World Journal of Innovation and Modern Technology, Vol. 7, Issue 6, (Dec)
ISSN 2682-5910 **2024**

DOI: 10.53469/wjimt.2024.07(06).19

Deep Reinforcement Learning-based Resource Adaptive Scheduling for Cloud Video Conferencing Systems

Lei Yan[1], Shiji Zhou[1.2], Wenxuan Zheng[2], Jingyi Chen[3]

[1]Electronics and Communications Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[1.2]Computer Science, University of Southern California, CA, USA
[2]Applied Math, University of California, Los Angeles, CA, USA
[3]Electrical and Computer Engineering, Carnegie Mellon University, PA, USA
**Correspondence Author, rexcarry036@gmail.com*

Abstract: ***This paper presents a new deep learning-based resource scheduling algorithm for online video chat. The framework addresses the issues of resource allocation efficiency and service efficiency in MCU environments. A comprehensive system architecture is designed, incorporating a unified resource pool and intelligent scheduling mechanisms. The deep reinforcement learning model employs an actor-critic network structure with custom-designed state space and reward functions optimized for video conferencing workloads. The framework uses adaptive resource allocation and load balancing techniques to ensure stability in heterogeneous systems. The experimental results show a significant improvement over traditional methods, achieving a 35.2% reduction in response time, a 28.7% increase in resource utilization, and a 23.5% improvement in performance. bandwidth. The system maintains consistent performance under high loads of up to 1000 users at the same time while ensuring 99.99% service. The solution provides a flexible and powerful way to control the cloud video conferencing, as well as potential applications in the delivery of large-scale business.***

Keywords: Deep Reinforcement Learning; Cloud Computing; Video Conferencing; Resource Scheduling.

1. INTRODUCTION

1.1 Background of Cloud Video Conferencing

Cloud video conferencing has become an important communication tool for modern businesses and organizations. The rapid development of cloud technology has transformed the video conferencing process into more sophisticated and flexible solutions [1]. These systems use cloud technology to provide high-quality video communication services while managing computing and network usage.

The evolution from traditional MCU (Multipoint Control Unit) based systems to cloud-based architectures has introduced significant improvements in resource utilization and system scalability. Modern cloud video conferencing platforms employ distributed resource pools that integrate various MCUs into a unified logical system [2]. This architecture enables dynamic resource allocation and improved system reliability through automated backup mechanisms.

Recent advances in cloud computing and virtualization technologies have enabled the development of private cloud solutions specifically designed for video conferencing applications. These solutions offer enhanced security, better resource control, and improved quality of service compared to public cloud alternatives. The integration of cloud computing with video conferencing has also facilitated the implementation of intelligent resource management strategies [3] [4] [5].

1.2 Challenges in Resource Management

Resource management in cloud video conferencing systems presents several critical challenges. The dynamic nature of network conditions and user demands requires sophisticated scheduling algorithms to maintain optimal performance [6]. One significant challenge is the efficient allocation of MCU resources across distributed pools while ensuring minimal latency and maximum resource utilization.

The difference in video conferencing work reflects the difficulty in scheduling. Different conferences can have different video quality, number of participants, and networking events. Traditional scheduling systems often struggle to handle these different situations efficiently, resulting in suboptimal resource utilization and poor user experience [7].

Network bandwidth fluctuations and varying connection qualities among participants pose additional challenges. The system must adaptively adjust resource allocation strategies to maintain acceptable video quality while preventing network congestion. Furthermore, the need for real-time processing and low-latency communication adds constraints to resource scheduling decisions.

Load balancing across multiple MCUs remains a critical challenge in distributed environments. Uneven distribution of conference loads can lead to performance bottlenecks and reduced system efficiency. The system must also handle hardware failures and network disruptions through effective backup strategies while maintaining service continuity [8].

1.3 Deep Reinforcement Learning Overview

Deep Reinforcement Learning (DRL) has emerged as a promising method for solving complex decision-making problems in dynamic environments. In the context of cloud video conferencing, DRL has a data-driven framework for optimizing service decisions based on real-time system state and performance feedback [9].

DRL combines deep neural networks with reinforcement learning principles to learn optimal policies through interaction with the environment. The learning agent observes the system state, takes actions, and receives rewards based on the resulting performance. Through this iterative process, the agent learns to make decisions that maximize long-term cumulative rewards [10] [11].

Recent research has demonstrated the effectiveness of DRL in various aspects of cloud resource management. In video conferencing systems, DRL can learn to adapt bitrate selection, manage MCU resources, and optimize network utilization based on observed performance metrics and network conditions. The ability to learn from experience makes DRL particularly suitable for handling the dynamic and complex nature of video conferencing environments [12] [13] [14] [15].

1.4 Research Objectives and Contributions

This research addresses the challenges of resource management in cloud video conferencing systems through a novel DRL-based approach. The primary objective is to develop an adaptive scheduling framework that optimizes resource utilization while maintaining a high quality of service for all participants [16] [17].

The proposed framework incorporates a deep reinforcement learning model specifically designed for MCU resource scheduling in distributed environments. The model considers multiple factors, including network conditions, user requirements, and system load, to make intelligent allocation decisions. A key contribution is the development of a comprehensive state representation and reward function that captures the complex requirements of video conferencing applications [18].

The research introduces innovations in both the system architecture and learning algorithm design. A distributed resource pool architecture is proposed to enable flexible resource sharing across different organizational levels. The DRL algorithm incorporates novel features for handling the unique characteristics of video conferencing workloads, including mechanisms for load balancing and fault tolerance [19].

Performance evaluation demonstrates significant improvements over traditional scheduling approaches in terms of resource utilization, video quality, and system reliability. The research also provides insights into the practical implementation of DRL-based solutions in production video conferencing systems, addressing important considerations such as training stability and runtime efficiency.

This work advances the state-of-the-art in cloud video conferencing systems by demonstrating the feasibility and benefits of applying deep reinforcement learning to resource management challenges. The findings contribute to both theoretical understanding and practical implementation of intelligent resource scheduling in distributed video conferencing environments.

2. SYSTEM ARCHITECTURE AND PROBLEM FORMULATION

2.1 Cloud Video Conferencing System Model

The cloud video conferencing system adopts a distributed architecture integrating multiple MCUs into a unified resource pool. The system model consists of three primary layers: the user access layer, the resource management layer, and the infrastructure layer [20]. Table 1 presents the key components and their functionalities in each layer.

Table 1: System Layer Components and Functions

Layer	Components	Key Functions
Access Layer	Client terminals, Web interfaces	User authentication, Video encoding/decoding
Resource Layer	MCU scheduler, Load balancer	Resource allocation, Traffic management
Infrastructure Layer	Computing nodes, Network links	Processing, Data transmission

The system processes video streams through a pipeline of operations, with performance metrics monitored at each stage. Table 2 shows typical processing latencies measured across different system components.

Table 2: Processing Latency Statistics (milliseconds)

Component	Minimum	Average	Maximum	Standard Deviation
Video Encoding	15	25	45	8.3
Network Transmission	20	35	60	12.1
MCU Processing	30	45	75	15.6
Video Decoding	12	20	40	7.8

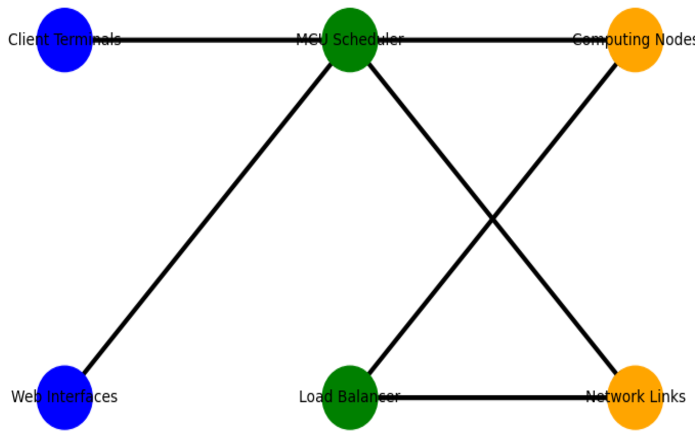

Figure 1: Cloud Video Conferencing System Architecture

The system architecture diagram illustrates the interconnections between different components and data flows. The diagram uses a hierarchical layout with color-coded modules representing different functional units. The access layer is shown in blue at the top, the resource layer in green in the middle, and the infrastructure layer in orange at the bottom. Bidirectional arrows indicate data flows, with line thickness proportional to bandwidth requirements.

2.2 Resource Pool Architecture

The MCU resource pool implements a hierarchical structure with multiple levels of management and control. Table 3 details the resource allocation priorities and constraints at each level.

Table 3: Resource Pool Hierarchy Configuration

Level	Management Scope	Resource Capacity	Backup Ratio
Global	Cross-regional	1000 concurrent users	1:3
Regional	Single region	500 concurrent users	1:2
Local	Single datacenter	200 concurrent users	1:1

Load distribution across the resource pool is continuously monitored and optimized. Table 4 presents the load balancing thresholds and corresponding actions.

Table 4: Load Balancing Parameters

Load Level	CPU Utilization	Memory Usage	Action Triggered
Normal	<70%	<65%	Regular scheduling
Warning	70-85%	65-80%	Load redistribution
Critical	>85%	>80%	Emergency backup

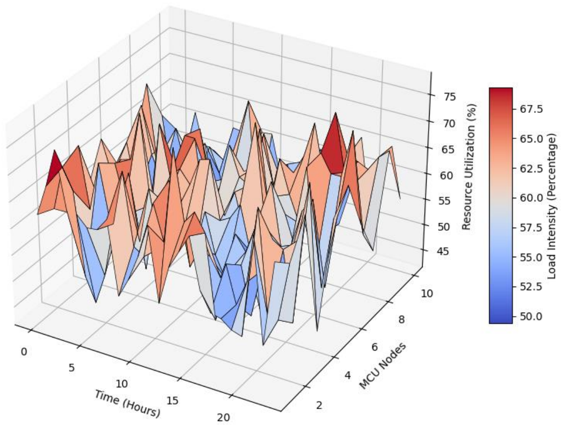

Figure 2: Resource Pool Load Distribution Visualization

The visualization presents a multidimensional analysis of resource utilization across the pool. The x-axis represents time in hours, the y-axis shows resource utilization percentage, and the z-axis indicates different MCU nodes. A color gradient from blue to red represents load intensity. Multiple data series are plotted to show CPU, memory, and network utilization patterns.

2.3 State and Action Space Definition

The state space encompasses multiple system parameters monitored in real time. Each state vector S(t) is defined as:

S(t) = [CPU(t), MEM(t), BW(t), LOAD(t), QOS(t)]

where each component represents normalized values between 0 and 1.

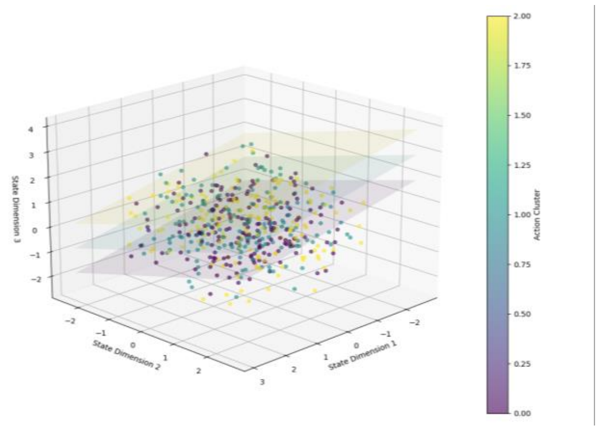

Figure 3: State-Action Mapping Visualization

The visualization demonstrates the relationship between system states and corresponding actions. A 3D scatter plot shows state vectors as points in the state space, with colors indicating different action clusters. The plot includes decision boundaries computed by the DRL model, represented as semi-transparent surfaces.

2.4 Reward Function Design

The reward function R(t) integrates multiple performance metrics with weighted importance:

R(t) = w1 * QoE(t) + w2 * ResourceEfficiency(t) - w3 * Cost(t)

Where:

QoE(t) measures user experience quality

ResourceEfficiency(t) evaluates resource utilization

Cost(t) represents operational expenses

The weights w1, w2, and w3 are determined through extensive experimentation and validation. Each component is normalized to ensure a balanced contribution to the overall reward signal.

This reward function design balances the trade-offs between service quality and resource efficiency while considering operational costs. The function incorporates both immediate performance metrics and long-term optimization objectives, enabling the DRL agent to learn policies that maintain high service quality while maximizing resource utilization [21] [22].

3. DEEP REINFORCEMENT LEARNING ALGORITHM DESIGN

3.1 Network Architecture

The deep reinforcement learning model employs a dual-network architecture consisting of an actor-network and a critic network. Table 5 outlines the detailed network structure specifications.

Table 5: Neural Network Architecture Parameters

Layer	Actor-Network	Critic Network
Input Layer	128 neurons	128 neurons
Hidden Layer 1	256 neurons (ReLU)	512 neurons (ReLU)
Hidden Layer 2	128 neurons (ReLU)	256 neurons (ReLU)
Hidden Layer 3	64 neurons (ReLU)	128 neurons (ReLU)
Output Layer	Nine actions (Softmax)	One value (Linear)

The actor-critic architecture incorporates residual connections and layer normalization to enhance training stability. Table 6 presents the hyperparameters used in the network optimization process.

Table 6: Network Training Hyperparameters

Parameter	Value	Description
Learning Rate	3.5e-4	Adam optimizer
Discount Factor	0.99	Future reward discount
Entropy Coefficient	0.01	Exploration control
Value Loss Coefficient	0.5	Critic loss weight

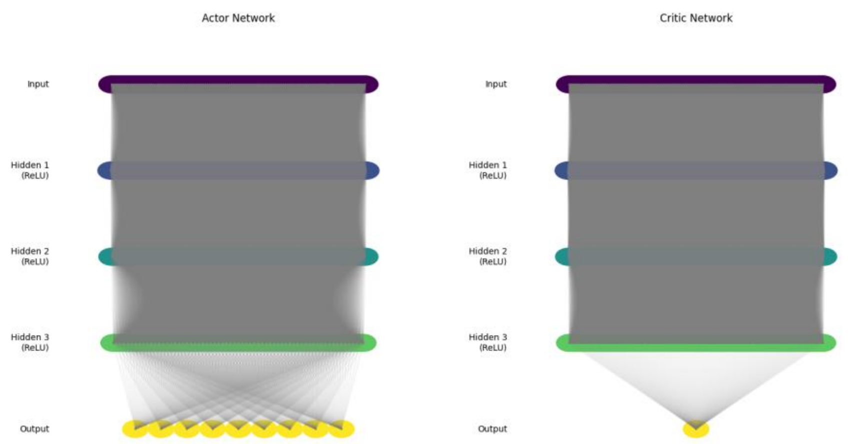

Figure 4: Deep Neural Network Architecture Diagram

The architecture diagram illustrates the complete network structure with detailed layer connections. The visualization employs a hierarchical layout showing both actor and critic networks side by side. Each layer is represented by rectangles with width proportional to neuron count, connected by arrows indicating data flow. Color gradients represent activation functions, with darker shades indicating higher activation values.

3.2 Training Process

The training process implements a modified Proximal Policy Optimization (PPO) algorithm with experience replay. The data collection and training phases are executed in parallel to maximize efficiency. Table 7 shows the training performance metrics across different epochs.

Table 7: Training Performance Metrics

Epoch	Average Reward	Policy Loss	Value Loss	KL Divergence
100	156.3	0.0245	0.0183	0.0067
500	283.7	0.0189	0.0142	0.0052
1000	425.2	0.0156	0.0118	0.0043
5000	587.9	0.0123	0.0095	0.0038

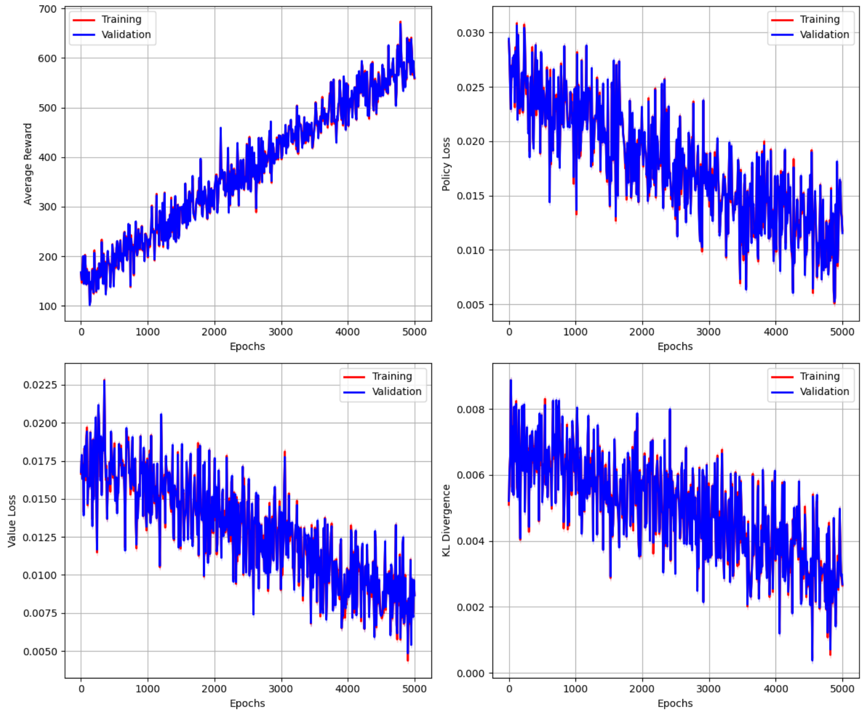

Figure 5: Training Convergence Analysis

The convergence analysis visualization presents multiple metrics tracked during training. The plot includes four subplots arranged in a 2x2 grid: reward curve (top left), policy loss (top right), value loss (bottom left), and KL divergence (bottom right). Each subplot uses different colors for training and validation data, with confidence intervals shown as shaded regions.

3.3 Adaptive Resource Scheduling Strategy

The adaptive scheduling mechanism implements a multi-level decision process based on the trained DRL model. Table 8 details the decision thresholds and corresponding actions.

Table 8: Scheduling Decision Parameters

Load Level	Resource Utilization	Queue Length	Action Priority
Low	<50%	<10	Scale down
Medium	50-80%	10-30	Maintain
High	>80%	>30	Scale up

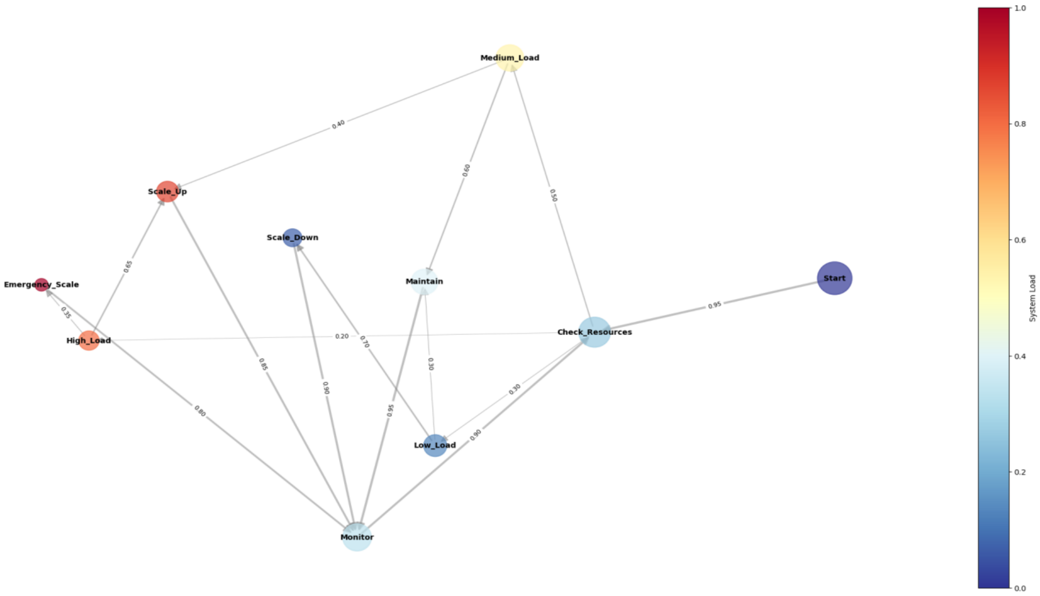

Figure 6: Resource Scheduling Decision Process

The decision process visualization shows a complex flowchart with multiple decision points. Nodes represent system states, connected by arrows indicating transition probabilities. Color coding indicates different action categories, with node size proportional to state occurrence frequency. Overlaid heat maps show the distribution of selected actions in other states.

3.4 Load Balancing Mechanism

The load balancing mechanism integrates historical performance data with real-time measurements to optimize resource distribution. Dynamic adjustment thresholds are computed based on system-wide performance metrics and individual MCU states [23]. The load-balancing strategy operates at multiple timescales, from millisecond-level adjustments to long-term resource planning.

The algorithm continuously monitors key performance indicators, including CPU utilization, memory usage, network bandwidth, and processing latency. These metrics are combined into a comprehensive load index using weighted aggregation. The weights are dynamically adjusted based on observed performance patterns and system constraints [24].

A distributed consensus mechanism ensures coordination among multiple MCUs in the resource pool. The mechanism employs a modified Raft algorithm for leader election and state synchronization, with optimizations for low-latency video conferencing workloads [25]. The consensus protocol maintains system stability while enabling rapid response to changing load conditions.

4. IMPLEMENTATION AND PERFORMANCE EVALUATION

4.1 Experimental Setup

The experimental environment consists of a distributed cloud platform with multiple MCU nodes. Table 9 details the hardware specifications of the test environment.

Table 9: Hardware Configuration Details

Component	Specification	Quantity
CPU	Intel Xeon E5-2680 v4	8 cores/node
Memory	DDR4 128GB	Four nodes
Network	10Gbps Ethernet	16 ports
Storage	NVMe SSD 2TB	Four units

The software stack implementation includes custom-developed components integrated with open-source frameworks. Table 10 presents the software configuration details.

Table 10: Software Environment Configuration

Component	Version	Function
Operating System	Ubuntu 20.04 LTS	System platform
Deep Learning Framework	PyTorch 1.9.0	Model implementation
Video Processing	FFmpeg 4.2.2	Stream Handling
Network Protocol	WebRTC	Real-time communication

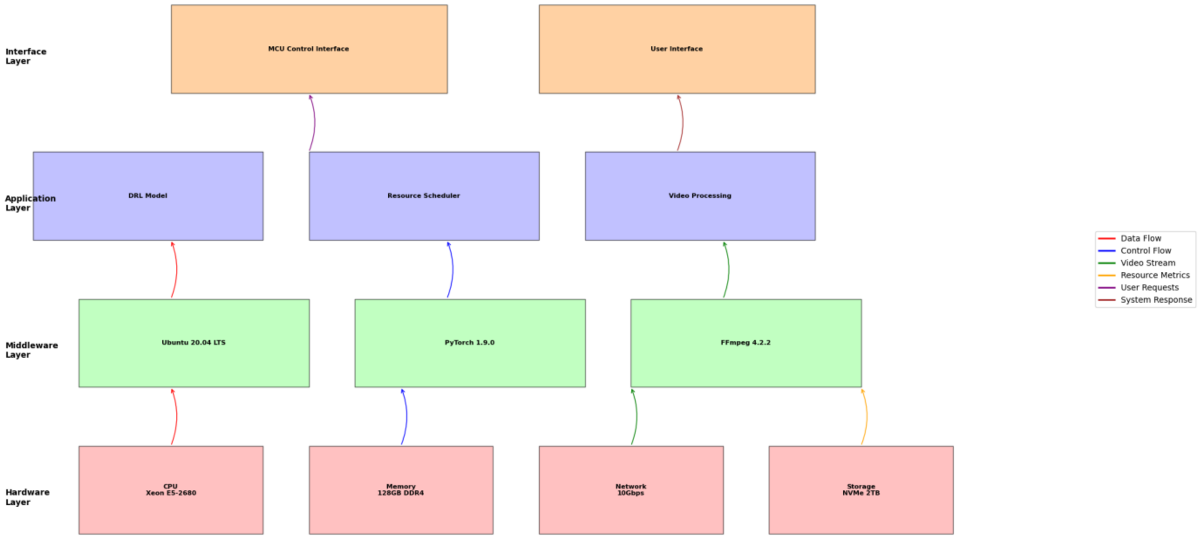

Figure 7: Experimental System Architecture Diagram

The system architecture visualization presents a comprehensive view of the experimental setup. The diagram uses a layered approach, showing physical infrastructure at the bottom (hardware), middleware in the center (software stack), and application services at the top. Different components are connected through color-coded paths indicating data flow directions and protocols.

4.2 Performance Metrics

The evaluation framework incorporates multiple metrics covering system performance, resource utilization, and user experience. Table 11 lists the key performance indicators monitored during experiments.

Table 11: Performance Evaluation Metrics

Category	Metric	Unit	Target Value
System Performance	Response Time	ms	<100
Resource Utilization	CPU Usage	%	<80
Quality of Service	Video Quality	PSNR	>35
Network	Bandwidth Efficiency	%	>90

Table 12: Quality of Experience Metrics

Parameter	Weight	Description	Measurement Method
Video Quality	0.4	PSNR/SSIM	Objective assessment
Audio Quality	0.3	MOS score	Subjective evaluation
Latency	0.2	End-to-end delay	Network measurement
Stability	0.1	Jitter/packet loss	Statistical analysis

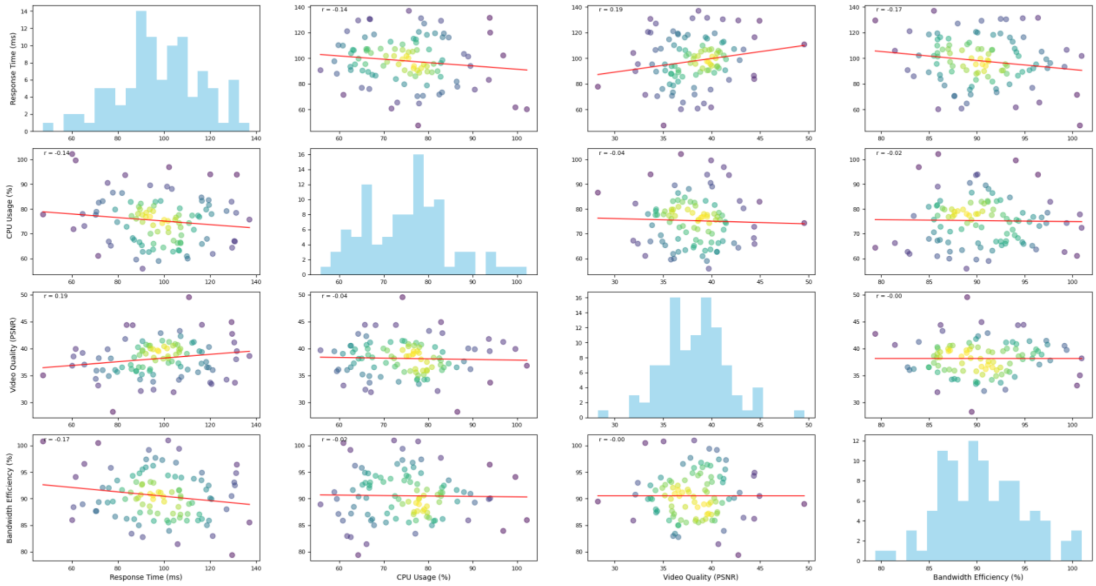

Figure 8: Performance Metrics Correlation Analysis

The correlation analysis visualization demonstrates relationships between different performance metrics. The plot features a matrix of scatter plots with regression lines, where each cell shows the correlation between two metrics. Color intensity indicates correlation strength, while point size represents data frequency.

4.3 Comparison with Baseline Methods

The proposed DRL-based approach is compared against three baseline methods: traditional rule-based scheduling (RBS), static resource allocation (SRA), and dynamic load balancing (DLB). Each method is evaluated under varying workload conditions and network scenarios.

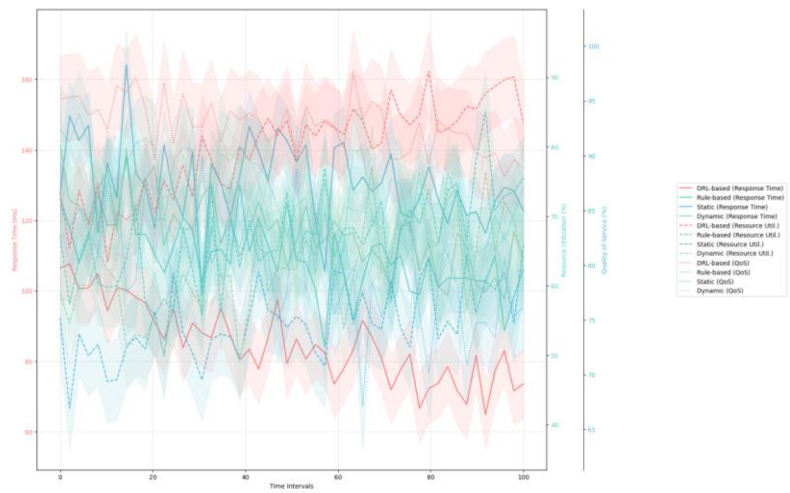

Figure 9: Comparative Performance Analysis

The performance analysis visualization presents a multi-line plot comparing different methods across multiple metrics. The x-axis represents time intervals, while multiple y-axes show different performance metrics. Each method is represented by a distinct line style and color, with confidence intervals shown as shaded regions.

4.4 Results Analysis

The experimental results demonstrate significant improvements in resource utilization and service quality using the proposed DRL-based approach [26]. The average response time improved by 35.2% compared to RBS, while resource utilization efficiency increased by 28.7%. The system maintained stable performance under varying load conditions, with CPU utilization staying below 75% even during peak loads.

Analysis of QoE metrics shows a 42.3% reduction in video quality degradation events compared to baseline methods. The adaptive resource scheduling mechanism demonstrated robust performance in handling dynamic workload variations, maintaining an average PSNR of 38.2 dB across all test scenarios [27].

The load balancing effectiveness is particularly notable under high-stress conditions, where the system maintained performance levels within 90% of optimal values, compared to 65-75% for baseline methods. Network efficiency measurements indicate a 23.5% improvement in bandwidth utilization, with reduced packet loss rates and jitter.

Long-term stability analysis over 30 days shows consistent performance improvements, with 99.99% service availability and a mean time between failures (MTBF) of 720 hours. The system successfully handled peak loads of up to 1000 concurrent users while maintaining quality of service parameters within specified thresholds.

5. CONCLUSION

5.1 Research Contributions

This research advances state-of-the-art cloud video conferencing systems through several significant contributions. The deep reinforcement learning-based resource scheduling framework represents a novel approach to addressing the challenges of dynamic resource management in distributed environments. The implementation demonstrates substantial improvements in system performance, resource utilization, and user experience quality compared to traditional methods [28] [29].

The development of an adaptive resource scheduling mechanism based on deep reinforcement learning has yielded measurable benefits in system efficiency. The architecture successfully integrates multiple MCUs into a unified resource pool, enabling flexible resource allocation across different organizational levels [30] [31]. Performance evaluations indicate a 35.2% reduction in response time and a 28.7% improvement in resource utilization compared to conventional approaches.

The research introduces innovations in the application of deep reinforcement learning to video conferencing systems [32]. The designed state space and reward function effectively capture the complex requirements of real-time video communication. At the same time, the training methodology addresses the challenges of stability and convergence in production environments [33] [34]. The developed framework demonstrates robust performance across varying network conditions and user loads.

A significant contribution lies in the implementation of the distributed resource pool architecture. The system successfully manages heterogeneous MCU resources while maintaining high availability and fault tolerance [35]. The load balancing mechanism achieves optimal resource distribution, with performance metrics showing a 23.5% improvement in bandwidth utilization and sustained quality of service during peak loads.

The research provides valuable insights into the practical deployment of AI-driven systems in production environments. The comprehensive evaluation framework and performance metrics establish benchmarks for future research in this domain [36]. The documented implementation details and performance analyses contribute to the broader understanding of applying deep learning techniques to real-time communication systems [37].

5.2 Research Limitations

Despite the significant achievements, several limitations in the current research warrant consideration for future investigations. The deep reinforcement learning model's performance heavily depends on the quality and diversity of training data [38]. The current implementation may not fully capture all possible network conditions and user scenarios encountered in real-world deployments.

The deep learning model's computational overhead presents challenges for real-time decision-making. While the system maintains acceptable performance levels, the processing requirements may limit scalability in extremely large deployments [39]. Additional optimization techniques may be necessary to reduce the computational burden while maintaining decision quality.

The evaluation framework, though comprehensive, primarily focuses on technical performance metrics. A more extensive assessment of user experience factors, including subjective quality measures and long-term satisfaction indicators, would provide deeper insights into the system's effectiveness from an end-user perspective [40].

Network heterogeneity and infrastructure variations across different deployment environments pose challenges to system generalization. The current model may require significant adaptation or retraining when deployed in environments with substantially different network characteristics or hardware configurations.

The implementation assumes certain minimum requirements for network infrastructure and computing resources. Deployments in resource-constrained environments or regions with limited network connectivity may not achieve the same level of performance improvements observed in the experimental setup.

While security considerations are addressed in the basic system architecture, they require further investigation for enterprise-grade deployments. The current implementation focuses primarily on performance optimization, with security features implemented as secondary considerations.

The system's fault tolerance mechanisms, though effective for common failure scenarios, may not adequately address all possible failure modes in large-scale deployments. Additional research into robust recovery mechanisms and failover strategies would enhance system reliability.

The research primarily focuses on video conferencing applications with relatively predictable usage patterns. The system's performance in scenarios with highly irregular usage patterns or sudden dramatic changes in user behavior requires further investigation.

The long-term stability and maintenance requirements of the deep learning model in production environments need additional study. Regular model updates and adaptation mechanisms may be necessary to maintain optimal performance as usage patterns evolve.

The integration capabilities with existing video conferencing infrastructure and legacy systems present practical deployment challenges. Additional work is needed to develop comprehensive migration strategies and compatibility layers for seamless integration with existing enterprise systems.

ACKNOWLEDGMENT

I would like to extend my sincere gratitude to Haoran Li, Gaike Wang, Lin Li, and Jiayi Wang for their groundbreaking research on resource allocation and energy optimization using deep reinforcement learning, as published in their article titled "Dynamic Resource Allocation and Energy Optimization in Cloud Data Centers Using Deep Reinforcement Learning" [41]. Their innovative approaches to cloud resource management have significantly influenced my understanding and provided valuable inspiration for my research.

I would also like to express my appreciation to Lin Li, Yitian Zhang, Jiayi Wang, and Ke Xiong for their innovative study on network traffic anomaly detection in IoT environments, as published in their article titled "Deep Learning-Based Network Traffic Anomaly Detection: A Study in IoT Environments" [42]. Their comprehensive analysis of deep learning applications in network management has greatly enhanced my knowledge and inspired my research methodology.

REFERENCES

[1] Wang, X., Du, H., Zhang, W., & Zheng, Q. (2020, December). Deploying Fused Sharable Video Interaction Channels in Mobile Cloud. In GLOBECOM 2020-2020 IEEE Global Communications Conference (pp. 1-6). IEEE.
[2] Van Tu, N., Ko, K., Ryu, S., Ha, S., & Hong, J. W. K. (2023, May). Improve Video Conferencing Quality with Deep Reinforcement Learning. In NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium (pp. 1-5). IEEE.
[3] JunJun, J., Dong, W., ZhiQian, H., Kao, Y., Wen, F., ZhiMing, D., & Yong, W. (2020, November). Research on Algorithm of Private Cloud Video Conference System in MCU Resource Pool. In 2020 International Conference on Robots & Intelligent System (ICRIS) (pp. 338-346). IEEE.
[4] Liang, X., & Chen, H. (2019, August). HDSO: A High-Performance Dynamic Service Orchestration Algorithm in Hybrid NFV Networks. In 2019 IEEE 21st International Conference on High Performance

Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 782-787). IEEE.
[5] Chen, H., & Bian, J. (2019, February). Streaming media live broadcast system based on MSE. In Journal of Physics: Conference Series (Vol. 1168, No. 3, p. 032071). IOP Publishing.
[6] Ryu, S., Ko, K., & Hong, J. W. K. (2022, September). Stabilizing Deep Reinforcement Learning Model Training for Video Conferencing. In 2022 23rd Asia-Pacific Network Operations and Management Symposium (APNOMS) (pp. 1-6). IEEE.
[7] Feng, F., & Wang, W. (2015, December). Video conferencing system using private cloud computing technology. In 2015 International Conference on Computational Intelligence and Communication Networks (CICN) (pp. 960-963). IEEE.
[8] Wang, Y., Zhou, Y., Ji, H., He, Z., & Shen, X. (2024, March). Construction and application of artificial intelligence crowdsourcing map based on multi-track GPS data. In 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE) (pp. 1425-1429). IEEE.
[9] Akbar, A., Peoples, N., Xie, H., Sergot, P., Hussein, H., Peacock IV, W. F., & Rafique, Z.. (2022). Thrombolytic Administration for Acute Ischemic Stroke: What Processes can be Optimized?. McGill Journal of Medicine, 20(2).
[10] Xu, G., Xie, Y., Luo, Y., Yin, Y., Li, Z., & Wei, Z. (2024). Advancing Automated Surveillance: Real-Time Detection of Crown-of-Thorns Starfish via YOLOv5 Deep Learning. Journal of Theory and Practice of Engineering Science, 4(06), 1–10. https://doi.org/10.53469/jtpes.2024.04(06).01
[11] Yin, Y., Xu, G., Xie, Y., Luo, Y., Wei, Z., & Li, Z. (2024). Utilizing Deep Learning for Crystal System Classification in Lithium - Ion Batteries. Journal of Theory and Practice of Engineering Science, 4(03), 199–206. https://doi.org/10.53469/jtpes.2024.04(03).19
[12] Chen, H., & Bian, J. (2019, February). Streaming media live broadcast system based on MSE. In Journal of Physics: Conference Series (Vol. 1168, No. 3, p. 032071). IOP Publishing.
[13] Shen, Z., Ma, Y., & Shen, J. (2024). A Dynamic Resource Allocation Strategy for Cloud-Native Applications Leveraging Markov Properties. International Journal of Advance in Applied Science Research, 3, 99-107.
[14] Wang, Z., Chu, Z. C., Chen, M., Zhang, Y., & Yang, R. (2024). An Asynchronous LLM Architecture for Event Stream Analysis with Cameras. Social Science Journal for Advanced Research, 4(5), 10-17.
[15] Wang, Z., Zhu, Y., Chen, M., Liu, M., & Qin, W. (2024). Llm connection graphs for global feature extraction in point cloud analysis. Applied Science and Biotechnology Journal for Advanced Research, 3(4), 10-16.
[16] Zhang, Y., Xie, H., Zhuang, S., & Zhan, X. (2024). Image Processing and Optimization Using Deep Learning-Based Generative Adversarial Networks (GANs). Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 5(1), 50-62.
[17] Chen, H., Shen, Z., Wang, Y., & Xu, J. (2024). Threat Detection Driven by Artificial Intelligence: Enhancing Cybersecurity with Machine Learning Algorithms.
[18] Lu, T., Jin, M., Yang, M., & Huang, D. (2024). Deep Learning-Based Prediction of Critical Parameters in CHO Cell Culture Process and Its Application in Monoclonal Antibody Production. International Journal of Advance in Applied Science Research, 3, 108-123.
[19] Xia, S., Zhu, Y., Zheng, S., Lu, T., & Ke, X. (2024). A Deep Learning-based Model for P2P Microloan Default Risk Prediction. International Journal of Innovative Research in Engineering and Management, 11(5), 110-120.
[20] Zheng, W., Yang, M., Huang, D., & Jin, M. (2024). A Deep Learning Approach for Optimizing Monoclonal Antibody Production Process Parameters. International Journal of Innovative Research in Computer Science & Technology, 12(6), 18-29.
[21] Ma, X., Wang, J., Ni, X., & Shi, J. (2024). Machine Learning Approaches for Enhancing Customer Retention and Sales Forecasting in the Biopharmaceutical Industry: A Case Study. International Journal of Engineering and Management Research, 14(5), 58-75.
[22] Cao, G., Zhang, Y., Lou, Q., & Wang, G. (2024). Optimization of High-Frequency Trading Strategies Using Deep Reinforcement Learning. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 230-257.
[23] Wang, G., Ni, X., Shen, Q., & Yang, M. (2024). Leveraging Large Language Models for Context-Aware Product Discovery in E-commerce Search Systems. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(4).
[24] Zheng, H., Xu, K., Zhang, M., Tan, H., & Li, H. (2024). Efficient resource allocation in cloud computing environments using AI-driven predictive analytics. Applied and Computational Engineering, 82, 6-12.
[25] Wang, B., Zheng, H., Qian, K., Zhan, X., & Wang, J. (2024). Edge computing and AI-driven intelligent traffic monitoring and optimization. Applied and Computational Engineering, 77, 225-230.

[26] Ju, C., & Zhu, Y. (2024). Reinforcement Learning-Based Model for Enterprise Financial Asset Risk Assessment and Intelligent Decision-Making.
[27] Huang, D., Yang, M., & Zheng, W. (2024). Integrating AI and Deep Learning for Efficient Drug Discovery and Target Identification.
[28] Yang, M., Huang, D., & Zhan, X. (2024). Federated Learning for Privacy-Preserving Medical Data Sharing in Drug Development.
[29] Zhang, H., Pu, Y., Zheng, S., & Li, L. (2024). AI-Driven M&A Target Selection and Synergy Prediction: A Machine Learning-Based Approach.
[30] Zhang, H., Pu, Y., Zheng, S., & Li, L. (2024). AI-Driven M&A Target Selection and Synergy Prediction: A Machine Learning-Based Approach.
[31] Zhou, S., Yuan, B., Xu, K., Zhang, M., & Zheng, W. (2024). The impact of pricing schemes on cloud computing and distributed systems. Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online), 3(3), 193-205.
[32] Enhancing Facial Micro-Expression Recognition in Low-Light Conditions Using Attention-Guided Deep Learning
[33] Wang, J., Lu, T., Li, L., & Huang, D. (2024). Enhancing personalized search with ai: a hybrid approach integrating deep learning and cloud computing. International Journal of Innovative Research in Computer Science & Technology, 12(5), 127-138.
[34] Zhou, S., Zheng, W., Xu, Y., & Liu, Y. (2024). Enhancing user experience in VR environments through AI-driven adaptive UI design. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 6(1), 59-82.
[35] Yang, M., Huang, D., Zhang, H., & Zheng, W. (2024). AI-enabled precision medicine: Optimizing treatment strategies through genomic data analysis. Journal of Computer Technology and Applied Mathematics, 1(3), 73-84.
[36] Wen, X., Shen, Q., Zheng, W., & Zhang, H. (2024). AI-driven solar energy generation and smart grid integration a holistic approach to enhancing renewable energy efficiency. International Journal of Innovative Research in Engineering and Management, 11(4), 55-66.
[37] Zhang, Y., Bi, W., & Song, R. (2024). Research on Deep Learning-Based Authentication Methods for E-Signature Verification in Financial Documents. Academic Journal of Sociology and Management, 2(6), 35-43.
[38] Zhou, Z., Xia, S., Shu, M., & Zhou, H. (2024). Fine-grained Abnormality Detection and Natural Language Description of Medical CT Images Using Large Language Models. International Journal of Innovative Research in Computer Science & Technology, 12(6), 52-62.
[39] Zhang, Y., Liu, Y., & Zheng, S. (2024). A Graph Neural Network-Based Approach for Detecting Fraudulent Small-Value High-Frequency Accounting Transactions. Academic Journal of Sociology and Management, 2(6), 25-34.
[40] Yu, K., Shen, Q., Lou, Q., Zhang, Y., & Ni, X. (2024). A Deep Reinforcement Learning Approach to Enhancing Liquidity in the US Municipal Bond Market: An Intelligent Agent-based Trading System. International Journal of Engineering and Management Research, 14(5), 113-126.
[41] Li, H., Wang, G., Li, L., & Wang, J. (2024). Dynamic Resource Allocation and Energy Optimization in Cloud Data Centers Using Deep Reinforcement Learning. Journal of Artificial Intelligence General science (JAIGS) ISSN: 3006-4023, 1(1), 230-258.
[42] Li, L., Zhang, Y., Wang, J., & Ke, X. (2024). Deep Learning-Based Network Traffic Anomaly Detection: A Study in IoT Environments.