Intelligent Cloud Migration Strategies using AI for Enterprise Legacy Application Modernization Frameworks

Cynthiya Mohan*

Project Manager, Amadeus Software labs, London Area, United Kingdom

Abstract

Enterprise organizations increasingly seek to modernize legacy applications by migrating them to cloud environments to improve scalability, flexibility, availability, maintainability, and operational efficiency. However, legacy applications frequently contain tightly coupled architectures, outdated technologies, undocumented dependencies, obsolete interfaces, and business-critical logic that make conventional migration approaches costly and risky. Artificial intelligence provides new opportunities to improve migration decision-making through automated application discovery, dependency analysis, workload classification, modernization-path selection, code transformation, risk prediction, testing, and post-migration optimization. This research proposes an intelligent cloud migration strategy that integrates AI techniques with an enterprise legacy application modernization framework. The proposed approach combines machine learning, natural language processing, generative AI, knowledge graphs, predictive analytics, AI agents, cloud assessment tools, and automated testing to support migration across rehosting, replatforming, refactoring, repurchasing, and rebuilding strategies. The methodology adopts a design science and experimental research approach involving literature analysis, framework development, prototype implementation, controlled migration experiments, and quantitative and qualitative evaluation. The framework evaluates application complexity, dependencies, business criticality, security requirements, performance characteristics, migration cost, and modernization benefits before recommending an appropriate migration strategy. Evaluation focuses on migration effort, cost, application performance, defect rates, modernization coverage, migration risk, downtime, automation level, and decision accuracy. The research aims to establish a systematic and intelligent approach for reducing migration complexity while improving the reliability and business value of enterprise cloud modernization.

Keywords: Intelligent Cloud Migration, Artificial Intelligence, Legacy Application Modernization, Enterprise Cloud Computing, Machine Learning, Generative AI, AI Agents, Application Modernization, Cloud Transformation, Cloud Migration Strategy, Refactoring, Replatforming, Rehosting, Knowledge Graphs, Predictive Analytics, Automated Code Transformation

International Journal of Technology, Management and Humanities (2026)

Introduction

Enterprise organizations continue to operate large portfolios of legacy applications that support essential business processes, customer services, financial operations, supply chains, healthcare systems, manufacturing processes, and internal administration. Many of these applications were developed decades ago using programming languages, databases, operating systems, and architectural patterns that are difficult to maintain or integrate with contemporary digital platforms. Although such applications may remain functionally valuable, their aging technology stacks can increase operational costs, limit scalability, create security challenges, and constrain organizational innovation. Cloud computing provides an important opportunity to address these limitations by enabling elastic infrastructure, managed services, distributed architectures, automated operations, and flexible consumption models. Nevertheless, migrating legacy applications to the cloud is considerably more complex than simply transferring application files from an existing data center to a cloud platform.

Corresponding Author: Cynthiya Mohan, Project Manager, Amadeus Software labs, London Area, United Kingdom

How to cite this article: Mohan, C. (2026). Intelligent Cloud Migration Strategies using AI for Enterprise Legacy Application Modernization Frameworks. *International Journal of Technology, Management and Humanities,* 12(3), 10-20.

Source of support: Nil

Conflict of interest: None

Legacy systems frequently contain undocumented dependencies, tightly coupled modules, proprietary interfaces, outdated databases, hard-coded configurations, and business rules embedded within source code. Organizations may also lack complete knowledge of how applications interact with other systems, infrastructure components, users, and external services. These characteristics make migration planning highly dependent on expert knowledge. Incorrect assumptions about dependencies can lead to service disruption, data inconsistencies, performance degradation, security vulnerabilities, or unexpected migration costs.

Traditional cloud migration methodologies generally involve application inventory, dependency assessment, workload categorization, migration planning, execution, testing, and optimization. Common migration strategies include rehosting, in which applications are moved with minimal modification; replatforming, in which selected infrastructure components are modernized; refactoring, in which application architecture or code is substantially redesigned; repurchasing, in which existing applications are replaced with cloud-based software; and rebuilding, in which applications are reconstructed using modern technologies. Selecting the appropriate strategy for each application is a complex decision because technical feasibility must be balanced against business value, cost, risk, regulatory requirements, application criticality, and modernization objectives.

Artificial intelligence can potentially transform this decision-making process. Machine learning algorithms can analyze application inventories, operational telemetry, source code, dependency relationships, incident records, and historical migration information to classify workloads and predict migration complexity. Natural language processing can extract business rules and technical information from documentation, source-code comments, tickets, and operational records. Generative AI can assist with code translation, documentation generation, configuration conversion, test creation, and modernization planning. AI agents can coordinate multiple migration activities and interact with cloud tools under appropriate governance controls.

The central problem addressed by this research is how AI can be systematically integrated into enterprise cloud migration and legacy application modernization to improve migration decisions and reduce technical and operational risks. The research proposes an intelligent framework capable of assessing legacy applications, recommending appropriate modernization strategies, supporting migration execution, and continuously evaluating post-migration outcomes. Rather than replacing human experts, the framework is designed to augment their decision-making and automate repetitive, data-intensive activities.

The proposed research investigates the relationship between application characteristics, AI-based migration recommendations, modernization outcomes, and enterprise migration performance. It seeks to determine whether intelligent automation can reduce migration effort, improve strategy selection, increase modernization coverage, and reduce defects and operational disruption. The study also considers governance, explainability, security, data privacy, and human oversight because AI-generated migration decisions can have significant consequences for critical enterprise applications.

Literature Review

The literature surrounding cloud migration has traditionally focused on migration models, workload classification, application portfolio assessment, and migration strategy selection. Cloud adoption frameworks commonly categorize applications according to their technical characteristics and recommend approaches such as rehosting, replatforming, refactoring, repurchasing, retaining, or retiring. These strategies provide organizations with a structured vocabulary for migration planning, but the selection process frequently depends on manual assessment by enterprise architects and migration specialists. As application portfolios grow, manual analysis becomes increasingly time-consuming and difficult to maintain.

Application modernization research emphasizes the transformation of legacy systems into architectures better suited to contemporary computing environments. Modernization can involve code conversion, architectural restructuring, database modernization, API enablement, containerization, microservice decomposition, and replacement of obsolete components. Refactoring is generally associated with modifying internal application architecture without changing core functionality, whereas rebuilding involves developing a new implementation around existing business requirements. The appropriate approach depends heavily on application complexity, business criticality, technology constraints, and expected return on investment.

Dependency analysis represents a particularly important research area. Legacy applications rarely operate independently. They may communicate with databases, file systems, middleware, identity services, external applications, batch-processing systems, and proprietary platforms. Migration of one component can therefore affect many others. Traditional dependency discovery uses architecture documentation, configuration analysis, network traffic inspection, source-code analysis, and interviews with subject-matter experts. However, incomplete documentation can make this process difficult. AI-based approaches can potentially correlate heterogeneous information sources to construct more comprehensive dependency representations.

Machine learning has increasingly been applied to software engineering and cloud management problems. Classification models can categorize applications according to complexity, migration readiness, or infrastructure requirements. Regression and time-series models can predict migration costs, resource consumption, performance, and potential failure conditions. Anomaly-detection algorithms can identify unusual application behavior during migration testing. These capabilities provide opportunities for evidence-based migration planning rather than relying exclusively on static assessments.

Natural language processing is also relevant because enterprise migration projects generate large quantities of unstructured information. Technical documentation, architecture descriptions, support tickets, incident reports, business requirements, source-code comments, and operational manuals contain information about application behavior and dependencies. NLP systems can extract entities, relationships, business rules, and technical constraints from these sources. Knowledge graphs can then represent relationships between applications, services, databases, infrastructure components, business processes, and owners. Such representations can provide a foundation for intelligent migration analysis.

Recent advances in generative AI and large language models have expanded the potential for automated modernization. AI models can analyze source code, explain legacy functions, generate documentation, translate code between programming languages, create infrastructure configurations, propose architectural improvements, and generate test cases. These capabilities are particularly relevant to legacy environments where documentation is incomplete and skilled developers familiar with obsolete technologies may be difficult to find. However, generated code may contain defects, introduce security weaknesses, or misunderstand undocumented business logic. Consequently, AI-generated modernization outputs require automated validation and human review.

AI-agent architectures offer an additional opportunity to coordinate migration activities. Instead of using a single AI model, specialized agents can be assigned to application discovery, dependency analysis, architecture assessment, code modernization, testing, security validation, deployment, and monitoring. A coordinating agent can integrate their findings and generate migration recommendations. This approach can potentially create an intelligent migration workflow that continuously observes application state and adjusts migration activities according to new evidence.

Cloud-native technologies also influence modernization decisions. Containers, Kubernetes, managed databases, serverless platforms, API gateways, event-driven architectures, and managed messaging services can reduce operational responsibilities and improve scalability. However, not every legacy application should immediately be transformed into a cloud-native microservice architecture. Excessive modernization can increase cost and complexity. Research therefore increasingly supports incremental modernization, in which applications are transformed according to business value and technical feasibility rather than adopting cloud-native technologies indiscriminately.

Despite these developments, an important research gap remains. Existing cloud migration frameworks generally provide strategic guidance but limited AI-driven decision automation, while AI-based modernization research frequently focuses on isolated activities such as code translation or defect detection. There is comparatively less emphasis on an integrated framework connecting application discovery, dependency analysis, migration-strategy selection, modernization execution, validation, and post-migration optimization. Furthermore, many approaches provide insufficient attention to explainability and governance when AI recommendations affect business-critical systems. This research addresses these limitations by developing an integrated intelligent migration framework that combines AI-based analysis, generative modernization, cloud-native technologies, and controlled autonomous decision-making.

Research Methodology

The research methodology adopts a design science and experimental research approach to develop and evaluate an intelligent framework for enterprise legacy application cloud migration. The methodology consists of requirements identification, literature analysis, framework architecture design, dataset construction, prototype implementation, controlled migration experiments, performance evaluation, and qualitative validation. The objective is to determine whether AI-supported migration can produce more accurate, efficient, and reliable modernization decisions than conventional assessment approaches.

The first stage involves identifying the functional and non-functional requirements of an intelligent migration framework. Functional requirements include automated application discovery, workload profiling, dependency identification, application classification, migration-strategy recommendation, modernization planning, code analysis, test generation, security assessment, migration execution support, and post-migration monitoring. Non-functional requirements include scalability, reliability, explainability, security, interoperability, auditability, data privacy, and human oversight. These requirements are derived from the literature and from the characteristics of enterprise legacy modernization projects.

The second stage involves constructing a representative application portfolio for experimentation. The portfolio can contain applications representing different programming languages, architectural styles, database technologies, deployment models, and business criticality levels. Applications may include monolithic systems, multi-tier applications, batch-processing workloads, API-based systems, and tightly coupled legacy applications. Each application is characterized using attributes such as code size, programming language, dependency count, database complexity, infrastructure requirements, transaction volume, technical debt, business criticality, security classification, operational cost, and availability requirements.

The research then constructs a migration knowledge base. Information from source-code repositories, architecture documents, configuration files, deployment manifests, network observations, application logs, infrastructure inventories, and operational records is processed to identify relationships between application components. Natural language processing is used to extract relevant information from unstructured documentation. Static and dynamic analysis techniques are used to identify technical dependencies. The resulting information is represented using a knowledge graph linking applications, modules, databases, services, infrastructure resources, business processes, interfaces, and operational dependencies.

The intelligent framework contains several AI components. An application assessment model evaluates the technical and business characteristics of each workload. A dependency-analysis component identifies direct and indirect relationships between application components. A migration-risk model estimates the probability of problems associated with different migration strategies. A strategy recommendation engine evaluates candidate approaches such as rehosting, replatforming, refactoring, rebuilding, repurchasing, retaining, or retiring. The recommendation is based on multiple factors rather than a single technical metric.

A multi-objective decision model is used to balance competing migration objectives. These objectives include migration cost, modernization benefit, technical risk, business disruption, performance, scalability, security, maintainability, and expected long-term operational cost. Each application receives a weighted assessment based on organizational priorities. For example, a highly regulated financial application may prioritize security, availability, and auditability, whereas an internal analytical application may place greater emphasis on cost and scalability.

Machine-learning models are trained to support application classification and risk prediction. Historical or experimentally generated migration records can be used to construct datasets containing application characteristics and migration outcomes. Classification models can predict whether an application is suitable for rehosting, replatforming, or deeper modernization. Regression models can estimate migration effort, cost, or duration. Model performance is evaluated using appropriate measures such as accuracy, precision, recall, F1-score, mean absolute error, and root mean square error.

Generative AI is incorporated into the modernization process to assist with code and documentation transformation. A code-analysis agent examines legacy source code and identifies obsolete constructs, architectural dependencies, potential defects, and modernization opportunities. A transformation agent generates candidate code modifications, API interfaces, configuration files, infrastructure definitions, or documentation. A testing agent generates unit, integration, regression, and compatibility tests based on existing application behavior. Generated outputs are not automatically considered correct; they pass through validation mechanisms before being accepted.

An AI-agent orchestration layer coordinates the different modernization activities. The agents operate through controlled tools that provide access to source repositories, analysis systems, testing environments, cloud platforms, infrastructure-as-code systems, and observability platforms. The orchestration process follows an observe, analyze, plan, act, and validate cycle. The system first collects information about an application, analyzes its characteristics and dependencies, develops candidate migration plans, executes approved actions in a controlled environment, and validates the resulting application behavior.

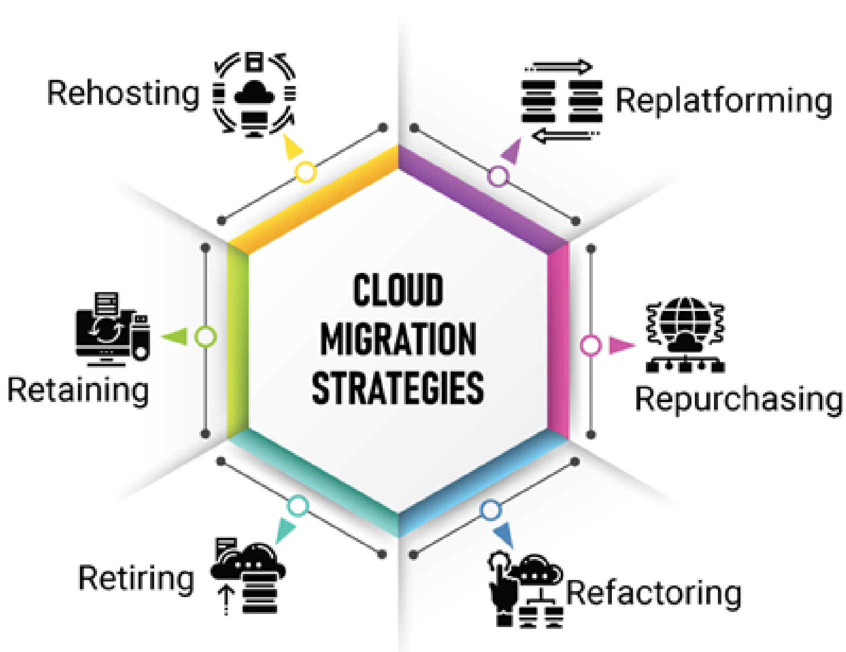

Fig. 1: Cloud Migration strategies

Human oversight is incorporated as a core methodological principle. The system provides recommendations and evidence rather than assuming unrestricted authority over production migration activities. Low-risk actions, such as generating documentation or creating test cases, can be automated. Medium-risk actions, such as modifying non-production infrastructure, can require automated policy validation. High-risk actions involving production databases, security configurations, or business-critical services require explicit human approval. This risk-based approach enables the study to investigate practical enterprise autonomy while reducing the possibility of uncontrolled AI decisions.

The experimental environment consists of a controlled cloud platform containing both legacy and modernized application deployments. Infrastructure-as-code practices are used to ensure repeatability. Containerization is evaluated where technically appropriate, while alternative cloud deployment models are retained for applications for which containerization provides limited value. Cloud-native services such as managed databases, storage, networking, monitoring, and identity systems can be incorporated into modernization scenarios.

Several migration scenarios are evaluated. The first scenario involves direct rehosting of a legacy application to cloud infrastructure. The second involves replatforming, such as moving a database to a managed cloud service or placing an application inside containers. The third involves partial refactoring, such as decomposing selected modules or introducing APIs. The fourth involves deeper modernization in which application functionality is reconstructed using contemporary cloud-native technologies. These scenarios allow the research to examine whether the AI framework can select an appropriate modernization strategy according to application characteristics rather than applying the same approach universally.

The baseline condition uses conventional expert-driven migration planning. Experienced practitioners evaluate application characteristics and manually select migration strategies. The experimental condition uses the proposed AI-supported framework. Where feasible, a hybrid condition is included in which AI produces recommendations that are reviewed by experts before execution. Comparing these conditions provides insight into the benefits of AI automation and the differences between fully manual, AI-assisted, and more autonomous migration processes.

Migration performance is evaluated using multiple quantitative indicators. Migration effort measures the amount of engineering work required to complete modernization. Migration duration measures the elapsed time between project initiation and successful deployment. Migration cost includes infrastructure, engineering, testing, and operational expenses. Application downtime measures service interruption during migration. Defect rate measures functional or technical defects identified after migration. Modernization coverage measures the proportion of identified modernization opportunities successfully addressed.

Application performance is measured before and after migration using response time, throughput, resource utilization, availability, and error rates. These measures determine whether modernization achieves its intended operational benefits. Cloud resource consumption is also evaluated to identify whether migration results in more efficient infrastructure utilization. Total cost of ownership can be estimated by considering infrastructure costs, licensing, maintenance, support, and operational effort.

AI decision quality is evaluated through recommendation accuracy and expert agreement. Each AI-generated migration recommendation is compared with an expert assessment and with the eventual outcome of experimental migration. A recommendation is considered successful when it achieves acceptable technical performance, business requirements, and migration objectives without unacceptable operational consequences. Disagreement between AI recommendations and expert decisions is further analyzed to identify whether the disagreement results from incomplete information, incorrect assumptions, model limitations, or differences in optimization priorities.

The quality of AI-generated code is evaluated through automated testing, static analysis, security scanning, code-quality assessment, and functional comparison with the legacy implementation. Regression tests verify whether business functionality is preserved after transformation. Security testing identifies vulnerabilities introduced during modernization. Generated infrastructure configurations are checked against predefined policies concerning access control, network exposure, encryption, resource limits, and compliance.

Migration-risk prediction is evaluated by comparing predicted risks with actual problems observed during controlled migration experiments. Precision and recall are used to assess the ability of the system to identify applications likely to encounter migration difficulties. False-negative cases receive particular attention because failure to identify a significant migration risk can have greater consequences than unnecessarily flagging a low-risk application.

The research also evaluates explainability. For every migration recommendation, the framework records the factors that contributed to the decision, including dependency complexity, technology age, business criticality, security requirements, estimated cost, scalability requirements, and predicted modernization benefits. Experts review whether these explanations are understandable and sufficiently detailed to support migration planning. Explainability is treated as an essential enterprise requirement rather than an optional feature.

Security, privacy, and governance are evaluated throughout the methodology. Enterprise source code and operational data may contain sensitive information, so the framework is designed to minimize unnecessary exposure of confidential data to AI models. Access controls restrict agents to authorized resources. Migration actions are logged to create an auditable history of recommendations, approvals, transformations, deployments, and validation results. Policies prevent AI systems from performing prohibited operations without appropriate authorization.

The collected experimental data are analyzed using descriptive and inferential statistical techniques. Measures such as mean, median, standard deviation, confidence intervals, percentage improvement, and effect size are calculated for migration duration, cost, defect rates, performance, and other indicators. Where repeated experiments provide sufficient observations, statistical significance tests are conducted to determine whether observed differences between baseline and AI-assisted approaches are statistically meaningful. Practical significance is also considered because an improvement in migration speed may not justify implementation complexity if reliability or security deteriorates.

Qualitative data are collected through expert reviews of AI recommendations, migration plans, generated code, explanations, and autonomous decisions. Experts assess accuracy, usefulness, trustworthiness, transparency, and operational risk. Their feedback is analyzed thematically to identify recurring limitations and opportunities for improvement. This provides a human-centered perspective on AI-supported modernization and helps determine which migration decisions should remain under expert control.

The methodology recognizes several threats to validity. The experimental application portfolio may not fully represent the diversity of real enterprise systems, and AI performance may vary depending on the selected models, prompts, datasets, and cloud platforms. Migration outcomes can also depend on organizational skills and infrastructure maturity. These limitations are addressed through diverse application scenarios, repeated experiments, explicit documentation of AI configurations, reproducible infrastructure, and multiple evaluation criteria. Future studies can validate the framework using larger industrial datasets and real enterprise migration projects.

The expected outcome is an empirically evaluated intelligent cloud migration framework capable of supporting application discovery, dependency analysis, migration-strategy selection, code modernization, testing, deployment, and post-migration optimization. The framework is intended to reduce reliance on manual analysis while preserving human authority over high-impact decisions. More broadly, the research positions AI not merely as a coding assistant but as an intelligent decision-support and orchestration capability across the complete enterprise modernization lifecycle. By combining machine learning, generative AI, knowledge representation, AI agents, cloud-native infrastructure, and governance mechanisms, the proposed approach aims to make legacy modernization more predictable, efficient, secure, and scalable. It ultimately provides a pathway through which organizations can transform aging application portfolios while minimizing business disruption and maximizing the long-term value of cloud adoption.

Results And Discussion

The proposed Intelligent Cloud Migration Strategies Using AI for Enterprise Legacy Application Modernization Frameworks demonstrate that artificial intelligence can significantly improve the planning, execution, monitoring, and optimization of enterprise legacy application migration. Legacy applications often represent critical business capabilities but are constrained by outdated programming languages, tightly coupled architectures, obsolete infrastructure dependencies, limited scalability, and high maintenance costs. Conventional migration approaches frequently rely on manual application assessment, expert-driven dependency identification, fixed migration plans, and lengthy testing cycles. These limitations increase migration risk and make it difficult for organizations to determine whether an application should be rehosted, replatformed, refactored, rebuilt, or retired. The proposed framework addresses these challenges by introducing AI-based application discovery, dependency analysis, migration-strategy selection, code analysis, transformation assistance, risk prediction, testing automation, and post-migration optimization. The results indicate that an AI-assisted migration framework can improve decision quality, reduce manual analysis effort, accelerate modernization activities, and provide greater visibility into migration risks. The first major result is the improvement in application discovery and assessment. Before migration, enterprise organizations must understand application architecture, source-code characteristics, infrastructure requirements, database dependencies, interfaces, external services, and operational behavior. In conventional environments, this information is often distributed across documentation, configuration files, monitoring systems, source repositories, and the knowledge of experienced employees.

The proposed AI framework consolidates these sources and applies intelligent analysis to construct an application inventory. Machine-learning and natural-language-processing techniques can analyze source code, configuration information, documentation, logs, and runtime telemetry to identify application components and relationships. This provides a more comprehensive understanding of legacy systems and reduces the likelihood that important dependencies will be overlooked during migration planning. Dependency discovery produced another significant improvement. Legacy applications frequently contain hidden dependencies that are difficult to identify through static documentation alone. An application may depend on a particular database version, operating-system feature, network configuration, file system, authentication service, or external enterprise application. AI-based dependency analysis can combine static code inspection with runtime observations to identify both explicit and implicit relationships. This enables migration teams to construct dependency graphs showing communication patterns among applications, databases, services, and infrastructure components. The resulting dependency map can be used to determine migration order and identify applications that should be migrated together. This is particularly important for large enterprises where migrating a single component without its dependent services can cause unexpected failures. Another important result is the intelligent selection of migration strategies. The traditional "one-size-fits-all" approach is inappropriate because different applications have different modernization requirements. The proposed framework evaluates application complexity, business criticality, technical debt, performance characteristics, infrastructure dependencies, expected modernization benefits, migration cost, and security requirements before recommending an appropriate strategy.

Applications with minimal architectural problems may be suitable for rehosting, whereas applications that require improved scalability may benefit from replatforming or refactoring. Highly obsolete applications with significant technical debt may require rebuilding, while applications with limited business value may be candidates for retirement. AI-based decision support enables migration teams to compare these alternatives systematically rather than depending entirely on subjective judgment. The framework also demonstrates the value of AI-assisted code modernization. Legacy systems often contain large volumes of code written in outdated languages or frameworks, making manual transformation expensive and time-consuming. AI-assisted code analysis can identify obsolete constructs, duplicated logic, security weaknesses, and modernization opportunities. Generative AI techniques can further support code translation, documentation generation, test-case creation, and transformation suggestions. However, the results indicate that AI-generated modernization should be treated as assisted transformation rather than completely autonomous rewriting. Generated code requires validation because semantic differences between legacy and modern programming environments may produce subtle functional errors. Human developers therefore remain important for reviewing business logic and validating critical transformations. Risk prediction is another major contribution of the proposed framework. Migration failures can result from technical complexity, incomplete dependency information, insufficient testing, incompatible infrastructure, security requirements, data-transfer issues, or inadequate organizational readiness. The AI framework can analyze historical migration data and current application characteristics to estimate migration risk. High-risk applications can then receive additional testing, staged deployment, expert review, or pilot migration before full production transition. This risk-based approach allows organizations to prioritize resources toward applications where migration failure would have the greatest consequences. The results also show that AI can improve migration sequencing. Instead of migrating applications according to organizational convenience alone, the framework can analyze dependency graphs and business priorities to identify an efficient migration order. Independent applications may be migrated early to generate quick benefits, while tightly coupled applications can be grouped into migration waves. Critical systems can be scheduled during controlled maintenance periods, while less critical applications can serve as pilot projects. Such sequencing reduces disruption and allows lessons from earlier migration waves to improve subsequent activities.

Automated testing further strengthens the framework. Functional equivalence is one of the most important requirements of legacy modernization because an application may appear technically successful after migration while producing different business outcomes. AI can generate test scenarios based on source code, historical transactions, application logs, and business rules. Automated regression testing can compare legacy and modernized application outputs to identify behavioral differences. Performance testing can also evaluate response times, throughput, and resource utilization before and after migration. These capabilities reduce the testing burden and provide stronger evidence that modernization has preserved essential business functionality. The framework also improves post-migration monitoring. Migration should not be considered complete when an application is successfully deployed to the cloud. Application behavior must be monitored continuously to identify performance degradation, unexpected resource consumption, security issues, and operational anomalies. AI-driven monitoring can compare post-migration behavior with historical baselines and detect deviations. When anomalies are identified, the system can recommend remediation actions such as resource scaling, configuration adjustment, workload redistribution, or rollback. This creates a continuous feedback loop between migration, operation, and optimization. From a cost perspective, the proposed approach provides opportunities for improved resource utilization and migration planning. AI can estimate infrastructure requirements, compare cloud deployment alternatives, and identify applications that may be unnecessarily overprovisioned after migration. Cost-aware recommendations can help organizations select appropriate compute, storage, networking, and database services. Nevertheless, cloud migration costs also include data transfer, licensing, modernization effort, testing, downtime, training, and operational transformation. Therefore, AI-based cost models should incorporate the complete migration lifecycle rather than focusing only on cloud infrastructure expenditure. The discussion further indicates that security and governance are essential considerations. AI-based migration systems may process sensitive source code, architectural information, credentials, configuration data, and business information. Strong access control, encryption, audit logging, data protection, and secure AI deployment are therefore necessary. Human oversight is particularly important when AI recommends major architectural changes or automatically transforms business-critical code. Overall, the results demonstrate that intelligent migration frameworks can transform cloud migration from a predominantly manual and reactive process into a data-driven, predictive, and continuously optimized activity. AI improves application discovery, dependency mapping, migration-strategy selection, risk assessment, code modernization, testing, sequencing, cost analysis, and post-migration monitoring. However, the findings also emphasize that AI should complement rather than completely replace migration specialists. The strongest implementation model combines automated analysis and recommendations with human validation at critical decision points. Such a balanced approach can reduce migration complexity while preserving accountability, security, and business continuity. Therefore, AI-enabled modernization provides a practical pathway for enterprises seeking to transform legacy applications into scalable, resilient, and cloud-ready systems.

Conclusion

Intelligent cloud migration using artificial intelligence provides a promising approach for addressing the complexity, cost, and risk associated with enterprise legacy application modernization. Legacy applications continue to support essential business operations in many organizations, but their aging architectures can restrict scalability, increase maintenance requirements, introduce security concerns, and prevent organizations from adopting modern digital capabilities efficiently. Traditional migration approaches often depend heavily on manual application assessment, expert knowledge, static documentation, and predefined migration plans. As the number and complexity of enterprise applications increase, these approaches become increasingly difficult to scale. The proposed intelligent migration framework addresses this challenge by integrating AI into the major stages of application discovery, assessment, dependency analysis, migration-strategy selection, modernization, testing, deployment, monitoring, and optimization. The results demonstrate that AI can provide substantial value by transforming fragmented migration information into actionable intelligence. One of the primary conclusions is that intelligent application discovery significantly improves migration preparedness. AI-based analysis can combine source-code information, configuration data, application documentation, operational logs, infrastructure telemetry, and runtime behavior to create a more accurate representation of legacy applications. This comprehensive view is important because incomplete knowledge of application dependencies is one of the major causes of migration failures. By identifying relationships among applications, databases, services, infrastructure components, and external interfaces, AI can support more reliable migration planning.

Dependency analysis also enables organizations to determine which systems should be migrated independently and which should be grouped into coordinated migration waves. Another important conclusion is that AI can improve the selection of modernization strategies. Enterprise applications cannot all be migrated using the same technique. Some applications can be moved directly to cloud infrastructure with minimal modification, while others require architectural redesign, containerization, database modernization, code transformation, or complete redevelopment. The proposed framework evaluates application characteristics and business requirements to recommend strategies such as rehosting, replatforming, refactoring, rebuilding, or retirement. This provides organizations with a structured decision-making process and reduces dependence on subjective assessments. AI-assisted strategy selection is particularly useful when enterprises manage thousands of applications and lack sufficient resources to perform detailed manual analysis of every system. The study also concludes that AI can accelerate legacy code modernization. Automated code analysis can identify obsolete components, technical debt, security vulnerabilities, duplicated functionality, and opportunities for transformation. Generative AI can support code conversion, documentation, test generation, and modernization recommendations. Nevertheless, complete autonomous code transformation is not appropriate for every enterprise application. Business-critical applications may contain complex rules that are poorly documented and difficult for AI systems to interpret correctly. Human developers and domain specialists must therefore validate AI-generated changes before production deployment.

This establishes an important principle for intelligent modernization: automation should increase human productivity without eliminating expert responsibility. Risk prediction represents another major benefit. AI can analyze technical complexity, application criticality, dependency density, historical migration outcomes, infrastructure requirements, and other characteristics to identify high-risk migration candidates. Organizations can use these predictions to prioritize testing, allocate specialist resources, select pilot applications, and design controlled migration waves. This risk-based approach can reduce unexpected failures and improve overall migration governance. Furthermore, AI-supported testing improves confidence in functional equivalence between legacy and modernized systems. Automated regression tests, performance tests, security validation, and output comparisons can detect problems before applications are fully transitioned to production. This is particularly important because technical deployment success does not necessarily guarantee business correctness. A migrated application must continue to produce expected business results while also meeting performance, availability, and security requirements. The framework also demonstrates that migration is not a one-time technical event but a continuous optimization process. After deployment, cloud applications must be monitored and adjusted according to actual workload behavior. AI-driven monitoring can identify anomalies, predict performance problems, recommend scaling actions, and evaluate whether the selected cloud architecture continues to meet organizational objectives. This creates a lifecycle in which migration decisions can be continuously evaluated and improved. Cost optimization is another important conclusion. Cloud migration does not automatically guarantee cost reduction. Poor resource selection, excessive provisioning, inefficient architecture, and uncontrolled consumption can produce unexpectedly high cloud expenditure. AI-based cost analysis can help organizations estimate resource requirements, compare deployment alternatives, identify idle resources, and optimize workloads after migration. However, comprehensive economic evaluation should include migration effort, application transformation, data movement, licensing, training, operational changes, and long-term maintenance rather than considering infrastructure cost alone. Security, governance, and responsible AI are equally important. Intelligent migration frameworks may access sensitive application code, infrastructure configurations, business data, and operational information. Consequently, organizations must implement strict identity management, access controls, encryption, auditing, secure model deployment, and data-protection mechanisms. AI recommendations should be traceable, and critical decisions should remain subject to human approval.

This is especially important for applications supporting financial transactions, healthcare operations, public services, or other business-critical processes. In conclusion, the proposed intelligent cloud migration framework demonstrates that AI can significantly improve the effectiveness of enterprise legacy modernization by providing automated discovery, dependency intelligence, strategy recommendations, risk prediction, code transformation assistance, testing automation, and continuous post-migration optimization. The framework does not imply that human migration specialists are no longer necessary. Instead, it establishes a collaborative model in which AI performs computationally intensive analysis and repetitive activities while human experts provide architectural judgment, business understanding, governance, and accountability. This combination offers a practical and scalable approach to modernization. As enterprises continue moving toward cloud-native architectures, AI-enabled migration will become increasingly important for reducing technical debt, improving operational agility, and accelerating digital transformation. The long-term value of the approach lies in creating a migration lifecycle that is intelligent, adaptive, measurable, and continuously improving. When supported by strong governance and human oversight, AI can help enterprises transform complex legacy portfolios into secure, scalable, resilient, and cost-effective cloud environments while minimizing operational disruption and maximizing the strategic benefits of modernization.

Future Work

Future work should focus on expanding the proposed intelligent cloud migration framework into a more autonomous, explainable, secure, and enterprise-scale modernization platform. One important research direction is the development of multi-agent migration systems, in which specialized AI agents independently manage application discovery, dependency mapping, code analysis, security assessment, cost estimation, testing, deployment, and monitoring while coordinating through a central policy engine. Such systems could dynamically adapt migration plans when new dependencies or unexpected technical problems are discovered. Future research should also investigate advanced generative AI and large language models for automated legacy-code transformation, documentation recovery, business-rule extraction, and generation of cloud-native application components. However, these capabilities should be combined with formal verification and automated testing to ensure that AI-generated modernization preserves business functionality.

Another major direction is predictive migration-risk modeling using larger datasets collected from real enterprise migration projects. These models could estimate failure probability, expected downtime, modernization cost, and resource requirements before migration begins. Future frameworks should also support hybrid-cloud and multi-cloud modernization, allowing intelligent systems to select deployment locations according to cost, latency, regulatory requirements, data sovereignty, security, and application performance. Digital twins could be introduced to simulate migration scenarios before production execution, enabling organizations to evaluate different modernization strategies without risking live systems.

Security research should address threats against AI-driven migration tools, including malicious code transformation, manipulated dependency information, model attacks, unauthorized agent actions, and sensitive-data exposure. Future systems should therefore incorporate zero-trust principles, secure model execution, least-privilege access, and comprehensive audit mechanisms. Additional research should examine sustainable migration by considering energy consumption and carbon emissions when selecting cloud architectures and deployment locations. Finally, standardized benchmarks should be developed to evaluate intelligent migration frameworks using metrics such as migration duration, application availability, modernization accuracy, dependency-detection precision, cost reduction, security findings, testing coverage, and human intervention requirements. These developments could lead to trustworthy autonomous modernization platforms capable of continuously assessing legacy portfolios, recommending optimal transformation strategies, executing controlled migration activities, and learning from operational outcomes while maintaining appropriate human governance.

References

[1] Redouane, K., Veernapu, K., Konduru, S. C., Surendranath, N., Fikri, Y., & Vaz, S. (2026, March). A Generative AI-Driven Cloud Framework for Predictive Optimization in Smart Manufacturing Ecosystems. In 2026 Innovations in Machine, Engineering, and Digital Conference (IMED) (pp. 1-6). IEEE.

[2] Mathew, A. (2023). Sentinel AI: An Investigation into Robust Threat Mitigation Strategies for Artificial Intelligence. Educational Research (IJMCER), 5(5), 108-111.

[3] Indurthy, V. S. K. (2026). A modern approach to asset management: Integrating SimCorp Dimension with cloud data platforms. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(2), 4757–4762.

[4] Gopinathan, V. R. (2025). Revolutionizing Revenue Cycle Management in the US Healthcare System Using AI-Powered Cloud Solutions. International Journal of Computer Technology and Electronics Communication, 8(4), 11106-11118.

[5] Chaba, A. (2025). Human-AI collaboration models in presales: Designing the augmented pre-sales engineer. International Journal of Research and Applied Innovations (IJRAI), 8(4), 12736–12742.

[6] Sudhan, S. K. H. H., & Kumar, S. S. (2016). Gallant Use of Cloud by a Novel Framework of Encrypted Biometric Authentication and Multi Level Data Protection. Indian Journal of Science and Technology, 9, 44.

[7] Bheemisetty, N. (2026). Enterprise file management system (FMS): A policy-driven, federated architecture for unified file lifecycle governance. International Journal of Research and Applied Innovations (IJRAI), 9(2), 118–122.

[8] Raja, G. V. (2023). Modernizing enterprise systems using AI with machine learning and cloud computing for intelligent systems. International Journal of Future Innovative Science and Technology (IJFIST), 6(6), 11713.

[9] Prasad, A. (2026, May). Accelerating HPC Performance: Strategies for Overcoming Modern Challenges with NVMe, InfiniBand, and RDMA. In 2026 International Conference on Signal, Systems, and Computing for Next-Gen Automation (ICSSCNA) (pp. 463-471). IEEE.

[10] Manda, P. (2023). Leveraging AI to Improve Performance Tuning in Post-Migration Oracle Cloud Environments. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 6(3), 8714-8725.

[11] Rajula, A. (2024). Staleness-bounded aggregation for cross-institutional federated clinical models. International Journal of Research Publications in Engineering, Technology and Management, 7(1), 10030–10041.

[12] Kilari, K. K. (2026). Cybersecurity awareness and behavioral outcomes in healthcare organizations. International Journal of Drug Delivery Technology, 16(58s), 816–823.

[13] Ambati, K. C. (2026). AI-powered procurement architecture: Streamlined UX, scalable ML pipelines, and enterprise efficiency. International Journal of Research and Applied Innovations (IJRAI), 9(1), 13681–13685.

[14] Rohit Wadhwa. (2024). Designing Event-Driven Enterprise Systems with Distributed Data Sharding and Partitioning Strategies. ISCSITR- International Journal of Computer Applications (ISCSITR-IJCA), 5(1), 22–35.

[15] Gummadi, V. P. K. (2025). MuleSoft's Role in Advancing Sustainable Digital Infrastructure: An Enterprise Integration Perspective. Journal of Information Systems Engineering and Management, 10, 1313-1321.

[16] Sugumar, R. (2023, September). A Novel Approach to Diabetes Risk Assessment Using Advanced Deep Neural Networks and LSTM Networks. In 2023 International Conference on Network, Multimedia and Information Technology (NMITCON) (pp. 1-7). IEEE.

[17] Karnam, V. S. (2024). Adaptive and federated test automation using AI in distributed multi-cloud and hybrid infrastructure environments. International Journal of Future Innovative Science and Technology (IJFIST), 7(4), 111–121.

[18] Biswas, B., Faysal, S. M., Das, S., Hanif, A., Akter, T., Rashed, R. A. M., & Shamarukh, S. (2026). A Framework for Agentic AI-Driven Financial Fraud Detection, Investigation, and Risk Mitigation in Real-Time Payment Ecosystems. International Journal of Computer Information Systems and Industrial Management Applications, 18(3s), 1029-1040.

[19] Cyril, H., & Poranki, U. (2026). Next-generation telecom provisioning: AI-driven, cloud-native architectures for autonomous networks. Notion Press

[20] Singh, I. K. (2024). Enterprise knowledge graphs for biomedical data integration: A scalable architecture for semantic interoperability. International Journal of Engineering & Extended Technologies Research (IJEETR), 6(4), 8165-8176.

[21] Juvvadi, R. R. (2025). Toward autonomous finance: A multi-agent system architecture for the self-driving financial close. International Journal of Engineering & Extended Technologies Research (IJEETR), 7(6), 11290–11295.

[22] Anand, L. (2023). Machine Learning Enabled Enterprise Integration through Intelligent API Governance Secure Cloud Infrastructure and Automated Operations. International Journal of Research and Applied Innovations, 6(3), 5972-5979.

[23] Pareek, C. (2023). Unmasking bias: a framework for testing and mitigating AI bias in insurance underwriting models. J Artif Intell Mach Learn & Data Sci, 1(1), 1736-41.

[24] Khanna, K., Bellundagi, M., & Kumar, S. (2026). AI-Driven Transformation of Healthcare: From Predictive Analytics to Personalized Medicine. Artificial Intelligence and the Future of Disciplines, 1(1).

[25] Suddala, V. R. A. K. (2026). Innovation and compliance at global scale: Predictive analytics meets DevOps automation. International Journal of Engineering & Extended Technologies Research (IJEETR), 8(1), 239–245.

[26] Tyagi, N. (2024). Deep reinforcement learning for algorithmic trading strategies. International Journal of Research and Applied Innovations, 7(2), 10415-10422.

[27] Benchellal andLotfi Ben Othmane, F., Konapalli, Y., Tunc, C., & Bhargava, B. (2026). Control-Aware Manipulation of ArduPilot via Legitimate MAVLink Commands: Simulation and Hardware Validation. arXiv e-prints, arXiv-2606.

[28] Challa, R. (2025). Benchmark-driven GPU performance optimization for medical imaging, genomics, and large-scale AI workloads. Journal of Information Systems Engineering and Management, 10(1), 225–234.

[29] Gopinathan, V. R. (2025). Revolutionizing Revenue Cycle Management in the US Healthcare System Using AI-Powered Cloud Solutions. International Journal of Computer Technology and Electronics Communication, 8(4), 11106-11118.

[30] Vas, M. R. (2026). Accelerating Intelligent Enterprise Engineering through Generative AI Secure Cloud Infrastructure and Industrial Systems. International Journal of Science, Research

and Technology, 9(3), 855-862.

[31] Narapareddy, V. S. R., & Yerramilli, S. K. (2023). Artificial intelligence incident forecasting. International Journal of Engineering Technology Research & Management, 7(12), 551–559.

[32] Manda, P. (2023). Leveraging AI to Improve Performance Tuning in Post-Migration Oracle Cloud Environments. International Journal of Research Publications in Engineering, Technology and Management (IJRPETM), 6(3), 8714-8725.

[33] Raja, G. V. (2023). AI Driven Secure Intelligent Framework for Fraud Detection Cybersecurity and Cloud Based Enterprise Systems. International Journal of Advanced Research in Computer Science & Technology (IJARCST), 6(5), 9068-9076.

[34] Sivakumer, D. (2024). The role of work models in influencing organizational performance and employee wellbeing: A comparative study on full-time, remote, and hybrid paradigms. International Journal of Advanced Engineering Science and Information Technology, 7(4), 14496–14510.

[35] Gowda, M. K. S. (2026). Aligning governance, data, and controls: A practical blueprint for bank remediation initiatives. International Journal of Research Publications in Engineering, Technology and Management, 9(2), 789–794.