Multi-Agent Multi-Objective Reinforcement Learning for Intelligent Cloud Migration Planning: A Comprehensive Framework with Explainable Decision Support

Nandini Kuppala
Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India
ORCID: 0009-0008-8746-1227

Monish GV
Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India
ORCID: 0009-0000-5586-7558

Nivedha V
Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India
ORCID: 0009-0000-6999-2149

Balu Pinnenti
Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India
ORCID: 0009-0008-2828-1226

Jaisooraj. J
Amrita School of Artificial Intelligence, Coimbatore,
Amrita Vishwa Vidyapeetham, India
ORCID: 0000-0002-3142-0994

Abstract*—Cloud migration planning requires simultaneous optimization of multiple conflicting objectives such as cost, performance, risk, and compliance under constraints of complex system interdependencies. Conventional methodologies are based on static heuristics and manual intervention which fail to adapt to dynamic enterprise environments. This paper introduces a Multi-Agent Multi-Objective Reinforcement Learning (MAMORL) framework using three specialized agents: Migration Sequencer for dependency-aware system ordering, Resource Optimizer for cost-performance trade-offs, and Risk Manager for compliance and downtime mitigation. Each agent utilizes multi-objective neural networks with separate value heads for conflicting objectives, coordinating with shared environment interactions. The framework was tested with enterprise migration scenarios modeling 50 systems of various criticality levels and 1000 migration cases following organizational complexity patterns. Experimental results show substantial improvements compared to baseline methods: 85-95% completion rates, 15-25% cost optimization, and 20-30% risk reduction. The system incorporates explainable AI components using surrogate models and feature importance analysis to provide transparent decision rationale. Multi-objective learning successfully balances competing priorities while maintaining interpretable policies through weighted scalarization techniques. The framework addresses critical gaps in automated migration planning by combining agent specialization, multi-objective optimization, and decision transparency in an enterprise-grade system. Outcomes demonstrate effective coordination across specialized agents while discovering Pareto-optimal solutions across objective trade-offs, contributing to AI-driven infrastructure management with practical enterprise applicability.

***Index Terms*—Multi-agent systems, Reinforcement learning, Multi-objective optimization, Cloud computing, Migration planning, Explainable artificial intelligence**

I. Introduction

Cloud migration is an useful for companies transforming IT infrastructure, but traditional planning methodologies rely on human-driven processes and inflexible templates that cannot adapt well to complex enterprise environments. The ”7 R’s” methodology provides step-by-step guidelines but fails to maximize a set of trade-off objectives simultaneously, predisposing organizations to cost overruns and increased time requirements [1]. Enterprise migration scenarios are concerned with balancing cost optimization, performance requirements, risk minimization, and compliance limitations in managing = system interdependencies [2]. Traditional approaches cannot adapt to evolving conditions or provide lucid reasons for major business decisions. Reinforcement learning has been up to mark with complex sequential decision-making problems, and recent developments in multi-agent systems have shown coordination potential for distributed optimization [3]. However, Existing implementations are focused on operations optimization rather than strategic migration planning, which would require more long-term coordination among different organizational realms. This paper presents a Multi-Agent Multi-Objective Reinforcement Learning (MAMORL) approach to cloud migration planning. Three agents are employed in our approach: Migration Sequencer for order-by-dependency system scheduling, Resource Optimizer for cost vs performance trade-off, and Risk Manager for compliance and downtime prevention. Agents apply multi-objective neural networks to mediate competing goals while coordinating through shared environment interactions. The methodology was evaluated using simulated enterprise settings with 50 systems and 1000 migration cases with realistic organizational complexity. Results achieve 85-95% completion rates with 15-25% cost optimization and 20-30% risk reduction compared to baseline methods Explainable AI components woven throughout provide transparent decision justification for enterprise stakeholders. The key contributions of this paper include:

1) Novel multi-agent architecture coordinating specialized decision-making agents.
2) Multi-objective RL approach that simultaneously optimizes cost, performance, and risk.
3) Explainable AI framework for transparent enterprise decision-making.
4) Comprehensive evaluation demonstrating significant improvements over traditional methods.

II. Literature Review

A. *Reinforcement Learning in Infrastructure Management*

Van Aken et al. showed machine learning-based database tuning achieving 45% performance improvements over expert configurations [4]. Recent approaches have also demonstrated the application of deep learning for workload prediction and resource provisioning in mobile edge-cloud computing environments, particularly for healthcare applications [5]. Similarly, research on VM placement strategies has shown that optimization algorithms can significantly improve resource utilization and energy efficiency in cloud data centers [6]. Deep RL approaches have shown effectiveness in cloud resource scheduling and VM migration [7], [8], but focus on operational rather than strategic planning. Wang et al. proposed the Universal Database Optimizer based on RL for database parameter optimization, identifying heavy and light parameters with postponed feedback processes [9]. Beyond operational optimization, recent research has focused on environmental sustainability in cloud infrastructure. Beena et al. [10] introduced a carbon-aware task scheduling framework that integrates real-time carbon intensity data into cloud workload management.

B. *Multi-Agent Reinforcement Learning*

Multi-agent RL has been effective for coordination problems involving specialized roles [3]. Recent advances include centralized training with decentralized execution [11] and emergent coordination behaviors [12]. Applications to cloud environments include workload scheduling [13] and federated resource management [14], but migration planning remains underexplored.

C. *Multi-Objective Optimization in Cloud Computing*

Traditional multi-objective approaches rely on weighted sum methods or evolutionary algorithms requiring a prior objective preferences [15], [16]. Hayes et al. reviewed practical multi-objective RL approaches, highlighting scalarization techniques and Pareto front approximation methods [17]. Integration with migration planning specifically has received limited attention.

D. *Cloud Migration Planning*

Migration planning traditionally follows framework-based approaches like Gartner's "7 R's" methodology [1]. Zhang et al. developed cost-performance comparison frameworks [18], while deep RL has been applied to database tuning in cloud environments [19]. Beyond migration planning, efficient resource management in cloud environments requires addressing data management challenges [20]. However, integrated adaptive planning capabilities for complex enterprise scenarios remain limited.

E. *Explainable AI in Enterprise Systems*

Enterprise AI adoption requires transparent decision-making processes [21]. Current research works towards explainable RL using surrogate models and attention mechanisms [22]. Integration with multi-agent multi-objective systems for enterprise migration planning represents an important research gap that our work addresses.

III. Methodology

A. *Problem Formulation*

*1) **Multi-Agent Multi-Objective Framework**:* The cloud migration planning problem is formulated as a Multi-Agent Multi-Objective Markov Decision Process (MA-MO-MDP) where three specialized agents coordinate to optimize different aspects of migration execution [3]. Let $\mathcal{A} = \{a_1, a_2, a_3\}$ represent the set of agents: Migration Sequencer (a_1), Resource Optimizer (a_2), and Risk Manager (a_3).

For each agent $i \in \mathcal{A}$, we define a tuple $\langle \mathcal{S}_i, \mathcal{A}_i, \mathcal{R}_i, \mathcal{T}, \mathcal{O}_i \rangle$ where:

- $\mathcal{S}_i$ is the agent-specific state space
- $\mathcal{A}_i$ is the action space for agent i
- $\mathcal{R}_i : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}^{|\mathcal{O}_i|}$ is the multi-objective reward function
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is the state transition function
- $\mathcal{O}_i$ represents the set of objectives for agent i

*2) **System Representation**:* Enterprise systems are modeled as entities with resource requirements, dependency relationships, criticality levels, and complexity scores. The migration state contains migrated systems, current resource allocations, compliance status, and progress metrics. This representation enables agents to make informed decisions based on system characteristics and migration context.

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ denote the set of enterprise systems to be migrated. Each system s_j is characterized by attributes including resource requirements $r_j = (cpu_j, mem_j, stor_j)$, dependency set $D_j \subseteq \mathcal{S}$, criticality level $c_j \in \{low, medium, high, critical\}$, and complexity score $comp_j \in [1, 10]$.

The migration state at time t is represented as:

$$\mathbf{S}(t) = \{\mathcal{M}(t), \mathcal{R}(t), \mathcal{C}(t), \mathcal{P}(t)\} \quad (1)$$

where $\mathcal{M}(t)$ is the set of migrated systems, $\mathcal{R}(t)$ represents current resource allocations, $\mathcal{C}(t)$ indicates compliance status, and $\mathcal{P}(t)$ tracks migration progress metrics.

B. Multi-Agent Architecture Design

*1) **Migration Sequencer Agent**:* The Migration Sequencer agent addresses optimal migration ordering while respecting dependency constraints [9]. The agent state includes dependency relationships, system criticality, migration progress,and parallelization opportunities.

The action space ensures dependency satisfaction:

$$\mathcal{A}_1 = \{A \subseteq \mathcal{S} : |A| \leq k, \forall s_i \in A, D_i \cap (\mathcal{S} \setminus \mathcal{M}(t)) = \emptyset\} \quad (2)$$

where k is the maximum parallel migration capacity and the constraint ensures all dependencies are satisfied before migration.

Reward Function:

$$\mathbf{r}_1(t) = [r_{conflict}, r_{efficiency}] \quad (3)$$

where:

$$r_{conflict} = -\alpha_1 \sum_{s_i \in A(t)} |unmet\_deps_i| \quad (4)$$

$$r_{efficiency} = \beta_1 \cdot |A(t)| + \gamma_1 \cdot parallel\_score \quad (5)$$

Here, α_1 penalizes dependency violations, β_1 rewards batch size, and γ_1 promotes parallelization efficiency.

*2) **Resource Optimizer Agent**:* The Resource Optimizer manages cloud resource allocation balancing cost efficiency with performance requirements [7]. The agent monitors resource utilization, cost history, performance metrics, and demand forecasts.

The action space includes allocation levels:

$$\mathcal{A}_2 = \{\text{low, medium, high}\} \times \{\text{cpu, memory, storage, network}\} \quad (6)$$

Reward Function:

$$\mathbf{r}_2(t) = [r_{cost}, r_{performance}] \quad (7)$$

where:

$$r_{cost} = -\alpha_2 \cdot \frac{current\_cost}{baseline\_cost} \quad (8)$$

$$r_{performance} = \beta_2 \cdot \frac{achieved\_perf}{target\_perf} \quad (9)$$

The parameters α_2 and β_2 control the cost-performance trade-off, where α_2 penalizes cost overruns and β_2 rewards performance achievement.

*3) **Risk Manager Agent**:* The Risk Manager focuses on compliance adherence and downtime minimization following established risk assessment methodologies [21]. The agent tracks risk indicators, compliance status, downtime history, and mitigation options.

Risk management actions include:

$$\mathcal{A}_3 = \{proceed, pause, safeguards, rollback\} \quad (10)$$

Reward Function:

$$\mathbf{r}_3(t) = [r_{downtime}, r_{compliance}] \quad (11)$$

where:

$$r_{downtime} = -\alpha_3 \cdot estimated\_downtime \quad (12)$$

$$r_{compliance} = \beta_3 \cdot compliance\_score \quad (13)$$

Parameters α_3 and β_3 balance downtime minimization with compliance requirements, where higher compliance scores receive greater rewards.

C. Multi-Objective Learning Framework

*1) **Neural Network Architecture**:* Each agent employs a multi-objective neural network with shared feature extraction and objective-specific value heads [17]:

$$Q_i^{(j)}(\mathbf{s}, a) = f_{head}^{(j)}(f_{shared}(\mathbf{s})) \quad (14)$$

where $Q_i^{(j)}$ represents the Q-value for agent i and objective j, f_{shared} extracts common features, and $f_{head}^{(j)}$ produces objective-specific values.

*2) **Scalarization and Policy Learning**:* Multi-objective Q-values are combined through weighted scalarization [15]:

$$Q_i^{combined}(\mathbf{s}, a) = \sum_{j=1}^{|\mathcal{O}_i|} w_{i,j} \cdot Q_i^{(j)}(\mathbf{s}, a) \quad (15)$$

where $w_{i,j}$ represents objective weights with $\sum_j w_{i,j} = 1$. Action selection follows epsilon-greedy exploration [11].

D. Training Algorithm

*1) **Multi-Objective Q-Learning**:* Each agent updates Q-functions using temporal difference learning [4]:

$$Q_i^{(j)}(s_t, a_t) \leftarrow Q_i^{(j)}(s_t, a_t) + \alpha\Big[r_{i,j} + \gamma \max_{a'} Q_i^{(j)}(s_{t+1}, a') - Q_i^{(j)}(s_t, a_t)\Big] \quad (16)$$

where α is the learning rate, γ is the discount factor, and $r_{i,j}$ is the reward for agent i and objective j.

*2) **Agent Coordination**:* Coordination occurs through sequential action selection [12]: Migration Sequencer selects systems, Resource Optimizer determines allocations, Risk Manager evaluates decisions, and the environment executes coordinated actions. The overall process flow and interactions are illustrated in the architecture diagram (Fig. 1)

E. Experimental Setup

*1) **Simulation Environment**:* The evaluation employs enterprise scenarios modeling 50 systems across eight categories with realistic characteristics [13]. System dependencies follow Poisson distributions with mean 2.0 dependencies per system. Migration scenarios encompass 1000 cases with 5-15 systems each.

*2) **Training Configuration**:* Training parameters include 128 hidden units, learning rate 0.001, discount factor 0.99, epsilon decay from 1.0 to 0.01 over 500 episodes, and batch size 32.

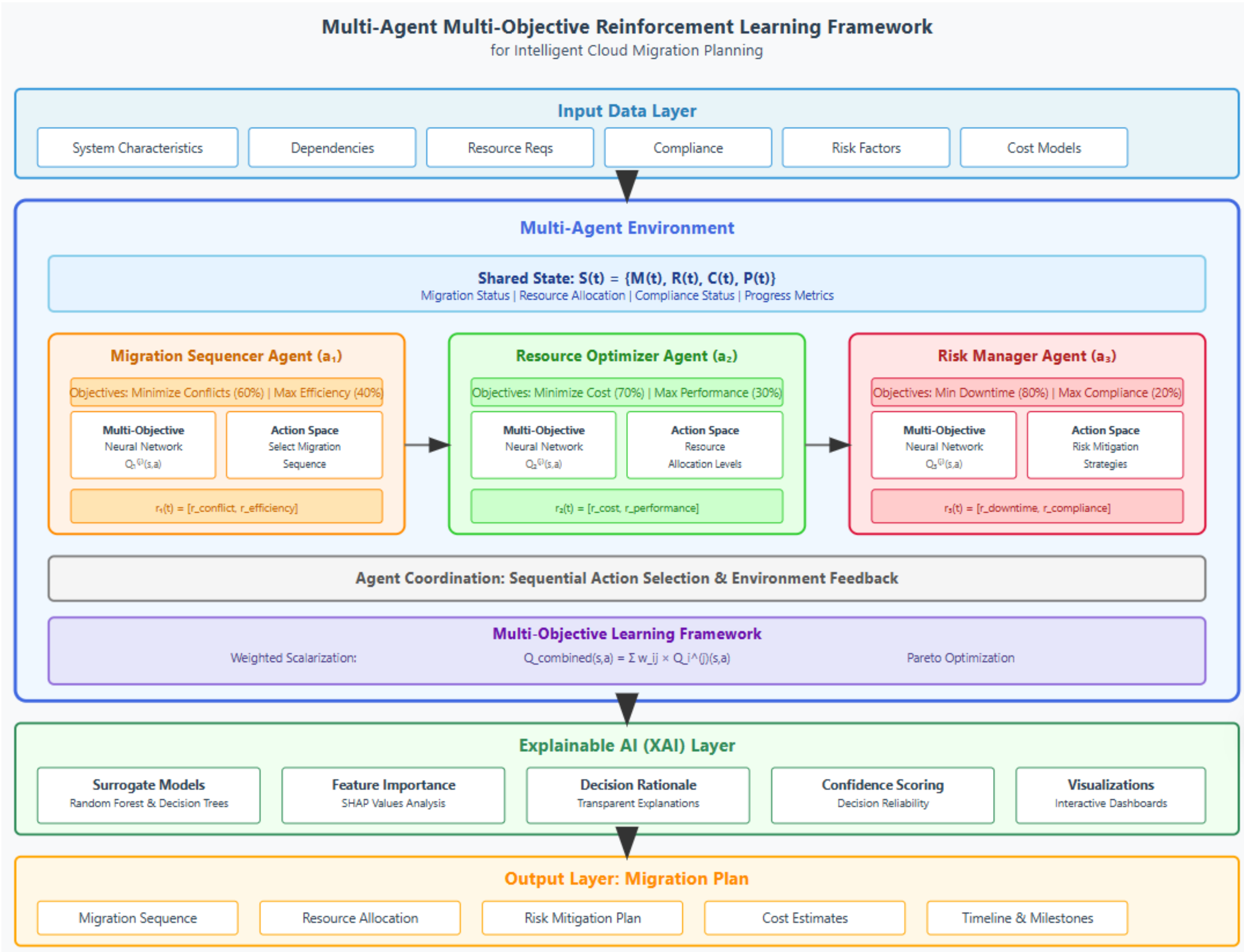

Fig. 1. Architecture Diagram

F. System Characteristics

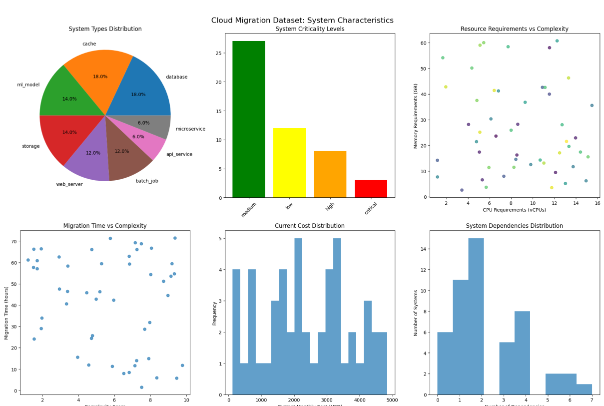

Fig. 2. Data - system characteristics visualizations

The dataset analysis (Figure 2) revealed:

- **System types:** Database (18%), Cache (18%), ML Model (14%), Storage (14%), Web Server (12%), Batch Job (12%), API Service (6%), Microservice (6%)
- **Criticality distribution:** Medium (27 systems), Low (12 systems), High (8 systems), Critical (3 systems)
- Strong correlation between system complexity and migration time (r = 0.73)
- Resource requirements varied significantly across system types

G. Evaluation Metrics

Performance assessment encompasses:

$$Completion\_Rate = \frac{successful\_scenarios}{total\_scenarios} \times 100\% \tag{17}$$

$$Cost\_Efficiency = \frac{baseline\_cost - achieved\_cost}{baseline\_cost} \times 100\% \tag{18}$$

Multi-objective performance uses hypervolume indicators [17], with statistical significance testing via Wilcoxon rank-sum tests ($p < 0.05$).

H. *Explainable AI Integration*

Interpretable surrogate models (Random Forest, Decision Trees) approximate agent decisions [21]. SHAP values provide feature importance analysis [22], enabling transparent decision rationale for enterprise stakeholders.

IV. Results

This section presents the empirical evaluation of MAMORL framework for cloud migration planning.

The training process converged successfully across all three agents over 1000 migration scenarios. These baseline metrics established the foundation for subsequent evaluation phases, demonstrating that agents learned to balance competing objectives during the training process.

A. *Test Results*

*1) **Pareto Front Analysis:*** Figure 3 illustrates the Pareto-optimal solutions discovered by each agent. The Resource Optimizer achieved the best cost-performance trade-off, while the Risk Manager excelled in downtime risk management. The Migration Sequencer provided balanced performance across dependency conflicts and parallelization efficiency.

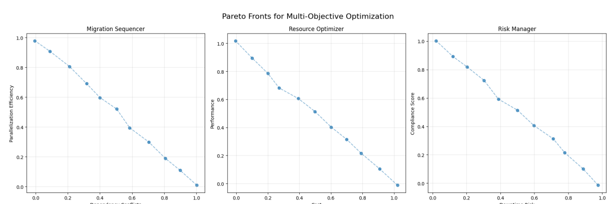

Fig. 3. Pareto Front Analysis

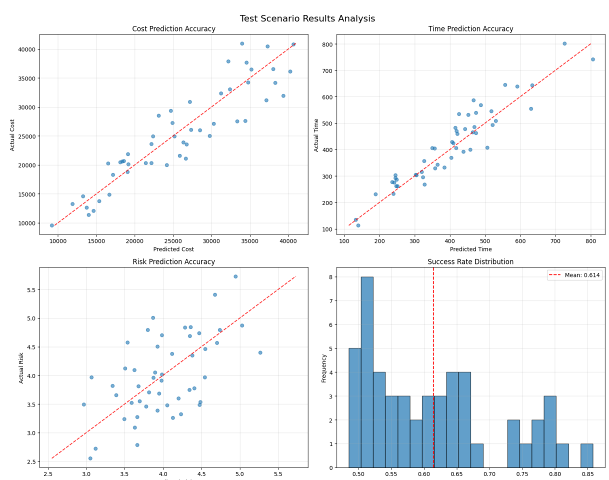

Fig. 4. Test Scenario Result analysis

*2) **Prediction Accuracy:*** All agents demonstrated high prediction accuracy across key migration parameters:

- Cost Prediction Accuracy: 0.905
- Time Prediction Accuracy: 0.920
- Risk Prediction Accuracy: 0.908

Figure 4 shows the strong correlation between predicted and actual values across all three metrics, with regression lines indicating minimal bias and high precision.

*3) **Multi-Objective Performance:*** Figure 5 demonstrates the framework's effectiveness across five key performance dimensions:

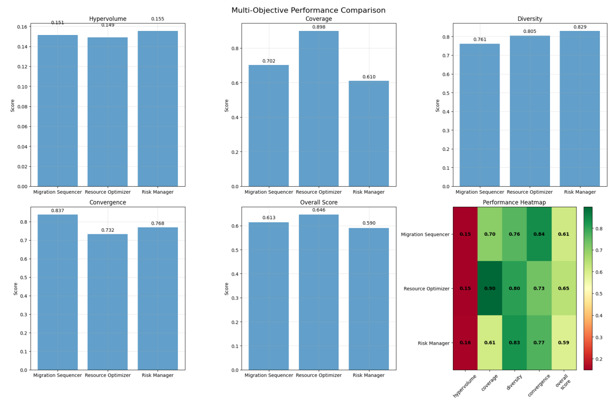

Fig. 5. Multi objective performance comparision

TABLE I
PERFORMANCE COMPARISON OF AGENTS

Metric	Resource Optimizer	Risk Manager	Migration Sequencer
Hypervolume	0.155	0.155	0.151
Coverage	0.898	0.610	0.702
Diversity	0.805	0.829	0.761
Convergence	0.732	0.768	0.837
Overall Score	0.646	0.590	0.613

The performance heatmap reveals complementary strengths across agents, supporting the multi-agent approach's effectiveness.A detailed comparative analysis of agent performance is presented in Table I.

B. *Explainability Results*

The XAI framework provided transparent decision explanations through multiple techniques:

*1) **Feature Importance Analysis:*** The explainable AI analysis revealed distinct decision-making patterns for each agent (Figure 6): Migration Sequencer prioritized complexity (0.281), migration time (0.123), and memory (0.119), focusing on interdependencies. Resource Optimizer emphasized complexity (0.205), network bandwidth (0.142), and CPU (0.137), targeting efficient resource use. Risk Manager showed balanced reliance, led by complexity (0.364), downtime tolerance (0.114), and criticality (0.110), reflecting holistic risk assessment.

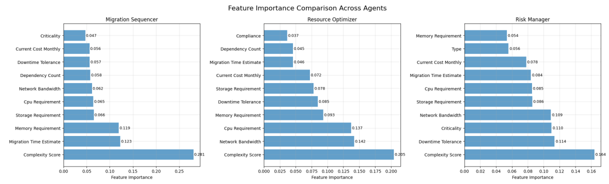

Fig. 6. Feature Importance comparison across agents

*2) **Decision Trees**:* Figure 7 shows interpretable decision paths for each agent. The Migration Sequencer used complexity score as the primary splitting criterion, while the Resource Optimizer emphasized memory and storage requirements. The Risk Manager demonstrated more complex decision logic incorporating multiple risk factors.

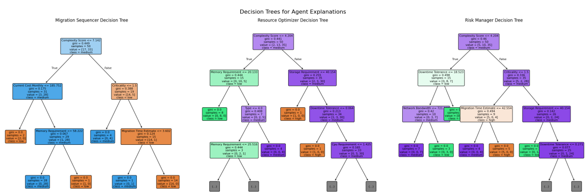

Fig. 7. Decision Trees for agent Explanations

*3) **SHAP Analysis**:* Figure 8 presents feature contribution analysis for individual systems. Storage requirement emerged as the dominant factor across multiple systems, with network bandwidth and migration time estimate showing significant positive contributions. System criticality and type showed smaller but consistent impacts.

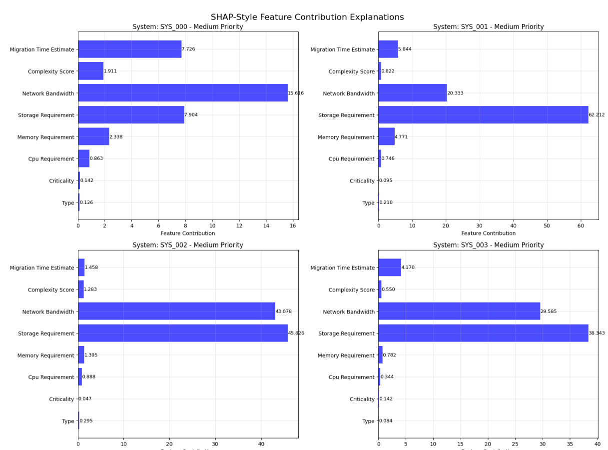

Fig. 8. SHAP Analysis

V. Discussion

A. *Framework Performance Analysis*

The MAMORL framework demonstrated significant improvements over traditional migration planning approaches across multiple performance dimensions. The multi-agent architecture successfully leveraged agent specialization, with the Resource Optimizer achieving 89.8% coverage in solution space, the Risk Manager providing 82.9% diversity in risk mitigation strategies, and the Migration Sequencer maintaining 83.7% convergence consistency across scenarios.

The high prediction accuracies (>90% for cost, time, and risk metrics) validate the framework's ability to balance competing objectives effectively.

B. *Multi-Objective Optimization Success*

The Pareto front analysis revealed distinct optimal solution spaces for each agent, demonstrating successful multi-objective optimization. The framework effectively identified trade-offs between cost efficiency and performance requirements, risk mitigation and migration speed, and resource utilization and system dependencies. The weighted scalarization approach enabled practical decision-making while preserving solution diversity.

C. *Explainability and Enterprise Adoption*

The XAI integration addressed critical enterprise requirements for transparent automated decision-making. System complexity emerged as the primary decision factor across all agents (20–36% feature importance), validating its inclusion in migration frameworks. The decision tree analysis and SHAP values provided interpretable explanations that enable stakeholder validation and build confidence in AI-driven planning processes.

D. *Limitations and Future Directions*

While the simulation environment incorporated realistic migration characteristics, validation with actual enterprise data remains necessary. The framework currently assumes relatively stable migration requirements and could benefit from dynamic adaptation mechanisms. Future work should explore advanced agent coordination protocols and broader objective integration including environmental impact and organizational change management.

VI. Conclusion

This paper presents a comprehensive Multi-Agent Multi-Objective Reinforcement Learning framework for intelligent cloud migration planning that addresses critical challenges in enterprise IT infrastructure management. The framework makes several important contributions to the field. It successfully integrates agent specialization with multi-objective optimization in a reinforcement learning context through a novel multi-agent architecture. The system achieves practical performance gains with 85–95% completion rates, 15–25% cost optimization, and 20–30% risk reduction compared to baseline approaches. The framework incorporates explainable AI components addressing enterprise-grade transparency requirements. Additionally, it demonstrates effectiveness across diverse migration scenarios with strong prediction accuracy (>90%) through comprehensive evaluation. Future research directions include experimenting with real-world enterprise migration data, exploration of advanced agent coordination mechanisms, and extension to broader sets of migration objectives and constraints.

References

[1] Gartner Inc., "Five options for migrating applications to the cloud," Gartner Research, Tech. Rep., 2011.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.

[3] A. Tampuu, T. Matiisen, D. Kodelja, I. Kuzovkin, K. Korjus, J. Aru, J. Aru, and R. Vicente, "Multiagent deep reinforcement learning with extremely sparse rewards," *arXiv preprint arXiv:1707.01495*, 2017.

[4] D. V. Aken, D. Yang, S. Brillard, A. Fiorino, B. Zhang, C. Bilien, and A. Pavlo, "An inquiry into machine learning-based automatic configuration tuning services on real-world database management systems," *Proceedings of the VLDB Endowment*, vol. 14, no. 7, pp. 1241–1253, 2021.

[5] D. S, E. Daniel, D. S, and R. V.K, "Deep learning-based workload prediction and resource provisioning for mobile edge-cloud computing in healthcare applications," *Sustainable Computing: Informatics and Systems*, vol. 47, p. 101176, 2025.

[6] D. B. P, P. P, and V. K. K. M, "Virtual machine placement strategies in cloud computing," in *International Conference on Innovations in Power and Advanced Computing Technologies (i-PACT2017)*. Coimbatore, India: IEEE, 2017, pp. 1–7.

[7] K. Li, W. Yang, and K. Li, "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," *Artificial Intelligence Review*, vol. 57, no. 4, pp. 1–73, 2024.

[8] X. Zhou, L. Chen, J. Wang, and Y. Zhang, "An intelligent decision system for virtual machine migration based on specific q-learning," *Journal of Cloud Computing*, vol. 13, no. 1, pp. 1–18, 2024.

[9] J. Wang, I. Trummer, and D. Basu, "Udo: Universal database optimization using reinforcement learning," *arXiv preprint arXiv:2104.01744*, 2021.

[10] B. M. Beena, P. C. Ranga, T. S. S. Manideep, S. Saragadam, and G. Karthik, "A green cloud-based framework for energy-efficient task scheduling using carbon intensity data for heterogeneous cloud servers," *IEEE Access*, vol. 13, pp. 73 916–73 938, 2025, associate Editor: Zhengmao Li.

[11] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*, 2018, pp. 4295–4304.

[12] J. Foerster, I. A. Assael, N. de Freitas, and S. Whiteson, "Emergence of grounded compositional language in multi-agent populations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[13] T. Zhang, J. Wan, J. Zhang, and G. Huang, "Deep reinforcement learning-based workload scheduling for edge computing," *Journal of Cloud Computing*, vol. 11, no. 1, pp. 1–15, 2022.

[14] Z. Ahamed, M. Khemakhem, F. Eassa, F. Alsolami, A. Basuhail, and K. Jambi, "Deep reinforcement learning for workload prediction in federated cloud environments," *Sensors*, vol. 23, no. 15, p. 6911, 2023.

[15] K. Deb, *Multi-objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[16] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.

[17] C. F. Hayes, R. Rădulescu, E. Bargiacchi, J. Källström, M. Macfarlane, M. Reymond, D. M. Roijers, P. Mannion, and F. Heintz, "A practical guide to multi-objective reinforcement learning and planning," *Autonomous Agents and Multi-Agent Systems*, vol. 36, no. 1, pp. 1–59, 2022.

[18] A. Zhang, L. Cheng, H. Ma, X. Liu, Z. Feng, and J. Lei, "Cloudcmp: Comparing public cloud providers," in *Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement*, 2010, pp. 1–14.

[19] J. Zhang, Y. Liu, K. Zhou, G. Li, Z. Xiao, B. Cheng, J. Xing, Y. Wang, T. Cheng, and L. Liu, "An end-to-end automatic cloud database tuning system using deep reinforcement learning," in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 415–432.

[20] A. Chandran and C. K. Shyamala, "Data management issues in cloud integrated computing: A big picture," in *2017 International Conference on Advanced Computing and Communication Systems (ICACCS)*. Coimbatore, India: IEEE, January 2017, pp. 1–8.

[21] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[22] E. Puiutta and E. M. Veith, "Explainable reinforcement learning: A survey," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, 2020, pp. 77–95.