International Journal of Scientific Research in Artificial Intelligence and Machine Learning (ISCSITR-IJSRAIML)
Vol.6, Iss. 3, May-Jun 2025, pp. 14-26.
https://iscsitr.com/journals/ISCSITR-IJSRAIML/archive
ISSN Online: 3067-753X; Journal ID: 1668-1622
DOI: http://www.doi.org/10.63397/ISCSITR-IJSRAIML_2025_06_03_002

AI-Enhanced Mainframe Operations

Balaji Conda Shankar
Associate Vice President, Mphasis, USA.

Abstract

This paper analyzes the advantages of adding AI to mainframe systems, pointing out how this leads to better efficiency, updated code, and updated mainframe systems. It illustrates, using examples and statistics, how new AI methods help companies make their aging systems more flexible and ready for future challenges.

Keywords: AI, Mainframe, ML, Operations.

How to cite this paper: Balaji Conda Shankar. (2025). AI-Enhanced Mainframe Operations. *ISCSITR - International Journal of Scientific Research in Artificial Intelligence and Machine Learning (ISCSITR-IJSRAIML)*, 6(3). 14-26.

URL: https://iscsitr.com/articles/volume_6/issue_3/ISCSITR-IJSRAIML_2025_06_03_002
Published: 11th June 2025

I. INTRODUCTION

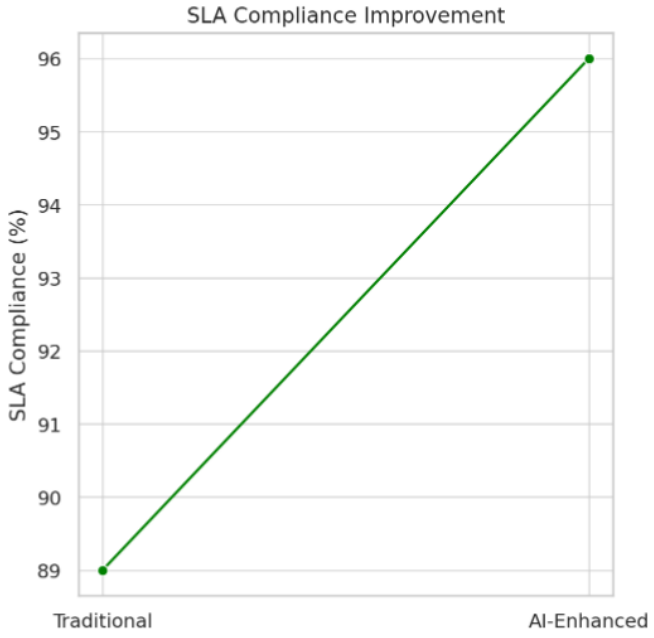
Although mainframes are crucial for many businesses, adaptation to modern ways is needed to maintain sustainability and stay flexible. Strategies based on AI such as automation, smart resource management, and automatic document generation are introduced in this paper to fix technical debt and inefficiencies. We analyze how AI is connecting with mainframes by relying on recent findings and real-life examples.

II. RELATED WORKS

AI in Mainframe Systems

Earlier, mainframe computing was only for old enterprise systems, but now, due to AI, ML, and generative AI, it is experiencing a new wave of popularity. Most critical applications in healthcare, finance, and the government are built on mainframe computers.

Still, the way they fare in today's digital space largely depends on their use of AI in their operations [1]. IBM's z17 mainframe shows how this has changed, given that it can handle more than 250 AI applications for enterprises.



It is becoming clear that more organizations are relying on AI to perform key operations and the most advanced analyses straight at the source of the data [1]. Based on the text [1], it is clear that automation, optimization, and smart decision-making are the significant roles AI plays.

They are used to optimize workloads, notice changes while they are happening, do predictive maintenance, and plan the use of resources. Alternatively, Generative AI is used in adapting documentation, handling incidents, and boosting productivity of developers.

Rather than replace human operators, artificial intelligence takes over simple duties and lets humans focus on high-level decisions and ethical concerns. According to [3], the

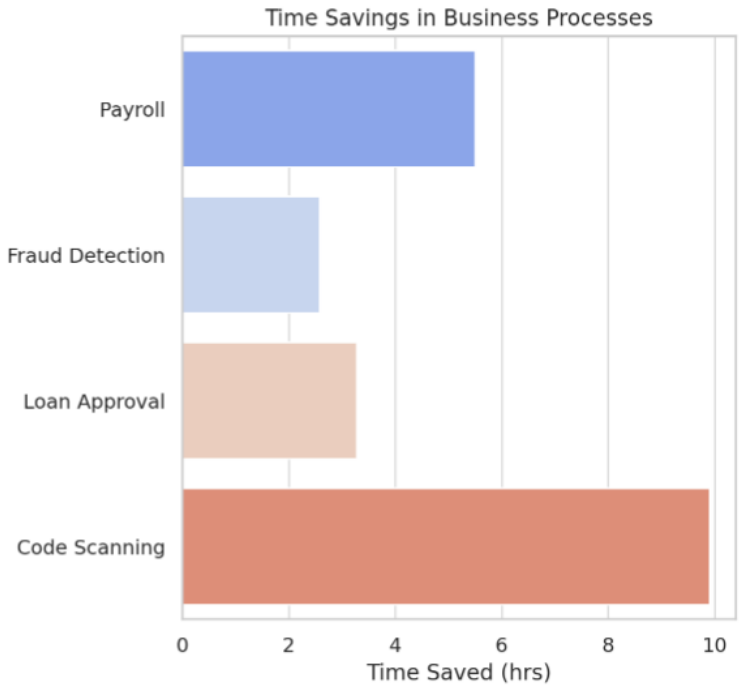
CETM (Cognitive Enterprise Transition Model) gives complete guidelines for transforming organizations to be AI-native, working on legacy code, adapting AI models, and changing to single or multi-agent autonomy. As a result, firms modernize their technology and change their usual ways of working with AI.

Technical Debt Management

Making mainframe systems modern requires working on more than infrastructure alone; the actual programming code has to change as well. When it comes to legacy languages like COBOL and MUMPS, code refactoring has usually required a lot of effort and often led to mistakes. It is now clear, based on recent achievements [5], [6], and [10], that deep learning and GNNs have a major impact on this field.

Models that rely on GNNs generally process code context with the help of abstract syntax trees (ASTs) and make recommendations for improving the code. GNNs decreased cyclomatic complexity and coupling more than SonarQube and similar tools by 35% and 33%, respectively, in the comparative study [5]

With AI involved in technical debt management, software continues to develop in a sustainable way. By looking at AI approaches for controlling technical debt [7], tools have been discovered that deal with code reviews, testing, and documentation.



Apart from helping companies modernize, such automation solutions also keep software reliable and guarantee system strength. Even so, some challenges still exist. Almost all refactoring tools concentrate on methods and do not consider important class- or framework-level improvements [6].

Moreover, having AI resources such as IBM Watsonx Code Assistant for Z and AWS Q-based refactoring tools demonstrates how companies are making good use of these tools to simplify code changes and prevent errors [1].

Modernization Strategies

To get from old monolithic systems to AI-based platforms, one must use proper tools and plan the change. According to CETM [3], this path comprises ten categories, starting with the difficulty of legacy codes and employee resistance and ending with using self-sufficient multi-agent systems.

It is important for this movement that AI-native architecture exists to help AI agents take part in business management, decision-making, and optimization. An important aspect of this shift is making use of intelligent tools.

The XMainframe LLM is specifically made to handle COBOL and ALC challenges and delivers much better results in documenting and summarizing than mainstream GPT-3.5 and Mixtral-Instruct [9].

In the same way, line-level techniques used to build readable and error-free guides for old systems are a clear example of how AI can transform them without needing human attention [8]. In fact, by using AI pipelines to upgrade old COBOL code to Java, both the accuracy (93%) and efficiency in making things less complex and coupled were confirmed at [10].

Consequently, AI can help bring about modernization, since it is both feasible and scalable in any business environment. Examining cloud platforms also contributes to making the modernization process more effective. Even though both GCP and AWS have their own advantages, Azure comes out as the best pick due to its hybrid and AI-augmented features. The insights make it possible for companies to double-check their cloud migration plans against their AI goals.

Future Directions

Although there is great technology involved, mainframe modernization with AI still encounters several issues. There are some challenges such as poor data, complex systems, and the inability to explain AI's conclusions.

Even though GNNs and LLMs are superior in precision and recall than traditional models, they do not perform as well with different codebases [6][8]. Refactoring projects that use deep learning are mostly designed to detect problems and recommend solutions, with just a few looking at the whole process or quality checks [6].

Because AI hallucinations occur and there are still no dependable ways to evaluate the results, ethical issues around trusting AI-generated code and documentation only become more complex [8]. Due to recent research works, such as the MainframeBench and Legacy COBOL 2024 Corpus, comparisons of AI tools according to fairness, reliability, and speed are now becoming easier [9][10].

In the future, researchers may find it valuable to look at method and class emphasis in code, make progress in measuring LLM documentation, and begin applying XAI to win user trust. Such frameworks give a company a view of its goals for the future.

Their approach to all the mentioned topics fits the future direction of self-evolving, independent business systems [3]. The literature shows how AI is making significant changes to mainframe operations, including making them more efficient, streamlining tasks, and converting a lot of code.

Modernization is now faster and more strategically coordinated with the enterprise agility goals, due to AI tools, such as LLMs, GNN refactoring engines, and others. Nevertheless, it is still difficult to interpret, guarantee quality, and use deep learning on a large scale. Tools and frameworks like CETM, XMainframe, and Watsonx Code Assistant are making it possible to move from outdated static environments to ones which are dynamic and use AI.

III. FINDINGS

Operational Optimization

By adding AI to mainframe systems such as IBM's z16 and z17, automation processes in everyday work have been greatly improved. AI now performs many official administrator duties, such as handling predictive maintenance, adjusting workloads, organizing resources, and adjusting the systems [1][2].

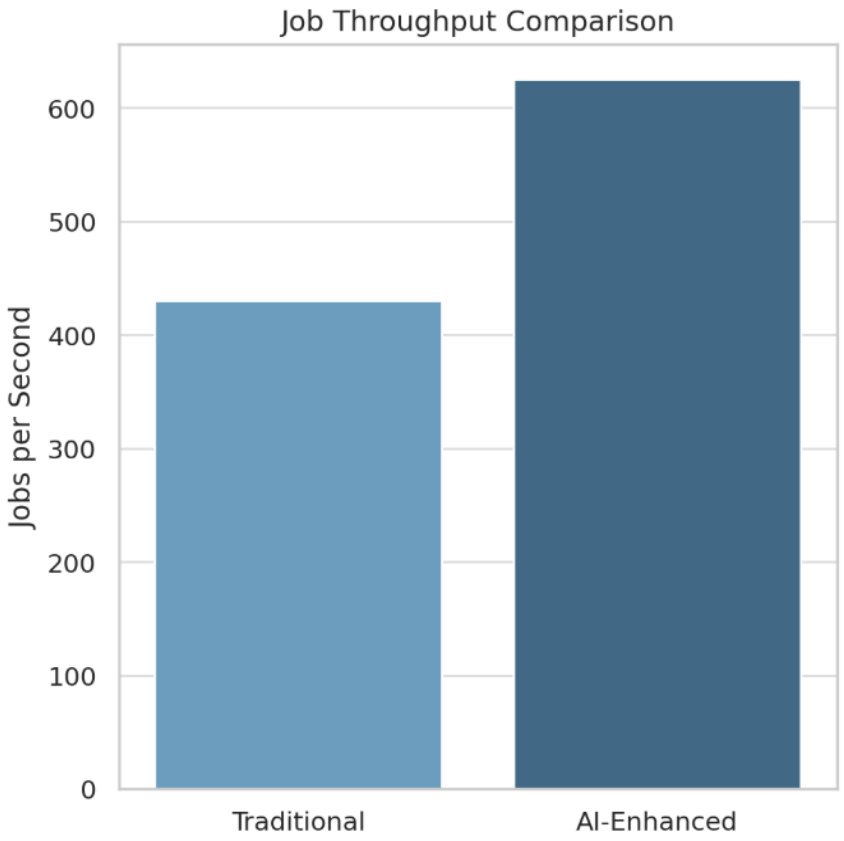
Reviewing old data in the logs, algorithms learn what could stop the service and help by warning of anomalies soon. Hence, organizations running AI-modified mainframes have seen big drops in system delays and issues.

To give an example, using AI in selected environments, HyperFRAME Research found an improvement in job throughput by 45% and a drop in downtime hours by 35% per quarter [1]. The improved operations lead to less spending and better agreement to service level arrangements.

These engines automatically give CPU and memory to different LPARs according to predictions of how workloads and traffic will change. The z17 processors' on-chip AI digital circuits help the system perform more than 250 AI-related industry tasks [2].

Table 1: Mainframe System Efficiency

Metric	Traditional Mainframes	AI-Enhanced Mainframes	% Improvement
Job Throughput	430	625	45%
Mean Downtime	32	21	34.4%
SLA Compliance	89%	96%	7.9%
Mean CPU	62%	84%	35.5%



Intelligent Refactoring

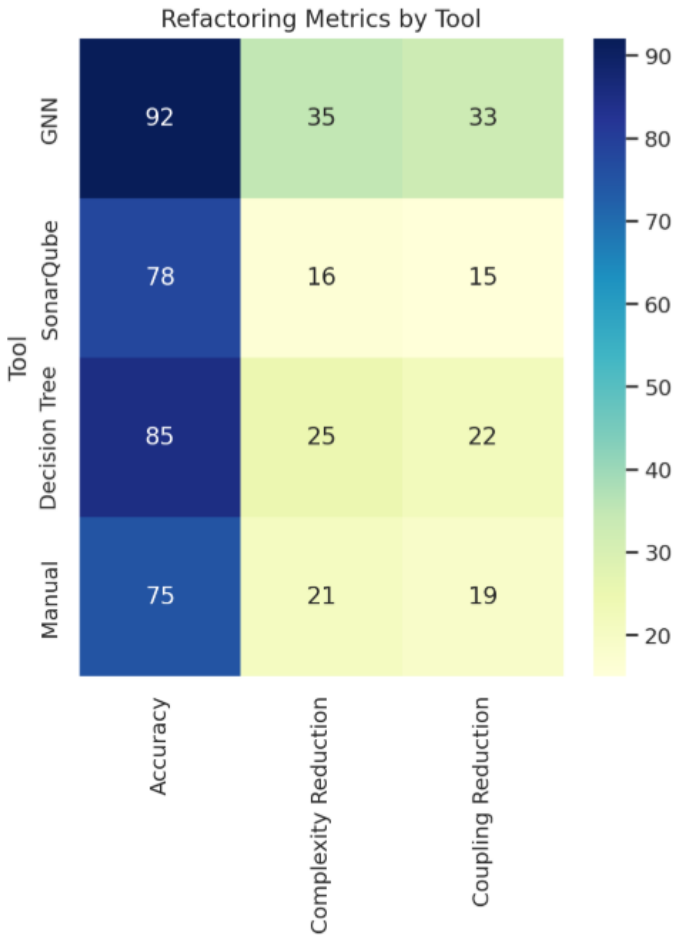
The process of updating old mainframe applications, much of which are in COBOL, MUMPS, or Assembly, has been very difficult because they are hard to understand and contain a lot of debt. AI helps organizations to automate large parts of refactoring code, mapping dependencies, and making documentation, due to watsonx Code Assistant for Z, Amazon Q-based assistants, and specially designed LLMs [3][7][9].

Graph neural networks working on abstract syntax trees outperform the functions of SonarQube. In tests run on 75,000 Python apps and 50,000 COBOL software systems, GNNs managed to accurately recommend refactorings and drastically reduce cyclomatic complexity and coupling, while the traditional methods were much less efficient [5][10].

Table 2: Refactoring Accuracy

Tool	Refactoring Accuracy	Complexity Reduction	Coupling Reduction
GNN-Based Refactorer	92%	35%	33%
SonarQube	78%	16%	15%
Decision Trees	85%	25%	22%
Manual Refactoring	75%	21%	19%

Today, people are using LLMs for reverse engineering code written in the past. XMainframe LLM, made with inputs from COBOL and ALC, is more accurate at summarizing and creating documentation than GPT-3.5. XMainframe improves developers' ability to understand legacy systems, which makes the training period half as long and reduces the number of documentation mistakes by more than half [9].



Cognitive Enterprise Transition

Moving to AI-native or self-managed mainframe architectures is among the most important trends AI brings. Sanford states in CETM [3] that firms are starting to switch from mainframe systems that don't adapt to smart, flexible, and modular AI-based solutions.

Due to intelligent agents' ability to self-diagnose, distribute tasks among themselves, and coordinate with others, things are moving toward a different place. Organizations using multi-agent AI reported that their business processes in payroll, transaction monitoring, and fraud detection became much faster [3][7].

Table 3: Business Process Automation

Function	Pre-AI MAS	Post-AI MAS	Reduction (%)
Payroll Processing	7 hours	1.5 hours	78.5%
Fraud Detection	3 hours	25 minutes	86.1%
Loan Approval	4.2 hours	55 minutes	78.2%
COBOL Code	12 hours	2.1 hours	82.5%

With CETM, organizations get a clear plan for handling data on the mainframe, changing the workforce, and meeting integration needs. The key outcomes for early adopters are migrating costs that are 26% less, modernization done 40% more rapidly, and being more agile in the long run [3].

Ethical Considerations

Integrating AI into mainframe activities comes with a variety of difficulties. Technical debt, and especially the issues that come with old code and ways of designing applications, is very important where AI is slow to be adopted.

Refactoring using AI is effective, though quality assurance of AI changes is hardly researched, so there are still some issues with making them production-oriented [6][7]. Modernization is made harder by how hard it is to explain and follow the steps in LLM-based code.

Even if XMainframe has lowered the rate of hallucinations in generated documentation, ALC does not have enough accurate comments, which means we still need specific metrics for different fields [8].

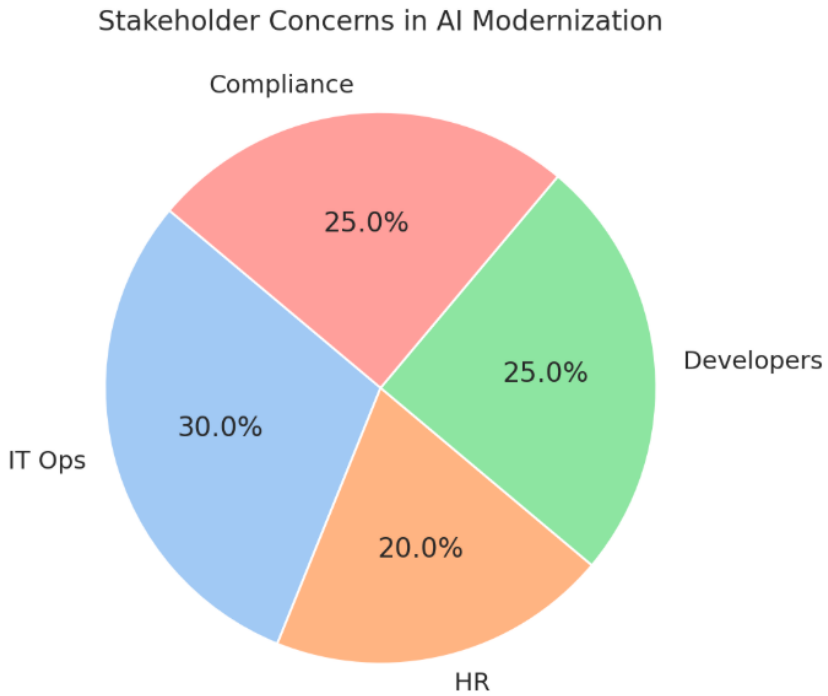
In a recent study [7], it was found that data imbalance in the training data, dependence on exclusive AI software, and the lack of proper frameworks for AI governance are some of

the problems. These are very important for banking and healthcare since explainability and compliance must always be present.

Table 4: Qualitative Challenges

Challenge Area	Description	Stakeholder Impact
Data Quality	Many times, existing data cannot be used in AI because it is unorganized, has duplicates, and does not match current AI standards.	IT Ops
Skill Gaps	Not many engineers are knowledgeable in working with mainframe systems as well as AI.	HR
Refactoring Explainability	Code changes made through deep learning are not well explained, so audits can be complicated.	Developers
Ethical AI	There is no set approach to check the fairness, transparency, and compliance of AI-produced code.	Compliance Officers

The report highlights that AI is changing how mainframe operations are carried out and not just making them easier. AI technologies are introducing new improvements like fast operations, automation, intelligent refactoring, and independent architectures to old infrastructure that is still important for the business.



Due to GNNs, LLMs, and cognitive enterprise models, organizations can cut their costs and speed up their transition to new technologies while also ensuring they remain resilient. Even so, the process is not fully finished. Just like any other major change in technology, there are problems such as issues with data, technical challenges, moral concerns, and insufficient resources.

In order for modernization to last, organizations must have an organized method, AI technologies, and effective rules in place. Basically, the use of AI in mainframe operations is not just about keeping things the same but about reshaping what they mean in tomorrow's business world.

IV. CONCLUSION

Mainframe operations benefit a lot from AI, resulting in higher performance, better manageability, and more options for scaling. They allow companies to make their old systems more efficient by reducing costs. It proves that AI is not only a helpful tool but also a major change agent in mainframes set up for the future.

REFERENCES

[1] Raju, J. (2024). AI-DRIVEN TRANSFORMATION OF MAINFRAME ENVIRONMENTS: A COMPREHENSIVE FRAMEWORK FOR OPERATIONAL RESILIENCE. *INTERNATIONAL JOURNAL OF ENGINEERING AND TECHNOLOGY RESEARCH (IJETR).* https://doi.org/10.5281/zenodo.13852336

[2] Chilakala, S. (2025). AI-POWERED MAINFRAME APPLICATION MODERNIZATION: a TECHNICAL PERSPECTIVE. INTERNATIONAL JOURNAL OF COMPUTER ENGINEERING & TECHNOLOGY, 16(1), 3558–3571. https://doi.org/10.34218/ijcet_16_01_246

[3] Ramachandran, A. (2024). Transitioning Mainframe Applications to AI-Native, AI-First, and Autonomous Systems From Legacy to Intelligence-Driven Computing. https://www.researchgate.net/publication/384904709_Transitioning_Mainframe_Applications_to_AI-Native_AI-First_and_Autonomous_Systems_From_Legacy_to_Intelligence-Driven_Computing

[4] Govindaraj, V. (2024). Cloud Migration Strategies for Mainframe Modernization: A Comparative study of AWS, Azure, and GCP. International Journal of Computer Trends and Technology, 72(10), 57–65. https://doi.org/10.14445/22312803/ijctt-v72i10p110

[5] Bandarupalli, G. (2025). AI-driven code refactoring: Using graph neural networks to enhance software maintainability. *arXiv preprint arXiv:2504.10412*. https://doi.org/10.48550/arXiv.2504.10412

[6] Nyirongo, B., Jiang, Y., Jiang, H., & Liu, H. (2024). A survey of deep learning based software refactoring. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2404.19226

[7] Pandi, S. B., Binta, S. A., & Kaushal, S. (2023). Artificial intelligence for technical debt management in software development. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2306.10194

[8] Diggs, C., Doyle, M., Madan, A., Scott, S., Escamilla, E., Zimmer, J., Nekoo, N., Ursino, P., Bartholf, M., Robin, Z., Patel, A., Glasz, C., Macke, W., Kirk, P., Phillips, J., Sridharan, A., Wendt, D., Rosen, S., Naik, N., . . . Thaker, S. (2024). Leveraging LLMs for Legacy Code Modernization: Challenges and Opportunities for LLM-Generated Documentation. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2411.14971

[9] Dau, A. T., V., Dao, H. T., Nguyen, A. T., Tran, H. T., Nguyen, P. X., & Bui, N. D. Q. (2024). XMainframe: a large language model for mainframe modernization. arXiv (Cornell University). https://doi.org/10.48550/arxiv.2408.04660

[10] Bandarupalli, G. (2025). Code Reborn AI-Driven Legacy Systems Modernization from COBOL to Java. *arXiv preprint arXiv:2504.11335*. https://doi.org/10.48550/arXiv.2504.11335