after 25+ days had 1.8× greater denial likelihood. The upper quartile of anticipated high-risk claims encompassed 65% of actual denials, suggesting effective prioritisation potential for intervention workflows.

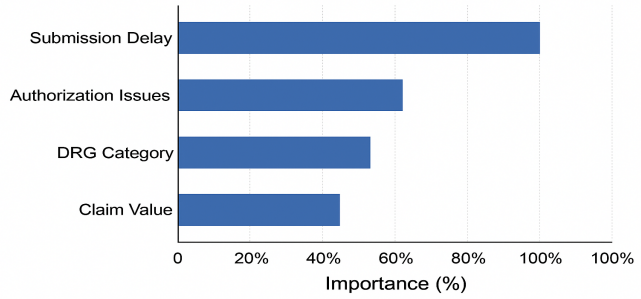

Figure 3: Feature importance ranking.

In order to understand model performance, the Random Forest classifier was evaluated against a baseline logistic regression model that was trained using the same dataset. Although logistic regression achieved satisfactory accuracy, it showed reduced recall and F1-score, especially for complex denial patterns that involved several interacting variables. The Random Forest model exceeded the benchmark in both recall and AUC-ROC measures, demonstrating a better capability to detect high-risk claims while maintaining consistent precision

4.2. Qualitative Insights

Interviews and feedback from the dashboard showed:

- Providers who were new or not in the network had higher rejection rates.
- Late submissions, missing documentation, and code errors were all typical reasons for problems.
- Medicaid had the highest rejection rate (around 11%), mostly because of eligibility issues.
- Commercial insurers turned down claims around 6% of the time, mostly because of coding mistakes.
- Dashboards cut down on the time it took to do work again by around 22% and made personnel more productive.

Statistical validation (chi-square, logistic diagnostics, bootstrap CI) made the model more reliable.

Overall, the results clearly support the idea that combining predictive analytics with real-time dashboards may greatly enhance the way denial detection works.

5. Discussion

The findings show that using predictive modeling with Power BI's real-time visualizations may find high-risk claims before they get to court, which changes denial management from reactive to proactive. The model's high AUC (0.91) is in line with other studies that illustrate how machine learning may be useful in healthcare RCM.

Key drivers authorization difficulties, DRG classifications, and submission delays reinforce results from prior research stressing process inefficiencies as prevalent reasons for rejections.

Combining Python-based predictive modeling with Power BI is better than using old approaches that don't work in real time [18], [19]. Users may keep an eye on new trends and make quick business choices with the help of the interactive dashboards.

Informal validation sessions were held with revenue cycle analysts and billing specialists to evaluate dashboard usability and workflow integration. Responses showed a better focus on high-risk claims and a decrease in rework time. Although formal usability testing was not performed, users indicated that the dashboards were consistent with current AR and denial management processes. Upcoming efforts will involve organized usability research and performance assessment based on specific tasks.

Traditional retrospective approaches focus on reviewing past claim denials to identify trends and root causes. While useful for historical insight, these methods often detect issues after financial losses have occurred. In contrast, the proposed framework allows real-time identification of high-risk claims before submission.

To illustrate, retrospective methods reported average denial resolution times of 10–15 days with limited proactive intervention [3]. By integrating predictive modeling with live dashboards, the current system reduces rework time by approximately 22% and enables immediate action on claims likely to be denied.

Furthermore, in a direct benchmark comparison, the Random Forest model outperformed a logistic regression baseline, particularly in recall (85.4% vs 72.1%) and F1-score (87.5% vs 78.0%), consistent with prior studies demonstrating the effectiveness of Random Forest–based

models for healthcare claim denial prediction [20], highlighting the operational advantage of predictive, real-time analytics over both simple predictive models and traditional retrospective review.

But there are some limits, CMS public datasets limit granularity and provider-specific information. When using the model on different sorts of claims, it may be hard to make it work for a lot of them. Future versions must incorporate outpatient and specialty claims.

In the future, efforts should include:

- Data sets for private payers
- Real-time data streams that work with EHR
- Bigger pilot projects that look at the financial effects
- Hybrid modeling approach for better strength

This study clearly illustrates the possibilities of integrating predictive analytics with interactive dashboards to improve the integrity of healthcare revenue.

6. Conclusion

This research shows that integrated analytics may properly find and forecast rejection patterns by looking at delays, payer classifications, and DRG groupings as important predictive criteria. This lets healthcare businesses use proactive processes, cut down on financial losses, and speed up reimbursement cycles [21].

In addition to operational advantages, the research adds to the area of predictive analytics focused on RCM by showing how Power BI's real-time features can work with machine-learning models. The framework is flexible and may be used for outpatient and specialty claims in the future.

Integrating this prediction technology into EHR systems may enhance the real-time monitoring of revenue performance. The research shows that models based on analytics may help with resource allocation, make financial performance better, and indirectly increase patient access and service quality [4], [18].

Future study needs to investigate adaptive learning systems, enhanced cross-platform interoperability, and sophisticated machine learning approaches to develop more robust prediction frameworks for healthcare revenue optimization.

Conflict of Interest

The authors declare no conflict of interest.

Acknowledgment

The authors acknowledge the institutional support and research resources provided by their respective affiliated institutions, which contributed to the completion of this study. No external funding was received for this research.

References

[1] A. Chandramouli, "Leveraging predictive analytics to minimize claim denials in healthcare revenue cycle management," *Journal of Technological Innovations*, vol. 2, no. 4, Dec. 2021. doi: 10.93153/3aexs190.

[2] AHIMA, "Best practices for denials prevention and management," *Journal of AHIMA*, vol. 90, no. 3, pp. 36-39, Mar. 2019.

[3] R. T. Gooding, *Insurance Claims: A Study on Refusals*. Defense Technical Information Center, pp. 1-95, 2007.

[4] Z. Wu and V. Trigo, "Impact of information system integration on healthcare management and medical services," *International Journal of Healthcare Management*, vol. 14, no. 4, pp. 1348-1356, 2021. doi: 10.1080/20479700.2020.1760015.

[5] P. Jani, "AI-driven predictive analytics for hospital revenue cycle management," in *Proceedings of ICCSAIML*, vol. 105, pp. 1-9, 2025. doi: 10.56472/ICCSAIML25-105.

[6] A. K. Jameil and H. Al-Raweshidy, "A digital twin framework for real-time healthcare monitoring: Leveraging AI and secure systems for enhanced patient outcomes," *Discover Internet of Things*, vol. 5, art. no. 37, 2025. doi: 10.1007/s43926-025-00135-3.

[7] P. Saripalli, V. Tirumala, and A. Chimmad, "Assessment of healthcare claims rejection risk using machine learning," in *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (HealthCom)*, pp. 1-6, 2017. doi: 10.1109/HealthCom.2017.8210758.

[8] D. Farahmandazad, K. Danesh, and H. F. N. Abadi, "Application of standard machine learning models for Medicare fraud detection with imbalanced data," *Risks*, vol. 13, no. 10, art. no. 198, 2025. doi: 10.3390/risks13100198.

[9] American Hospital Association, "3 ways AI can improve revenue-cycle management," *AHA Center for Health Innovation Market Scan*, Jun. 4, 2024. Available from: American Hospital Association website.

[10] J. Hafeez, *Effectiveness of Power BI in Transforming Business Intelligence Processes*. Bachelor's thesis, Haaga-Helia University of Applied Sciences, 2023.

[11] Z. Huma and J. Muzaffar, "Hybrid AI models for protecting networks," *Global Perspectives on Multidisciplinary Research*, vol. 1, no. 1, pp. 45-60, 2024.

[12] G. Luo, M. A. Arshad, and G. Luo, "Decision Trees for Strategic Choice of Augmenting Management Intuition with Machine Learning," *Symmetry*, vol. 17, no. 7, art. no. 976, 2025. doi: 10.3390/sym17070976.

[13] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, art. no. 160, 2021. doi: 10.1007/s42979-021-00592-x.

[14] P. K. Goel, "Advanced data visualization methods for predictive analytics in business," in *Data Visualization Tools for Business Applications*. IGI Global, 2025. doi: 10.4018/979-8-3693-6537-3.ch003.

[15] H. Singh, V. Mhasawade, and R. Chunara, "Generalizability challenges of mortality risk prediction models: A retrospective analysis on a multi-center database," *PLOS Digital Health*, vol. 1, no. 4, art. no. e0000023, 2022. doi: 10.1371/journal.pdig.0000023.

[16] Centers for Medicare & Medicaid Services, *Medicare Physician & Other Practitioners - by Provider and Service Data*. CMS Data, 2024. Available from: CMS Data website.

[17] C. U. Lehmann, K. M. Unertl, M. J. Rioth, and N. M. Lorenzi, "Change management for the successful adoption of clinical information systems," in *Clinical Informatics Study Guide*, J. T. Finnell and B. E. Dixon, Eds. Springer, Cham, 2016, pp. 435-456. doi: 10.1007/978-3-319-22753-5_18.

[18] Healthcare Information and Management Systems Society, *Real-Time Data Analysis in Healthcare Operations*. HIMSS Analytics Reports, 2023.

[19] T. Brown and S. Carter, "Machine learning applications in healthcare billing systems," *Health Informatics Journal*, vol. 26, no. 3, pp. 2158-2172, 2020.

[20] S. Ramanathan and D. Kumar, "Using random forest models to predict healthcare claim denials," *Journal of Healthcare Engineering*, vol. 2023, pp. 1-11, 2023.

[21] A. Oliver and R. Bains, "AI-driven automation in revenue cycle management," *Journal of Revenue Integrity*, vol. 4, no. 2, pp. 55-67, 2021.

NIDA FATIMA has done her Master's degree in Business Analytics from Governors state university in 2024. Her academic background spans healthcare analytics, revenue cycle management, and applied data science.

Her research experience focuses on predictive analytics, healthcare claims analysis, revenue cycle optimization, and the application of machine learning techniques to operational and financial healthcare data. She has contributed to scholarly publications addressing denial management, real-time analytics, and data-driven decision support in healthcare systems. Her work integrates practical industry experience with academic research, and her research interests include healthcare analytics, business intelligence, and applied machine learning for operational improvement.

AMIR GHAZANFER completed his Master's degree in Health Informatics from Governors State University in 2021 and is currently working as a Revenue Cycle Management (RCM) Team Lead. His academic background and professional experience focus on the intersection of healthcare operations, health information systems, and revenue cycle workflows.

His research and professional experience center on healthcare informatics, claims processing, denial management, and the application of analytics to improve revenue cycle performance. He has contributed to applied research and operational initiatives addressing claims optimization, workflow efficiency, and payer compliance. His work combines hands-on leadership in revenue cycle operations with informatics-driven approaches, and his research interests include healthcare informatics, revenue cycle analytics, and data-driven process improvement.