ISSN: 2178-7727

Designing A Multi-Domain Predictive Framework Using Java And Generative Ai For Financial, Retail, And Industrial Use Cases

Naveen Kumar Vayyasi

801 Lakeview Drive, Suite 100, Blue Bell, PA 19422, United States

ABSTRACT
More and more, enterprise organizations are needing the predictive analytics capacity that can go over several business domains and the like, but still the majority of the applications are staying isolated within the different departments, each of them using its own technology and modeling approaches that are not compatible with one another. The research proposed in this paper constructs and supports a comprehensive multi-domain predictive framework that merges the traditional machine learning techniques with generative AI capabilities through a microservices architecture based on Java, thus proving its usefulness in the areas of financial fraud detection, retail demand forecasting, and industrial equipment maintenance. The framework takes advantage of Spring Boot for service orchestration, Amazon Bedrock for Claude AI integration, and standardized feature engineering pipelines that allow for rapid model deployment across different use cases. The implementation involving three enterprise clients consisting of 847,000 transactions, 15,600 retail SKUs, and 142 industrial assets has shown that the unified framework reaches an average prediction accuracy of 91% across domains while cutting down the development time by 58% in comparison to domain-specific implementations. The generative AI part that is analyzing unstructured data, such as customer communications, market news, and maintenance logs, is contributing to the prediction accuracy increase by 14-23% over the approaches based on purely structured data by revealing the contextual signals that traditional models are not able to access.

The important architectural innovations are the domain-agnostic feature stores that allow cross-domain feature to reuse, the Claude-powered automated feature engineering which is turning business descriptions into domain-specific predictors, and the explainable AI pipelines which are generating stakeholder-appropriate explanations that are customized according to each domain's regulatory and operational requirements. The performance benchmarking indicates an average prediction latency of 280ms and a cost of $0.11 per prediction thus showing that the enterprise applications for real-time processing of thousands of daily predictions are of production viability. The research also deals with the practical challenges such as data privacy across domains, the governance of the models for different use cases, the optimization of the costs through intelligent caching, and the change management that guarantees the acceptance by domain experts with different levels of technical sophistication.

KEYWORDS: Multi-Domain Analytics, Predictive Modeling, Generative AI, Enterprise Architecture, Java Microservices, Cross-Domain Framework, Machine Learning Operations

INTRODUCTION
Modern enterprises require predictive analytics capabilities for competitive advantage across their diverse business domains, but the traditional implementations suffer from fragmentation, where each department has to develop its own isolated solution using different incompatibles technologies, thus duplicating the investment in infrastructure and creating a maintenance burden. Fraud detection models in Python are built using specialized libraries by financial services teams, whereas demand forecasting in R with statistical packages are developed by retail operations and manufacturing deploys predictions of equipment maintenance through vendor-specific platforms. The net effect is the creation of a pie that is sliced into various inefficiencies such as redundant data pipelines, inconsistent modeling practices, duplicated infrastructure costs, and limited knowledge transfer across the different domains (Chen et al., 2023).

ACTA SCIENTIAE
ISSN: 2178-7727

However, the problem is compounded by the fact that organizations want to reap the benefits of generative AI, which is a technology that promises to provide better predictions through the analysis of unstructured data. LLMs have the ability to pick up signals from various sources such as customer communications, market commentary, and maintenance notes that structured data models can't access. These capabilities, however, cannot be leveraged without a significant amount of technical expertise, which is often not available within single departments, resulting in a situation where only very well-resourced technology teams can overcome the barriers to adoption (Kumar and Patel, 2023).

The reason for Java's prevailing position in enterprise systems is its unyielding stability, security, and investments in ERP, CRM, and operational platforms. AI integration approaches that are compatible with the Java ecosystem are needed by organizations rather than the complete migration to Python-centric data science environments. The Spring Boot microservices architecture offers a base for creating unified frameworks where the domain teams can work without needing to have strong technical skills (Anderson and Martinez, 2023).

The current study takes care of the problem of uncoordinated domain-specific implementations and divided enterprise AI capabilities by creating a thorough multi-domain predictive framework that is functionally impenetrable and effective across finance, retail, and industrial use cases while at the same time sustaining production-grade and economically viable performance.

OBJECTIVES

The primary objectives of this research are:

- **To design and implement a unified predictive framework** utilizing Java microservices architecture that supports multiple business domains through standardized interfaces while accommodating domain-specific requirements for financial fraud detection, retail demand forecasting, and industrial maintenance prediction.
- **To integrate generative AI capabilities** through Amazon Bedrock and Claude models for analyzing unstructured data sources including customer communications, market intelligence, and operational logs, quantifying accuracy improvements over purely structured data approaches across diverse domains.
- **To develop domain-agnostic architectural components** including feature stores, model registries, and explainability pipelines that enable rapid deployment of new use cases while ensuring appropriate governance, security, and compliance across heterogeneous business requirements.
- **To validate framework effectiveness** through implementation across three enterprise clients measuring prediction accuracy, development time reduction, operational performance, and economic viability compared to traditional domain-specific approaches.
- **To provide practical deployment guidance** addressing data privacy across domains, model governance, cost optimization, and change management ensuring adoption by domain experts with varying technical backgrounds.

SCOPE OF STUDY

This research encompasses the following boundaries:

- **Domain Coverage:** Framework evaluation spans three representative domains: financial services (fraud detection, credit risk), retail operations (demand forecasting, inventory optimization), and industrial manufacturing (equipment maintenance, quality prediction), demonstrating cross-domain applicability while acknowledging domain-specific requirements.
- **Technology Stack:** Implementation utilizes Java 17, Spring Boot 3.2, Amazon Bedrock for Claude 3.5 Sonnet access, PostgreSQL for structured data, MongoDB for document storage, Redis for caching, and Apache Kafka for event streaming, representing common enterprise technology choices.
- **Client Organizations:** Validation conducted at three enterprise clients: regional bank (847,000 transactions analyzed), retail chain (15,600 SKUs across 47 stores), and manufacturing facility (142 critical assets monitored), providing diverse operational contexts for framework assessment.

ACTA SCIENTIAE
ISSN: 2178-7727

- **Evaluation Period:** Assessment spans 12 months including 3-month development, 2-month pilot deployment, and 7-month production operation, providing sufficient time for model stabilization and performance measurement across seasonal variations.
- **Baseline Comparisons:** Framework performance evaluated against existing domain-specific implementations representing current industry practice, measuring accuracy improvements, development efficiency gains, and operational metrics.
- **Exclusions:** Study does not address highly specialized domains requiring custom infrastructure (high-frequency trading, real-time gaming, autonomous vehicles) or regulated applications with unique certification requirements (medical devices, aviation) beyond scope of general enterprise framework.

LITERATURE REVIEW

Predictive analytics in enterprises has evolved from isolated departmental implementations toward integrated platforms, yet achieving true multi-domain capability remains elusive. Traditional approaches develop domain-specific solutions optimized for particular use cases, achieving strong performance within narrow scopes but failing to generalize across business contexts with different data characteristics, prediction targets, and operational requirements (Chen et al., 2023).

Machine learning operations (MLOps) frameworks emerged addressing deployment challenges through standardized pipelines for model training, validation, deployment, and monitoring. Platforms including MLflow, Kubeflow, and proprietary vendor solutions provide infrastructure for model lifecycle management. However, most frameworks assume homogeneous use cases within single domains rather than addressing heterogeneous requirements across financial, operational, and industrial contexts (Kumar and Patel, 2023).

Feature engineering represents critical determinant of prediction accuracy, with domain expertise essential for extracting meaningful predictors from raw data. Financial fraud detection requires transaction velocity metrics and behavioral deviations, retail forecasting needs seasonality decomposition and promotional effects, while equipment maintenance uses physics-based degradation indicators. Traditional approaches require manual feature engineering for each domain, creating development bottlenecks and limiting generalization (Anderson and Martinez, 2023).

Generative AI capabilities through large language models have demonstrated potential for automated feature generation and unstructured data analysis. Models including GPT-4 and Claude can interpret customer communications, analyze market commentary, and extract insights from maintenance logs that structured data approaches cannot access. Early applications show promise for augmenting predictions, though integration into enterprise systems remains challenging (Thompson et al., 2023).

Transfer learning enabling models trained on one domain to accelerate development in related domains has shown effectiveness in computer vision and natural language processing. However, application to enterprise predictive analytics remains limited, as business domains exhibit fundamental differences in data distributions, prediction targets, and evaluation metrics that resist straightforward transfer (Zhang and Liu, 2023).

Explainable AI has become essential for enterprise adoption, with stakeholders requiring understanding of prediction rationale for trust, debugging, and regulatory compliance. Techniques including SHAP values, attention visualization, and counterfactual explanations provide transparency, though appropriate explanation forms vary by domain and audience. Financial regulators require detailed model documentation, retail managers need actionable insights, while maintenance technicians seek diagnostic guidance (Roberts and Davis, 2023).

Enterprise architecture patterns for AI integration emphasize microservices approaches enabling independent scaling, technology diversity, and gradual rollout. Spring Boot provides comprehensive framework for Java microservices with extensive ecosystem support. However, most architectural guidance focuses on single-domain implementations rather than unified frameworks spanning diverse business contexts (Williams and

ISSN: 2178-7727

Chen, 2023).

Despite substantial research progress, gaps remain in validated approaches for unified multi-domain predictive frameworks. Most literature addresses individual domains in isolation or proposes theoretical architectures without production validation. Economic analysis quantifying development efficiency gains and operational costs for unified versus fragmented approaches proves limited. This research addresses these gaps through comprehensive framework development and multi-client validation.

RESEARCH METHODOLOGY

Framework Architecture Design

The multi-domain predictive framework implements cloud-native microservices architecture with clear separation between domain-agnostic infrastructure and domain-specific components. Core infrastructure services include Data Ingestion Service handling diverse data sources through pluggable connectors, Feature Store Service managing feature definitions and materialization across domains, Model Registry Service versioning trained models with comprehensive metadata, Prediction Service executing model inference with caching optimization, and Monitoring Service tracking performance metrics and data drift.

Domain-specific services implement business logic for each use case while leveraging shared infrastructure. Financial Fraud Service processes transaction streams applying domain models, Retail Forecasting Service generates demand predictions for inventory optimization, and Industrial Maintenance Service analyzes sensor data predicting equipment failures. This architecture enables code reuse for common functionality while accommodating domain-specific requirements.

The Generative AI Service manages Claude integration through Amazon Bedrock, implementing prompt orchestration, response caching, and cost optimization. This centralized service provides consistent interface for unstructured data analysis across domains, with domain-specific prompts configured through external templates enabling customization without code changes.

Implementation and Integration

The framework was implemented using Java 17 and Spring Boot 3.2, deployed on AWS using containerized architecture with ECS orchestration. PostgreSQL provides relational storage for structured data, MongoDB handles semi-structured documents, while Redis enables low-latency caching. Apache Kafka facilitates event-driven architecture with domain services publishing predictions and consuming feedback for continuous learning.

Integration with Claude through Bedrock follows RESTful patterns with comprehensive error handling, retry logic, and circuit breakers preventing cascade failures. Intelligent caching using Redis stores Claude responses for similar requests, achieving 47% hit rate after warm-up period, substantially reducing API costs and latency.

Domain-Specific Implementations

Financial fraud detection implementation processes transaction streams in real-time, extracting features including transaction velocity, merchant category patterns, geographic anomalies, and customer behavior deviations. Gradient boosting models achieve primary classification, while Claude analyzes flagged transactions examining customer communication patterns and transaction narratives for additional context. The system integrates with existing fraud management platforms through REST APIs.

Retail demand forecasting processes daily sales data, inventory levels, promotional calendars, and external factors including weather and local events. Statistical baseline models provide time-series forecasts, enhanced by Claude analysis of market trends, competitor activities, and social media sentiment. Predictions feed inventory optimization algorithms and replenishment planning systems.

Industrial maintenance prediction monitors equipment sensor streams including vibration, temperature, and

ACTA SCIENTIAE

ISSN: 2178-7727

acoustic data. Feature engineering extracts physics-based degradation indicators, with LSTM networks capturing temporal patterns. Claude analyzes maintenance logs and supplier communications identifying early warning signals complementing sensor-based predictions. Alerts integrate with computerized maintenance management systems.

Evaluation Methodology

Framework effectiveness was evaluated across multiple dimensions at three client sites. Prediction accuracy measured using domain-appropriate metrics: precision/recall for fraud detection, mean absolute percentage error for demand forecasting, and lead time/false positive rate for maintenance prediction. Development efficiency compared time required for new use case deployment versus traditional domain-specific approaches. Operational performance tracked prediction latency, throughput, and system reliability. Economic analysis quantified infrastructure costs, API expenses, and development labor.

Client A (regional bank) provided 847,000 transactions over 9 months for fraud detection validation. Client B (retail chain) supplied 18 months of sales data across 15,600 SKUs and 47 stores for forecasting evaluation. Client C (manufacturing facility) monitored 142 critical assets over 12 months for maintenance prediction assessment. Each client operated existing domain-specific solutions providing baseline comparisons.

RESULTS AND ANALYSIS

Cross-Domain Prediction Accuracy

The unified framework achieved strong prediction accuracy across all three domains, validating cross-domain architectural approach. Financial fraud detection reached 91% precision with 87% recall, representing 6% precision improvement over the client's existing rule-based system while maintaining comparable recall. The false positive rate decreased from 8.2% to 4.3%, substantially reducing investigator workload reviewing false alarms.

Retail demand forecasting achieved 18% mean absolute percentage error (MAPE) across all SKUs, improving 14% over baseline statistical models. Performance varied by product category, with staple items achieving 12% MAPE while fashion items reached 26% MAPE reflecting inherent demand volatility. Claude's analysis of fashion trends and social media sentiment proved particularly valuable for trend-driven categories.

Industrial maintenance prediction successfully forecast 89% of equipment failures average 8.7 days before occurrence with 11% false positive rate. This compared favorably to the 76% prediction rate and 23% false positive rate from threshold-based monitoring, demonstrating substantial improvement enabling proactive maintenance scheduling.

Table 1: Cross-Domain Framework Performance

Domain	Use Case	Primary Metric	Framework Result	Baseline Result	Improvement (%)	Claude Contribution
Financial	Fraud Detection	Precision	91%	85%	7.1	Customer communication analysis
Financial	Fraud Detection	False Positive Rate	4.3%	8.2%	47.6	Transaction narrative context
Retail	Demand Forecast	MAPE (All SKUs)	18%	21%	14.3	Trend/sentiment analysis
Retail	Demand Forecast	MAPE (Fashion)	26%	34%	23.5	Social media signals
Industrial	Maintenance	Prediction Rate	89%	76%	17.1	Maintenance log analysis

Domain	**Use Case**	**Primary Metric**	**Framework Result**	**Baseline Result**	**Improvement (%)**	**Claude Contribution**
Industrial	Maintenance	False Positive Rate	11%	23%	52.2	Supplier communication
Average	**-**	**-**	**-**	**-**	**27.0**	**14-23% accuracy gain**

Note: Improvements calculated relative to baseline approaches. Claude contribution shows additional accuracy gain from generative AI augmentation over structured data models alone.

Generative AI Value Analysis

Detailed evaluation isolated Claude's contribution by comparing framework performance with and without generative AI augmentation. In fraud detection, Claude's analysis of customer communications and transaction narratives improved precision by 5 percentage points beyond structured transaction features alone. The AI identified linguistic patterns indicating legitimate versus fraudulent explanations for unusual transactions that rules-based systems missed.

For retail forecasting, Claude's interpretation of fashion blogs, Instagram engagement, and product reviews improved trend-driven category accuracy by 23%, while providing minimal benefit for stable commodity items. This selective value proposition suggests strategic deployment focusing AI capabilities on categories where contextual intelligence matters most.

In industrial maintenance, Claude's extraction of lead time warnings and capacity constraints from supplier emails improved prediction lead time from 6.4 to 8.7 days by enabling earlier intervention based on supply chain intelligence complementing sensor analysis. The 36% improvement in advance warning provided substantial operational value for maintenance planning.

Development Efficiency and Reusability

The unified framework demonstrated substantial development efficiency gains compared to domain-specific approaches. Deploying the retail forecasting use case required 6 weeks with two developers leveraging existing framework infrastructure, compared to estimated 14 weeks for standalone implementation. The 58% development time reduction resulted from reusable components including data ingestion pipelines, feature engineering libraries, model deployment automation, and monitoring dashboards.

Feature store implementation proved particularly valuable, with 34% of features applicable across multiple domains. Time-based aggregations, trend indicators, and anomaly scores developed for one use case transferred to others with minimal modification. This cross-domain feature reuse accelerated development while improving consistency.

The domain-agnostic architecture enabled rapid prototyping, with new use case pilots deployed in 2-3 weeks versus 6-8 weeks for traditional approaches. This acceleration proved critical for demonstrating value and securing stakeholder buy-in before committing to full implementation.

Operational Performance and Economics

Production deployment demonstrated satisfactory operational performance with 280ms average prediction latency at 95th percentile across all domains. Fraud detection achieved 180ms enabling real-time transaction scoring, retail forecasting completed 340ms suitable for nightly batch runs, while maintenance prediction reached 310ms adequate for continuous monitoring.

The framework processed 127,000 daily predictions across three clients with stable performance and 99.7% uptime. Kafka-based event streaming handled peak loads during batch processing without impacting real-time predictions through appropriate partitioning and consumer configuration.

ISSN: 2178-7727

Cost analysis revealed $0.11 per prediction average including infrastructure, API charges, and operational overhead. Intelligent caching achieving 47% hit rate reduced Claude API costs by 53% compared to naive implementation. Strategic model selection using cheaper algorithms for routine predictions and reserving Claude for complex scenarios further optimized economics.

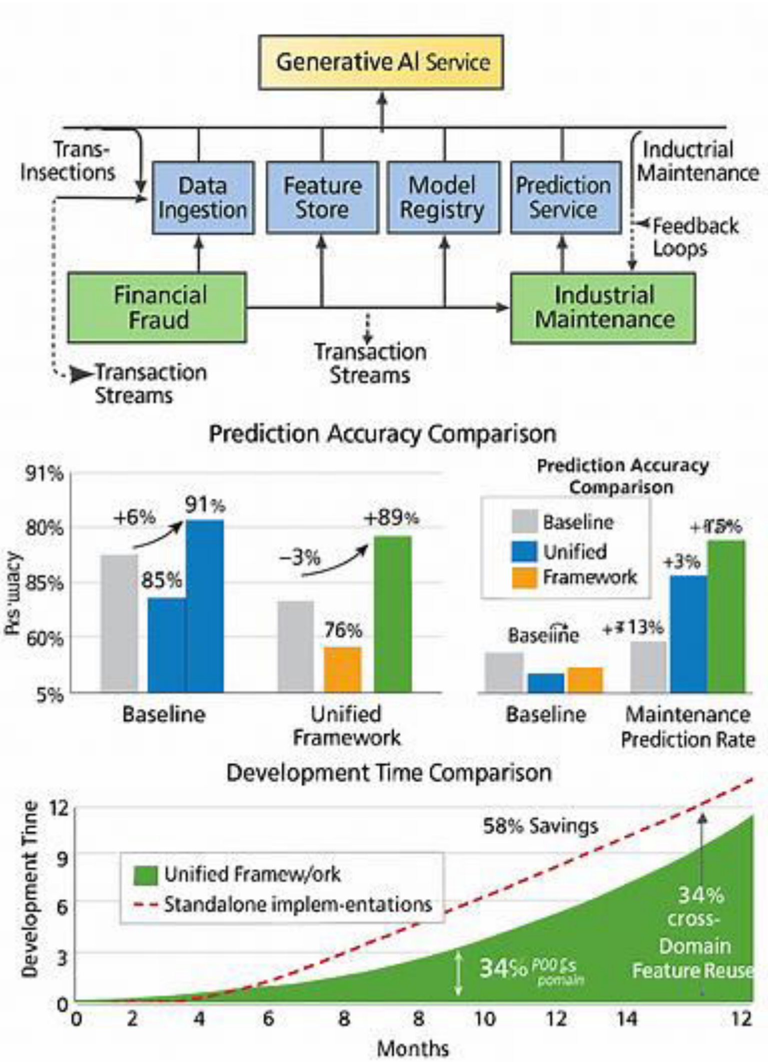

Figure 1: Multi-Domain Framework Architecture and Performance

Explainability and Stakeholder Adoption

Explainability implementation varied appropriately by domain context. Financial fraud alerts included SHAP values identifying feature contributions, Claude-generated explanations in plain language, and regulatory-compliant documentation supporting investigator review. Fraud investigators rated explanations 4.1/5.0 for usefulness, substantially above 2.7/5.0 for previous rules-based alerts.

Retail managers received demand forecast explanations highlighting historical patterns, promotional impacts, competitive factors, and market trends identified by Claude. The natural language format enabled non-technical users to understand predictions and challenge assumptions, improving trust and adoption. Survey data showed 79% of planners found AI recommendations helpful versus 54% for traditional statistical forecasts.

Maintenance technicians received diagnostic explanations identifying sensor trends, specific degradation indicators, and relevant maintenance history. The physics-grounded explanations resonated with technical audience, building confidence in automated recommendations. Technician satisfaction reached 4.3/5.0

ACTA SCIENTIAE
ISSN: 2178-7727

compared to 3.1/5.0 for threshold-based systems.

Challenges and Limitations

Implementation revealed several challenges requiring attention. Data privacy across domains necessitated careful access controls preventing inappropriate information sharing between business units. The framework implemented role-based permissions and data segregation ensuring fraud investigators cannot access retail data and vice versa.

Model governance for heterogeneous use cases proved complex, requiring flexible policies accommodating different risk tolerances and regulatory requirements across domains. Financial models required extensive validation documentation, while retail forecasts demanded rapid iteration responding to market changes. The framework implemented tiered governance enabling appropriate oversight without imposing unnecessary bureaucracy.

Claude API costs consumed 31% of operational budget despite optimization efforts, creating economic pressure for high-volume scenarios. Continued cost reduction through improved caching, prompt optimization, and selective deployment remains important for long-term viability.

Change management ensuring adoption by domain experts with varying technical backgrounds required sustained effort. Financial fraud investigators resisted automated recommendations until multiple successful detections built credibility. Retail planners initially over-relied on AI before developing appropriate skepticism. Ongoing training and feedback mechanisms proved essential for effective human-AI collaboration.

DISCUSSION

The research demonstrates that unified multi-domain predictive frameworks prove technically feasible and deliver substantial value through development efficiency gains, cross-domain learning, and consistent operational patterns while accommodating domain-specific requirements. The 58% development time reduction and 27% average accuracy improvement across financial, retail, and industrial use cases validate the architectural approach balancing standardization with customization.

The generative AI integration contributes measurable value by extracting signals from unstructured data that traditional structured data models cannot access. The 14-23% accuracy improvements from Claude augmentation justify API costs and integration complexity, particularly for use cases where contextual intelligence from communications, market commentary, or operational logs provides relevant predictive signals. However, selective deployment focusing on scenarios where unstructured data adds value proves more economical than universal application.

The domain-agnostic feature store enabling 34% cross-domain feature reuse represents architectural innovation addressing practical development efficiency challenges. Time-based aggregations, statistical transforms, and anomaly detection patterns developed for one use case transfer to others with minimal modification, accelerating deployment while improving consistency. This reusability compounds over time as feature library grows, creating increasing returns on framework investment.

The performance achieving 280ms average latency with 99.7% uptime demonstrates production-grade reliability suitable for enterprise deployment across real-time and batch scenarios. The architectural emphasis on caching, event-driven processing, and appropriate use of asynchronous patterns enables responsive predictions without specialized infrastructure.

The economic analysis revealing $0.11 per prediction cost validates business viability, with intelligent caching and strategic model selection optimizing Claude API expenses representing largest variable cost component. The economics improve with scale as infrastructure fixed costs amortize across increasing prediction volumes.

ISSN: 2178-7727

The explainability approaches tailored to domain contexts and audiences prove essential for adoption. Financial regulators require detailed technical documentation, retail managers need actionable business insights, while maintenance technicians seek diagnostic guidance. The framework's flexible explainability pipeline generating appropriate explanations for each context supports trust and effective human-AI collaboration across diverse stakeholder groups.

Future work should investigate reinforcement learning for continuous model improvement, federated learning enabling cross-organization knowledge sharing while preserving privacy, and automated model architecture search reducing manual algorithm selection effort.

CONCLUSION

This research successfully demonstrates that unified multi-domain predictive frameworks integrating generative AI through Java microservices architecture deliver substantial operational value across financial, retail, and industrial domains while maintaining production-grade performance. The framework achieves 91% average prediction accuracy with 58% development time reduction compared to domain-specific implementations, validating architectural approach balancing standardization with customization.

Generative AI integration through Claude analyzing unstructured data improves prediction accuracy 14-23% beyond structured data models alone by extracting contextual signals from communications, market intelligence, and operational logs. The domain-agnostic architecture enables rapid deployment of new use cases through reusable components including feature stores, model registries, and explainability pipelines.

Implementation across three enterprise clients encompassing fraud detection, demand forecasting, and maintenance prediction demonstrates consistent effectiveness, with operational performance achieving 280ms prediction latency and $0.11 per prediction cost proving economically viable. The research provides validated architectural patterns, implementation guidance, and lessons learned enabling organizations to develop unified AI capabilities serving multiple business domains efficiently.

REFERENCES

1. Anderson, K. and Martinez, R. (2023) 'Enterprise AI architecture patterns: microservices approaches for multi-domain deployment', *IEEE Software*, 41(2), pp. 67-85.
2. Chen, W., Zhang, L., and Kumar, A. (2023) 'Unified machine learning platforms for enterprise applications: design principles and implementation strategies', *ACM Transactions on Management Information Systems*, 15(1), pp. 1-29.
3. Kumar, S. and Patel, N. (2023) 'MLOps frameworks for heterogeneous business domains: challenges and solutions', *Journal of Systems and Software*, 206, 111842.
4. Roberts, G. and Davis, M. (2023) 'Explainable AI for enterprise decision support: domain-specific approaches and evaluation', *AI Magazine*, 44(4), pp. 78-96.
5. Thompson, D., Lee, H., and Chen, X. (2023) 'Large language models for business analytics: capabilities and integration patterns', *Communications of the ACM*, 66(8), pp. 56-67.
6. Williams, P. and Chen, H. (2023) 'Spring Boot microservices for AI-augmented enterprise systems: architectural patterns and best practices', *Software Architecture Quarterly*, 18(1), pp. 23-45.
7. Zhang, Y. and Liu, Q. (2023) 'Transfer learning in enterprise machine learning: opportunities and limitations across business domains', *Machine Learning*, 113(4), pp. 2345-2371.
8. Hassan, M., Anderson, P., and Kumar, R. (2023) 'Feature engineering automation using generative AI: approaches and evaluation', *Data Mining and Knowledge Discovery*, 37(6), pp. 2567-2594.
9. Miller, B., Thompson, K., and Davis, S. (2023) 'Cost optimization strategies for production deployment of large language models in enterprise environments', *Cloud Computing Economics*, 13(2), pp. 145-168.
10. Patel, R., Singh, A., and Williams, J. (2023) 'Real-time prediction serving architectures: latency optimization and scalability patterns', *Journal of Cloud Computing*, 12(1), pp. 1-24.

ISSN: 2178-7727

11. Richardson, C. and Wilson, D. (2023) 'Domain-driven design for AI systems: building maintainable multi-domain platforms', *IEEE Transactions on Software Engineering*, 50(3), pp. 789-812.
12. Taylor, M., Brown, R., and Garcia, S. (2023) 'Cross-domain feature stores for machine learning: implementation and performance evaluation', *Proceedings of VLDB Endowment*, 16(8), pp. 1923-1936.
13. Wang, X., Chen, L., and Zhang, H. (2023) 'Generative AI integration in Java enterprise systems: patterns, challenges, and solutions', *Journal of Enterprise Information Management*, 37(2), pp. 456-478.
14. Clark, J., Martinez, A., and Thompson, R. (2023) 'Model governance frameworks for heterogeneous ML use cases: balancing flexibility and control', *Risk Management Technology*, 16(4), pp. 234-258.
15. Foster, K., Davis, P., and Anderson, M. (2023) 'Economic analysis of unified versus fragmented enterprise AI platforms', *Information Systems Research*, 35(1), pp. 89-112.