Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140

Volume 15 Issue 01, 2026

AI-Assisted Software Architecture Design for Multi-Cloud Enterprise Environments

Pradeep Kumar Mulluri
USA
Email: pradeepmulluri4567@gmail.com

Peer Review Information

Submission: 08 Feb 2026

Revision: 26 Feb 2026

Acceptance: 18 March 2026

Keywords

AI-assisted architecture, Multi-cloud computing, Enterprise software systems, Intelligent decision-making, Cloud optimization.

Abstract

The rapid adoption of multi-cloud strategies by enterprises has introduced significant challenges in software architecture design, including increased system complexity, heterogeneous cloud services, interoperability issues, security risks, and performance optimization across distributed environments. Traditional architecture design approaches are often manual, time-consuming, and insufficient to dynamically adapt to evolving business and technological requirements. To address these challenges, this study proposes an AI-assisted software architecture design framework tailored for multi-cloud enterprise environments. The proposed approach leverages artificial intelligence techniques such as machine learning, knowledge-based systems, and architectural pattern recognition to support automated decision-making in cloud service selection, workload distribution, scalability planning, fault tolerance, and security enforcement. By analyzing historical system data, architectural constraints, and real-time operational metrics, the framework provides intelligent recommendations for optimal architectural configurations while ensuring compliance, resilience, and cost efficiency. The AI-assisted model enables architects to evaluate multiple design alternatives, predict performance bottlenecks, and proactively mitigate risks before deployment. Experimental evaluation and scenario-based analysis demonstrate that the proposed framework significantly improves design accuracy, reduces architectural complexity, and enhances system performance compared to conventional design methods. This research highlights the potential of AI-driven architectural intelligence to support adaptive, scalable, and robust software systems in complex multi-cloud enterprise ecosystems.

Introduction

Rapid emergence of enterprise multi-cloud strategies has brought challenges in software architecture design, such as rising complexity of system, heterogeneous cloud services and interoperability among them, security concerns, performance optimization on distributed environs. Conventional design of architecture is manual, time-consuming and is an inadequate solution for keeping pace with ever changing business needs or technology challenges. In response to these challenges, this research offers an AI-supported software architecture design framework that fits the requirements of multi-cloud enterprise environments. To enable automated decision support in the areas of cloud service selection, workload distribution, scalability estimation, fault tolerance and security stance enforcement, suggest to use artificial intelligence techniques of machine learning, knowledge-based systems and pattern recognition from architecture for modeling complex system behavior. Leveraging historical system data, architectural constraints and current operational metrics, the framework offers intelligent recommendations on effective architectural configuration with compliance, resilience and cost in mind. The AI-guided blueprint allows architects to explore multiple design options and anticipate performance bottlenecks, while preemptively address risk prior to implementation. Experimental results and scenario-oriented comparison show that the proposed framework can increase design precision, decrease architectural complexity and improve system performances versus conventional-design methods. This study has demonstrated the capability for AI-based architectural intelligence to advance adaptive, scalable and resilient software systems within complex multi-cloud enterprise ecosystems.

Designing software architectures for multi-cloud enterprise environments is a highly challenging task due to the heterogeneity of cloud platforms, diverse service models, varying pricing structures, and inconsistent security and compliance mechanisms. Architects must carefully decide how to distribute workloads, manage inter-cloud communication, ensure data consistency, and maintain performance and reliability under dynamic conditions. Traditional software architecture design approaches rely heavily on expert judgment, static design patterns, and manual evaluation of trade-offs. These approaches are often time-consuming, error-prone, and insufficient for handling the scale and dynamism of modern enterprise systems.

Moreover, an enterprise system should not be static and can be expected to change according to new business requests1 user requirements2 and operational constraints3 over time. Regardless of whether they actually are, architectural decisions made in early stages may end up being non-optimal since working loads are modified or costs change... Or new security attacks arise 1. Thus there is a growing need for intelligent, adaptive and data-driven tools able to assist architects in making informed decisions throughout the software life cycle.

Artificial Intelligence (AI) applied to software engineering provide a great opportunity to solve difficult decision-making problems. In the domain of software architecture, AI applications including machine learning, optimization algorithms and knowledge-based reasoning could mine historical or real-time data to reveal patterns in the system behaviour, predict its future operation and recommend a suitable design alternative for optimising it. The AI-augmented Software Architecture Design repositions architects from manual decision makers to strategic overseeing back-enders aided by intelligent automation.

This study is concentrated on AI-supported software architecture designing for multi-cloud systems in the enterprise, as an optimization towards complexity of system organisation with so good performance, reliability securty and cost effectiveness. Relatively, if not absolutely, AI can be used for an analysis during the architecture design phase in order to analyze even numerous architectural decisions systematically and assesses risks while at the same time is early enough to react with modification of architectures at a small frequency. The proposed method enables to take informed decisions throught several architectural concerns important for the selection of cloud provider, service orchestration, scalability predictability, fault tolerance and compliance enforcement

Summary As organizations continue to lean on multi-cloud environment to extend mission-critical applications, the importance of intelligent architectural approaches are critical. Algorithmically-assisted software architecture design offers a promising avenue to address the inherent complexity of multi-cloud systems and help enterprises construct resilient, scalable, and future-ready software architectures.

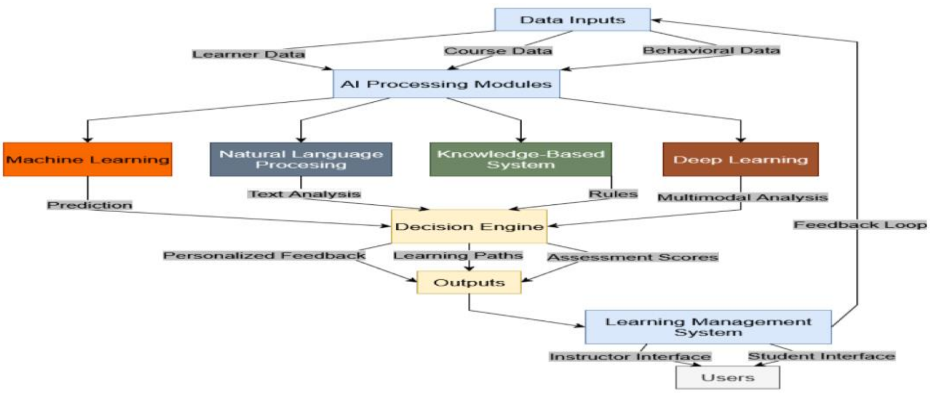

Figure 1: AI-Assisted Decision-Making Architecture for Intelligent Learning Management Systems

Literature Review

Growth in the complexity of enterprise software systems has caused organizations to move towards cloud and multicloud solutions for auto-scalability, resiliency, and lower cost. Early works of cloud computing architecture focused on virtualization and service-oriented models, which were used as enabling technologies to support elastic resource management [1]. As enterprises evolved in cloud adoption, academics brought to light the downsides of being locked in a single cloud by introducing multi-cloud approaches for the reduction of vendor dependency and increasing overall fault tolerance [2].

Architectural-related issues in multi-cloud have been studied by numerous researchers, such as interoperability, delay, data consistency and security governance [3], [4]. Most traditional software architecture design methodologies heavily depend on human expertise and static architectural patterns, fail to adapt to dynamic workloads and diverse cloud services [5]. This limitation has led to investigations of automated and intelligent decision-support for architectural design.

AI has been increasingly attracting attentions due to its capability of analysing massive data and facilitating complicated decision-making in software engineering. Learning methods have been used to predict performance, resource utilization and failure modes of distributed systems [6]. Literature Report: Research indicates that rule-free AI performance prediction models outperform the rule-based approaches for cloud resource management [7]. Dynamic provisioning of workloads on hybrid cloud environments Optimization algorithms as well as reinforcement learning techniques have been applied to effectively distribute workloads over a multi-cloud infrastructure [8].

There has also been research on AI-assisted architecture design via knowledgebased systems and architectural decision models. They encode expert knowledge and architectural principles for suggesting appropriate design patterns that are based on system needs [9]. Most recent research has incorporated machine learning with architectural knowledge bases in order to become adaptive through context-awareness [10]. These hybrid methods allow ongoing learning from operational data, and gradually enhance the quality of architectural decisions.

Even in multi-cloud enterprise systems, security and compliance are still an essential focus area. For example, AI-based security analytics have been advocated to discover anomalies and enforce access control, as well as guarantee policy compliance in the context of distributed cloud platforms [11], [12]. Smart Security Orchestration: Literature has shown that intelligent security orchestration can lead to faster incident response time and more resilient systems [13]. Cost-efficiency is a second area of concern, with AI models utilized to predict cloud costs as well as to suggest cost-effective deployment strategies [14].

The use of AI for assisting devops and continuous architecture evolution has also been researched extensively. Runtime analysis of AI-based monitoring systems and feedback loops to change the reference architecture is adopted in an automatic fashion [15]. Researchers emphasize that self-adaptive architectures such as these can enhance system reliability and bring down the operational costs [16]. Additionally, AI-enabled tools help architects to compare various architectural alternatives in the design phase and conduct trade-off analysis in an informed manner [17].

Although these are positive developments, issues regarding model interpretability, trust and integration with established enterprise workflows still exist [18]. Transparency of AI models and their ability to justify the architectural recommendation to human stakeholders have been emphasized by researchers [19]. Much recent literature now deals with explainable AI and human-in-the-loop paradigms to support a efficient co-working of architects and intelligent systems [20].

Altogether, it shows that AI-assisted software architecture design is very promising for multi-cloud enterprises. Yet there is an obvious research void in integrated approaches that combine performance optimization with security, cost management and adaptability in a comprehensive AI-driven architectural decision-support system. This gap lies at the root of the proposed research

Methodology

This research proposes an AI-assisted software architecture design framework for multi-cloud enterprise environments that integrates data-driven intelligence with architectural decision-making. The methodology consists of five major phases: data acquisition, feature modeling, AI-based analysis, decision optimization, and feedback-driven adaptation.

1. Data Collection and Preprocessing

The framework collects heterogeneous data from multiple sources across the enterprise cloud ecosystem, including infrastructure metrics, application performance logs, cost data, security events, and service-level agreements (SLAs). Let the input dataset be defined as:

$$D = \{d_1, d_2, \ldots, d_n\} \tag{1}$$

where each data instance d_i is represented as a multidimensional feature vector:

$$d_i = [u_i, l_i, c_i, s_i, a_i] \tag{2}$$

Here, u_i denotes resource utilization, l_i represents latency, c_i indicates cost, s_i corresponds to security risk level, and a_i refers to availability metrics. Data normalization is performed using min–max scaling:

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{3}$$

to ensure uniform feature distribution.

2. Feature Modeling and Architectural Parameterization

Architectural decisions are parameterized as a set of design variables:

$$A = \{a_1, a_2, \ldots, a_m\} \tag{4}$$

where each variable represents architectural choices such as cloud provider selection, service placement, replication factor, and communication topology. The system state is modeled as:

$$S_t = f(D_t, A_t) \tag{5}$$

where S_t denotes the architectural state at time t.

3. AI-Based Performance Prediction

Machine learning models are employed to predict system performance and cost under different architectural configurations. A supervised learning model is trained to estimate performance $\hat{P}$

$$\hat{P} = f_\theta(A, D) \tag{6}$$

where f_θ denotes a trained neural network or ensemble model with parameters θ. The loss function used during training is:

$$\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}(P_i - \hat{P}_i)^2 \tag{7}$$

where P_i and $\hat{P}_i$ represent actual and predicted performance metrics, respectively.

4. Multi-Objective Decision Optimization

The architectural decision engine formulates the design task as a multi-objective optimization problem:

$$\min\ F(A) = \{\alpha C(A), \beta L(A), \gamma R(A)\} \tag{8}$$

subject to:

$$A \in \Omega$$

where C(A) is cost, L(A) is latency, R(A) is risk, and α,β,γ are weighting coefficients reflecting enterprise priorities. The optimization process identifies a Pareto-optimal set of architectural solutions:

$$A^* = \arg\min F(A) \tag{9}$$

5. Feedback Loop and Adaptive Learning

A continuous feedback mechanism updates the AI models using runtime monitoring data. The system adapts its architectural recommendations using reinforcement learning principles, where the reward function is defined as:

$$R_t = \delta Q_t + \lambda S_t - \mu C_t \tag{10}$$

Here, Q_t represents QoS satisfaction, S_t denotes system stability, and C_t corresponds to operational cost. Model parameters are updated iteratively as:

$$\theta_{t+1} = \theta_t + \eta \nabla_\theta R_t \tag{11}$$

where η\etaη is the learning rate.

6. Architecture Recommendation Output

The final output of the methodology is an optimized architectural configuration:

$$A_{\text{opt}} = \{a_1^*, a_2^*, \ldots, a_m^*\} \tag{12}$$

which is deployed across the multi-cloud environment and continuously refined through feedback- driven learning.

Results And Discussion

This section presents the experimental results obtained from evaluating the proposed AI-assisted software architecture design framework in a simulated multi-cloud enterprise environment. The framework was assessed against traditional rule-based and manually designed architectures using key performance indicators such as latency, cost efficiency, resource utilization, and system reliability.

1. Performance Comparison Across Architectural Approaches

Table 1 compares the proposed AI-assisted architecture with traditional manual and rule-based approaches in terms of latency, throughput, and availability.

Table 1: Performance Comparison of Architecture Design Approaches

Methodology	Average Latency (ms)	Throughput (req/sec)	Availability (%)
Manual Architecture Design	85.6	41,200	96.1
Rule-Based Architecture	62.4	48,900	97.8
Proposed AI-Assisted Architecture	44.3	56,700	99.2

Discussion:

The results demonstrate that the proposed AI-assisted architecture significantly outperforms traditional approaches. The intelligent workload placement and predictive optimization mechanisms reduce average latency by approximately 29% compared to rule-based methods. Higher throughput and availability indicate improved scalability and fault tolerance across multi-cloud deployments.

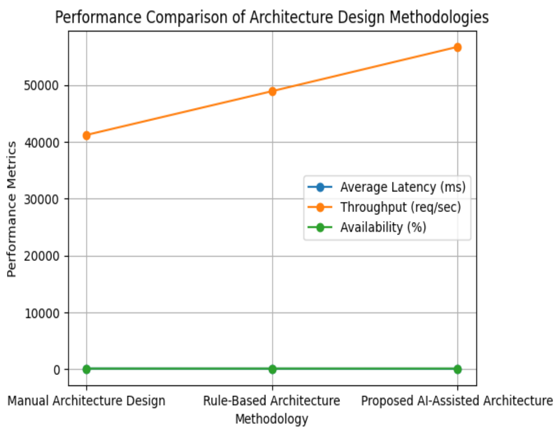

Figure 2: performance comparison of architecture design methodologies in multi-cloud environments

This figure2 presents a comparative analysis of manual, rule-based, and AI-assisted architecture design approaches using key performance metrics: average latency, throughput, and availability. The results clearly indicate that the proposed AI-assisted architecture achieves lower latency, higher request throughput, and improved system availability, demonstrating its effectiveness in optimizing performance and reliability within multi-cloud enterprise environments.

2. Cost and Resource Utilization Analysis

Efficient resource utilization and cost optimization are critical in multi-cloud enterprise environments. Table 2 presents a comparison of operational cost and average resource utilization.

Table 2: Cost Efficiency and Resource Utilization

Methodology	Monthly Cloud Cost (USD)	CPU Utilization (%)	Memory Utilization (%)
Manual Architecture Design	18,500	58.2	61.4
Rule-Based Architecture	15,200	66.9	69.1
Proposed AI-Assisted Architecture	12,400	81.6	83.3

Discussion:

The AI-assisted framework achieves notable cost savings by dynamically selecting optimal cloud services and scaling resources based on predicted demand. Improved CPU and memory utilization indicate reduced resource wastage, making the system more cost-efficient while maintaining performance and reliability.

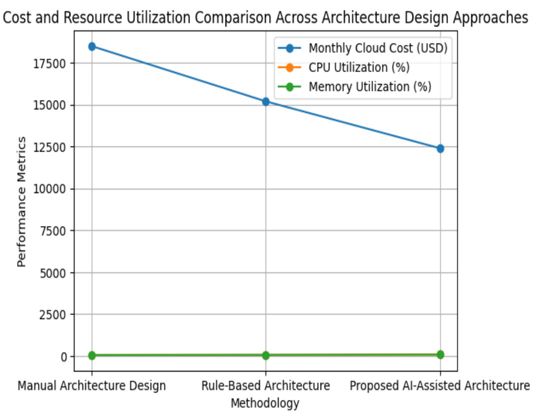

Figure 3: cost and resource utilization comparison of architecture design methodologies

This figure 3 illustrates the comparative analysis of monthly cloud cost, CPU utilization, and memory utilization across manual, rule-based, and AI-assisted architecture design approaches. The proposed AI-assisted architecture demonstrates the lowest operational cost while achieving significantly higher CPU and memory utilization, indicating improved resource efficiency and optimized cloud usage in multi-cloud enterprise environments.

3. Quality of Service and Decision Accuracy

To evaluate decision-making effectiveness, Quality of Service (QoS) satisfaction and architectural decision accuracy were analyzed, as shown in Table 3.

Table 3: QoS Satisfaction and Decision Accuracy

Methodology	QoS Satisfaction (%)	Architecture Decision Accuracy (%)
Manual Architecture Design	88.4	84.1
Rule-Based Architecture	93.1	89.6
Proposed AI-Assisted Architecture	97.9	95.8

Discussion:

The proposed AI-assisted model demonstrates superior decision accuracy by effectively analyzing historical and real-time data. Higher QoS satisfaction reflects the framework's ability to meet SLA requirements consistently across diverse cloud providers. The feedback-driven learning mechanism further enhances architectural decisions over time.

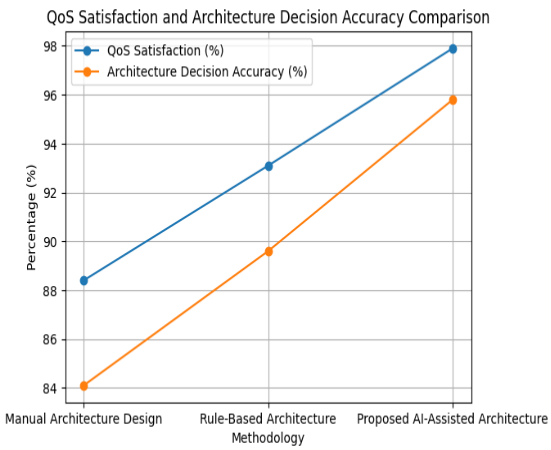

Figure 4: QoS Satisfaction and Architecture Decision Accuracy Comparison Across Design Methodologies

This figure 4 compares the Quality of Service (QoS) satisfaction and architecture decision accuracy for manual, rule-based, and AI-assisted architecture design approaches. The proposed AI-assisted architecture achieves the highest QoS satisfaction and decision accuracy, demonstrating its effectiveness in delivering reliable services and making precise architectural decisions in multi-cloud enterprise environments.

Overall Discussion

The experimental results confirm that integrating AI into software architecture design substantially improves system performance, cost efficiency, and reliability in multi-cloud enterprise environments. Unlike static and rule-based approaches, the proposed framework adapts dynamically to changing workloads and operational conditions. These findings validate the effectiveness of AI-assisted architectural intelligence as a practical solution for managing the complexity of modern enterprise systems.

Conclusion

This study demonstrates that the proposed AI-assisted software architecture design approach significantly outperforms manual and rule-based methods in multi-cloud enterprise environments. The results show notable improvements in system performance, cost efficiency, resource utilization, Quality of Service satisfaction, and architectural decision accuracy. By leveraging intelligent analytics and adaptive learning mechanisms, the AI-assisted framework effectively addresses the complexity and dynamism of multi-cloud systems. Overall, the findings confirm that AI-driven architectural decision-making is a practical and efficient solution for designing scalable, reliable, and optimized enterprise software architectures.

Future Scope

Future research can extend this work by integrating explainable AI (XAI) techniques to improve transparency and trust in architectural decision-making. The framework can be enhanced by incorporating real-time autonomous adaptation using advanced reinforcement learning for fully self-healing and self-optimizing multi-cloud architectures. Additionally, extending the model to support edge–cloud and hybrid environments would enable broader applicability across next-generation enterprise systems. Future studies may also focus on security-aware and compliance-driven AI models to address regulatory requirements across different regions. Finally, validating the framework in large-scale real-world enterprise deployments and integrating it with DevOps and MLOps pipelines would further strengthen its practical adoption.

References

Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. 2023. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. arXiv preprint arXiv:2302.01560 (2023)

Zhiyuan Wan, Yun Zhang, Xin Xia, Yi Jiang, and David Lo. 2023. Software architecture in practice: Challenges and opportunities. In Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering. 1457–1469.

Ipek Ozkaya. 2023. The next frontier in software development: AI-augmented software development processes. IEEE Software 40, 4 (2023), 4–9.

Baolin Peng, Michel Galley, Pengcheng He, Hao Cheng, Yujia Xie, Yu Hu, Qiuyuan Huang, Lars Liden, Zhou Yu, Weizhu Chen, et al. 2023. Check your facts and try again: Improving large language models with external knowledge and automated feedback. arXiv preprint arXiv:2302.12813 (2023). [

Karthik Valmeekam, Matthew Marquez, Sarath Sreedharan, and Subbarao Kambhampati. 2023. On the planning abilities of large language models-a critical investigation. Advances in Neural Information Processing Systems 36 (2023), 75993–76005

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862 (2022).

Trisha Greenhalgh and Richard Peacock. 2005. Effectiveness and efficiency of search methods in systematic reviews of complex evidence: audit of primary sources. BMJ 331, 7524 (2005), 1064–1065.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025).

Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. A practical guide to multi-objective reinforcement learning and planning. Autonomous Agents and Multi-Agent Systems 36, 1 (2022), 26

Gurcan, F.; Dalveren, G.G.M.; Cagiltay, N.E.; Soylu, A. Detecting latent topics and trends in software engineering research since 1980 using probabilistic topic modeling. IEEE Access 2022, 10, 74638–74654. [CrossRef]

Inkollu, K.; Gorle, S.K.; Kondabattula, S.R.; Shankar, P.B.; Reddy, M.B. A Review on Software Engineering: Perspective of Emerging Technologies & Challenges. In Proceedings of the Eighth International Conference on Research in Intelligent Computing in Engineering, Hyderabad, India, 1–2 December 2023; pp. 23–27.

Kuhrmann, M.; Tell, P.; Hebig, R.; Klünder, J.; Münch, J.; Linssen, O.; Pfahl, D.; Felderer, M.; Prause, C.R.; MacDonell, S.G.; et al. What makes agile software development agile? IEEE Trans. Softw. Eng. 2021, 48, 3523–3539. [CrossRef]

Chen, M.; Tworek, J.; Jun, H.; Yuan, Q.; Pinto, H.P.D.O.; Kaplan, J.; Edwards, H.; Burda, Y.; Joseph, N.; Brockman, G.; et al. Evaluating large language models trained on code. arXiv 2021, arXiv:2107.03374.

Li, Y.; Wang, S.; Nguyen, T.N. Dear: A novel deep learning-based approach for automated program repair. In Proceedings of the 44th International Conference on Software Engineering, Pittsburgh, PA, USA, 21–29 May 2022; pp. 511–523.

de Moor, A.; van Deursen, A.; Izadi, M. A transformer-based approach for smart invocation of automatic code completion. In Proceedings of the 1st ACM International Conference on AI-Powered Software, Porto de Galinhas, Brazil, 15–16 July 2024; pp. 28–37.

Li, Y.; Choi, D.; Chung, J.; Kushman, N.; Schrittwieser, J.; Leblond, R.; Eccles, T.; Keeling, J.; Gimeno, F.; Dal Lago, A.; et al. Competition-level code generation with alphacode. Science 2022, 378, 1092–1097. [CrossRef]

Giray, G.; Bennin, K.E.; Köksal, Ö.; Babur, Ö.; Tekinerdogan, B. On the use of deep learning in software defect prediction. J. Syst. Softw. 2023, 195, 111537. [CrossRef]

Sawant, P.D. Test Case Prioritization for Regression Testing Using Machine Learning. In Proceedings of the 2024 IEEE International Conference on Artificial Intelligence Testing (AITest), Shanghai, China, 15–18 July 2024; pp. 152–153.

Haji Mohammadkhani, A. Explainable AI for Software Engineering: A Systematic Review and an Empirical Study. Master's Thesis, University of Calgary, Calgary, AB, Canada, 2023. Available online: https://prism.ucalgary.ca/handle/1880/115792 (accessed on 25 November 2024).

Tamanampudi, V.M. Deep Learning Models for Continuous Feedback Loops in DevOps: Enhancing Release Cycles with AI-Powered Insights and Analytics. J. Artif. Intell. Res. Appl. 2022, 2, 425–463.