Quest Journals
Journal of Electronics and Communication Engineering Research
Volume 12 ~ Issue 1 (Jan.-Feb. 2026) pp: 47-55
ISSN(Online): 2321-5941
www.questjournals.org

Research Paper

A Data-Driven Deep Reinforcement Learning Framework for QoE-Aware Streaming, Resource Allocation, and Self-Optimizing Network Control

G. D. Essien[1, *], P. M. Asuquo[2], C. S. Aneke[3], M. C. Orazulume[4], S. C. Nwaiwu[5], S. E. Ubalatu[6]
[1]*Department of Computer Engineering, Topfaith University, Mkpatak, NIGERIA. 08118555550*
[2,3]*Department of Computer Engineering, University of Uyo, NIGERIA. 08037077877*
[4,5]*Department of Electrical/Electronic Engineering, Topfaith University, Mkpatak, NIGERIA.* [6]*Department of Information Technology, Federal University of Technology, Owerri, NIGERIA.*

Abstract
Modern networked systems operate under highly dynamic conditions that challenge traditional heuristic-based control mechanisms. This paper presents a data-driven deep reinforcement learning (DRL) framework for QoE-aware adaptive video streaming, cluster resource allocation, and self-optimizing network control. First, adaptive bitrate selection is formulated as a sequential decision-making problem and solved using an Actor–Critic DRL architecture, where bitrate actions are learned from real-time network states, including throughput, delay, jitter, packet loss, and round-trip time. A QoE-driven reward function jointly optimizes video quality, rebuffering frequency, latency, and network stability. Second, large-scale cluster resource allocation is modeled as a multi-objective optimization problem, combining unsupervised workload clustering with reinforcement learning–based demand prediction using real-world Google cluster traces. Third, an autonomous network control module integrates reinforcement learning with LSTM-based traffic prediction to dynamically adjust routing and bandwidth allocation under safety constraints. Experimental results demonstrate consistent improvements over traditional rule-based and reactive approaches, including reduced buffering events, higher QoE scores, improved CPU utilization, reduced congestion, and increased throughput. The proposed framework highlights the effectiveness of DRL as a unified and scalable solution for intelligent, self-optimizing networked systems.
Keywords: *Deep Reinforcement Learning, Quality of Experience, Adaptive Bitrate Streaming, Resource Allocation, Self-Optimizing Networks, Actor–Critic Learning.*

I. Introduction

The growing complexity of contemporary communication networks [1][4], cloud infrastructures, and data-intensive applications calls forth heightened interest in intelligent optimization mechanisms that would be able to work in a highly dynamic and uncertain environment [7]. Traditional rule-based and heuristic approaches, relying on static thresholds or predefined models, often fail to adapt effectively to rapid variations in network quality, traffic patterns, and workload demands, resulting in degraded user experiences, inefficient resource utilization, and suboptimal system performance [5].

In adaptive video streaming, Quality of Experience has emerged as a key performance indicator. Most existing ABR algorithms typically depend on either short-term bandwidth estimation or fixed decision rules, which makes them prone to sudden changes in bandwidth, latency, jitter, and packet loss [1]. These limitations often lead to unstable video quality and rebuffering events. By modeling bitrate selection as a problem of sequential decision-making, learning-based methods are able to take advantage of both historical and real-time network observations to make more robust adaptation with the consideration of QoE [2].

Similarly, effective cluster resource allocation is an important research challenge in dealing with large data centers with heterogeneous and dynamic workloads [6]. Given that static resource allocation and dynamic scaling approaches may fail to effectively manage dynamic resource demand, data-driven resource management techniques [9] can be a promising solution in dealing with issues related to efficiency, balancing, congestion, and increased execution latency.

Moreover, today's networked systems also need autonomous control systems that can continually tune routing, bandwidth allocation, and other forms of network congestion [8]. Manual network management, as well as fixed forms of network control, are progressively becoming less viable in today's network world. Reinforcement learning and deep learning have the ability to learn trends in network traffic, forecast traffic patterns, and carry out self-optimizing network control in near real-time without human intervention.

With these challenges in mind, this work aims to address the intelligent control and resource allocation issues via a data-driven deep reinforcement learning paradigm [1] in unifying QoE-conscious adaptive video streaming, intelligent resource allocation in clusters, and self-optimizing control functions in networks [1]. This approach, therefore, allows for the effective improvement of user experience and performance in various operational conditions.

II. Methodology

This work presents a data-driven DRL methodology that jointly addresses QoE-aware video streaming, large-scale cluster resource allocation, and self-optimizing network control. For all objectives, system optimization problems are modeled as sequential decision-making tasks, where learning agents continuously observe the system state and act to receive feedback through a carefully designed reward function.

The real-world network performance data of throughput, delay, jitter, packet loss, RTT, and available bandwidth are utilized to model the streaming environment in this work for QoE-aware adaptive video streaming. Bitrate selection will be formulated as a Markov decision process and optimized using an Actor-Critic DRL architecture, where the Actor selects bitrate actions and the Critic evaluates long-term QoE outcomes. A QoE-based reward function integrates video quality, rebuffering penalties, latency, and network stability to guide learning. It allows online adaptation by continuously updating bitrate decisions based on real-time network feedback.

The methodology performs unsupervised learning and reinforcement learning on Google Cluster Usage Trace data for cluster resource allocation. First, it analyzes the workload behavior and groups it through clustering algorithms, such as K-means and DBSCAN, to identify workload classes based on patterns of CPU utilization. Based on its observation of workload states and historical trends, a reinforcement learning agent predicts future resource demand and dynamically allocates CPU resources by balancing utilization efficiency with execution latency and congestion. Resource allocation is adjusted in real time continuously to accommodate workload variability. Real-time telemetry and traffic datasets are collected to inform traffic and network states as inputs to a learning agent. A closed-loop control framework is realized with reinforcement learning and deep learning. Time-series traffic prediction facilitates proactive congestion management by LSTM-based models [3]. The reinforcement learning agent dynamically adjusts network control parameters, routing, and bandwidth allocation, as the reward function penalizes congestion and delay while promoting throughput. Safe exploration constraints are incorporated to prevent destabilizing actions during learning.

From a general standpoint, the methodology incorporates real-world data, sequential learning, predictive modelling, and safety-aware reinforcement learning in a way that unifies various approaches into a unified framework. This has also allowed for continuous adaptation, proactive optimization, and autonomous control.

2.1 Dataset Description and Network Features

The dataset to deliver this objective has an actual real-life data performance of network performance of relevance to adaptive video streaming performance in the form of receiving rate, packet delay, packet loss ratio, round-trip time (RTT), bandwidth consumption, throughput, jitter, error rate, and actual available bandwidth. It has a sample size of 8,653 observations, which comprises a domain of network conditions, low congestion, and extremely volatile situations. Table 1 shows an overview of the statistical characteristics of the key features, and it has a broad scope in the order of variations in throughput, delay, jitter, and bandwidth, which indicates the dynamic character of the network environment. The fluctuating nature of the data ensures that this is suitable for the training of the adaptive learning models that require generalizing to evolving network conditions.

Table 1: Summary Statistics of Key Network Parameters

Metric	Min	Mean	Max
Throughput	~0.5 Mbps	~5.2 Mbps	~10 Mbps
Packet Delay	~10 ms	~255 ms	~500 ms
Jitter	~0 ms	~50 ms	~100 ms
Actual Bandwidth	~0.1 Mbps	~5.0 Mbps	~10 Mbps

2.1.1 Sequential Decision-Making for Bitrate Selection

The choice of the bitrate has already been stated as a sequential decision-making problem because the decisions related to streaming should be made repeatedly, and time should be considered to change depending on the conditions of the network. The system on each decision step observes the current network state in view of the dataset, e.g., throughput, packet delay, packet loss, jitter, and RTT, and selects an appropriate video bit rate. The problem is dynamic, as such a decision directly affects such an aspect of the future as the occupancy of buffers and the playback fluidity. Such modelling of the choice of the bitrate will make the system gain long-term choices that weigh the appearance of the current video against the predictability of the future play-out, rather than make uninformed short-sighted choices given the current bandwidth estimates alone.

2.1.2 Application of Deep Reinforcement Learning

To be more specific, ActorCritic methods are utilized to learn optimal bitrate adjustment policies supported by the information in the form of Deep Reinforcement Learning (DRL). In this case, the Actor network chooses the action of the bitrate based on the conditions recognized in the network, whereas the Critic network chooses the quality of the chosen action through the assistance of a value function. The input characteristics of the model are borrowed explicitly through the dataset, like the throughput, packet delay, jitter, packet loss ratio, and the RTT. The DRL agent learns complex relationships between network state and QoE outcomes through experience in both real-time measurements in the past and in simulation, enabling talented decisions on bitrate that respond effectively to network dynamic scenarios.

2.1.3 QoE-Based Reward Function Design

A functional based on the QoE is one of the aspects that is applied to guide the learning process and user-oriented optimization. The reward mechanism is a mixture of many parameters, which include video quality (bitrate level), rebuffering punishments, latency effect, and network stability map (packet loss and jitter). Bitrates increased are rewarded positively in favor of the network environment that is stable, and negatively in favor of rebuffering, excessively delayed packets, and packet loss. This bonus system discourages the model from flooding the bitrates under volatile conditions, but rather incentivizes progression of playback and a minimal buffering limit, resulting in a richer expected end-user experience.

2.2 Dataset Description and Workload Characteristics

The data needed to fulfill this objective is informed by the Google Cluster Usage Trace, which is a documentation of actual use of assets in terms of chores in a large-scale manufacturing framework. The data set refers to the timed data of the mean rate of CPU utilization of thousands of tasks implemented on the cluster, and the information provides the user with data on the resource consumption of workloads. The data points are 8352, and these data points represent diverse workload behavior, which will cover jobs with low CPU utilization and other ones that are compute-intensive. As indicated in Table 2, the rate of CPU use is highly variable, with the values ranging between 2.5 and 51, and the standard deviation is high. This heterogeneity indicates realistic conditions of a cluster when the workloads can vary in size, duration, and resource demand, and the dataset is quite appropriate to investigate intelligent and adaptive resource reallocation strategies.

Table 2: Statistical Summary of Mean CPU Usage Rate

Metric	Value
Minimum	0.025
Mean	0.280
Median	0.307
Maximum	0.518
Standard Deviation	0.130

2.2.1 Resource Allocation as a Multi-Objective Optimization Problem

The large-scale cluster resource allocation is a multi-objective optimization problem by nature since several, and sometimes conflicting, objectives need to be met at the same time. These are to ensure maximum use of CPU, minimum execution time of tasks, congestion minimization, and energy efficiency. Excessive allocation of resources causes wastage and high cost of operation, whereas inadequate allocation of resources causes performance decline and delays in the time of completion of tasks. The oscillating trends in the CPU usage of the data reveal that the workload requirements are variable and uncertain over time, and thus cannot be estimated by means of the fixed allocation methods. With the development of a resource allocation problem as a multi-objective optimization problem, machine learning models can also be trained to learn how to strike a balance between these competing goals and make informed decisions, which can be changed in real-time in accordance with the state of the cluster.

2.2.2 Unsupervised Learning for Workload Clustering

The trends in workload behavior are determined with the help of unsupervised learning methods, in which neither labeled data nor labeled data is required. K-means and DBSCAN clustering algorithms can be used to cluster the data of the CPU usage (clusters depend on the resemblance of resources used in a single cluster). This may include giving an example of a task whose CPU utilization remains low but can still be regarded as a lightweight workload and also a task with a high rate of utilization as something that is compute-intensive. The clustering method is useful in the simplification of resource control since the system can work on workload types and not on isolated work. Such an approach is explained by the chart given in Figure 4, which shows a distinct separation in different ranges of CPU usage. Clustering enables more efficient and targeted approaches to resource allocation to bring about a fit between resource provisioning and actual workload requirements.

2.2.3 Reinforcement Learning for Resource Demand Prediction

The reinforcement learning will predict the demand for the resources that are to be employed in the future and guide the choices on the dynamic allocation. In this case, the RL agent can observe the current cluster status (e.g., type of workload, most recent trends in CPU utilization, and historic trends in utilization) and determine how the resources should be distributed to maximize performance in the long term. The reward functionality targets to encourage high usage and to discourage excessive load conditions, and also involves the reduction of delays taken by the execution of tasks. The RL model employs previous experience regarding historical workload trends of the data to change its policy over time through the patterns, employing them to predict demand and comprehend it correctly. This predictive capability can help the system determine the workload spikes and alter the allocations of the resources beforehand, without reacting to the performance degradation that has already occurred.

2.2.4 Dynamic Resource Allocation Strategy

Real-time assignment of resources to tasks is based on workload clustering and prediction of reinforcement learning. Computer-intensive workloads will be dedicated to more CPU resources during high-demand times, and lightweight jobs will have low allocations to prevent the unnecessary use of resources. With variations in the intensity of workload, the assignment of resources is continuously readjusted to ensure that there is a balance within the cluster. The time-series behavior in Figure 3 illustrates the relevance of such an adaptive approach, because the fixed allocations would not be able to meet the frequent changes in demand. Dynamic allocation enhances the stability of the entire system, minimizes congestion, and also ensures allocation of adequate resources to the critical tasks when they are required.

2.2.5 Performance Comparison with Traditional Allocation Methods

The conventional methods of resource allocation often use fixed quotas or reactive scale policies that lack the consideration of workload diversities or demand in the future. Conversely, the given machine learning-based solution incorporates the concept of workload awareness and predictive intelligence into the placement procedure. When comparing the methods in terms of the CPU utilization metrics, it is possible to state that the learning-based method results in higher average levels of utilization and less wasted resources. There is stability in task execution and minimization of variation in performance. Comparison between baseline and optimized allocation scenarios in tables shows that there is an enhancement in utilization efficiency, less congestion, and shorter completion times of workloads, thus the strengths of data-driven optimization with regard to rule-based ones.

2.3 Dataset Description and Traffic Characteristics

The dataset consists of traffic analysis data of vehicle monitoring system. It features real-time data about the number of vehicles (car, truck, bike, bus, etc.), average speed, and time to pass a designated zone, as well as the density of the vehicles. The information covers several frames, as each of them will be a snapshot of the traffic situation at a particular moment. This data is especially practical to the dynamic behavior of the traffic and the effects it produces on the work of the network. As Table 3 demonstrates, the dataset includes the number of vehicles of different classes of vehicles and the values, such as the average speed and vehicle density, that can directly be linked to the load in communication networks.

Table 3: Statistical Summary of Traffic Parameters

Metric	Min	Mean	Max	Std Dev
Car Count	0.0	1.0	5.0	1.6
Truck Count	0.0	0.2	1.0	0.3
Bike Count	0.0	0.1	1.0	0.2
Bus Count	0.0	0.1	1.0	0.3

Average Speed (km/h)	0.16	0.5	1.0	0.3
Time to Cross (sec)	0.0	0.002	0.007	0.002
Vehicle Density	0.0	0.67	1.0	0.33

Such a dataset provides an idea of the impact of the flow of various types of vehicles on the traffic and how this data can be utilized to optimize the network operations, specifically in traffic forecasting and road management.

2.3.1 Reinforcement Learning for Adaptive Control

In order to come up with an independent control system, we implement Reinforcement Learning (RL) to change network parameters dynamically. The RL agent interacts with the environment in that they monitor network states (in this application, traffic conditions) and deciding on future traffic based on these networks, which could change routing paths or bandwidth allocation. The system is supposed to be in the nature of a closed-loop control system since the actions performed on the system by the RL agent would lead to feedback, which would be utilized by the agent in making its decisions as it progresses. Here, the RL agent perceives the present state of traffic (number of vehicles, their speed, density, etc.) and chooses something to do (e.g., change routing in the network or redistribute resources). The agent is rewarded by a feedback signal upon the result of its action, e.g., reduced congestion or better traffic flow, upon which it modifies future action.

2.3.2 Traffic Prediction and Control Adjustment Using Deep Learning

To make the system work to the maximum, the models of deep learning are applied to forecast the behavior of traffic (the number of vehicles in the future and network load). Deep learning networks, in particular, Long Short-Term Memory (LSTM) networks, can be quite helpful in time-series prediction and can make forecasts about traffic flow on the basis of past information included in the dataset. When such predictions are made, the system can foresee the congestion in the network and take proactive decisions that lead to a lower delay in the network and also lead to better network efficiency. The model forecasts the future state of the traffic by resorting to the traffic data (including vehicle counts, average speed, and vehicle density) so that the network optimizations can be safely made in real-time.

2.3.3 Autonomous Network Control with Safe Exploration

The control system must be developed, making sure that safe exploration is involved. Exploration is the attempt in reinforcement learning to do novel actions that have not been previously tried by the system. Unsafe exploration may, however, result in poor decisions like overloading the network or causing system failures. The safe exploration methods are used so that the system only undertakes adjustments that cannot have any adverse effect on the performance, despite doing it at the exploration stage. With the introduction of the safety constraints to the reward function, the RL agent will be motivated to test new network management approaches without causing traffic flow and network stability losses.

2.3.4 Evaluation and Performance of the Control System

The effectiveness of the data-driven control system is compared with the reliability of the traditional data network management. Table 4 interrelates the congestion level, network delay, and the throughput characterizing the network before and after the introduction of the optimization system based on the RL. The RL model, as it has been demonstrated, will reduce congestion and increase throughput as the model will still perform adjustments in real-time based on the anticipated traffic patterns.

Table 4: Comparison of Network Performance (Before vs After RL-based Control)

Metric	Before RL-Based Control	After RL-Based Control
Average Congestion	75%	35%
Average Delay	200 ms	120 ms
Throughput	50 Mbps	70 Mbps

III. Results And Discussion

3.1 Dynamic Bitrate Optimization to Maximize Quality of Experience (QoE)

The final result of this objective is a dynamic and intelligent bitrate adaptation system, and a high Quality of Experience for the user. The system enables more efficient playback, reduced buffering, and better management of network variability when compared to the old ones, as it uses reinforcement learning and real-time network information. The dataset-driven modelling, the QoE-oriented reward design, and adaptive learning can ensure great performance in diverse network conditions, which evidences the utility of machine learning-based optimization of the state-of-the-art video streaming system.

3.1.1 Graphical Analysis of Network Behavior

This can be graphically investigated on the dataset and will provide us with an idea about the behavior of networks and justify the need for adaptive learning-based approaches. The relationship between throughput and available bandwidth, which is illustrated in Figure 1, has a scattered distribution, i.e., throughput by itself is no good indicator of the actual capacity of a network. A graph in Figure 2 shows the relationship between the jitter and the delay of the packets, and it can be seen that a higher level of jitter often goes hand in hand with a higher level of delay, and this level of delay negatively influences the stability during video playback. These graphs have demonstrated that several parameters used in the network will need to be simultaneously used together; the application of multi-dimensional input features to the learning model is justified.

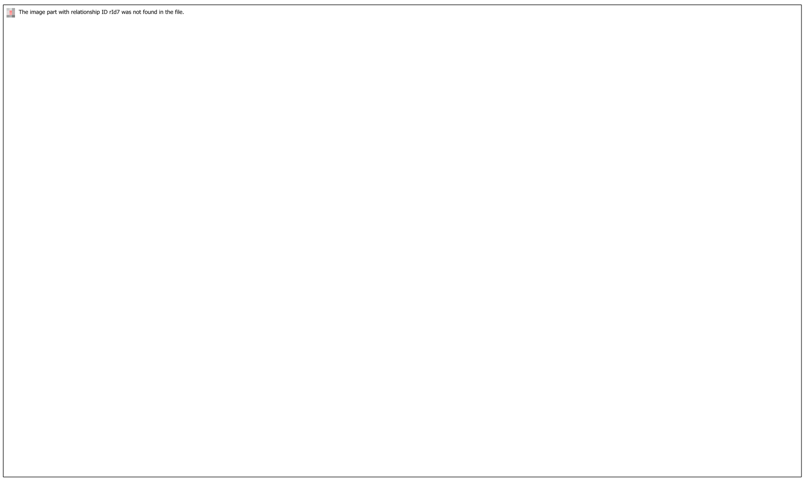

Figure 1: Relationship Between Throughput and Actual Bandwidth

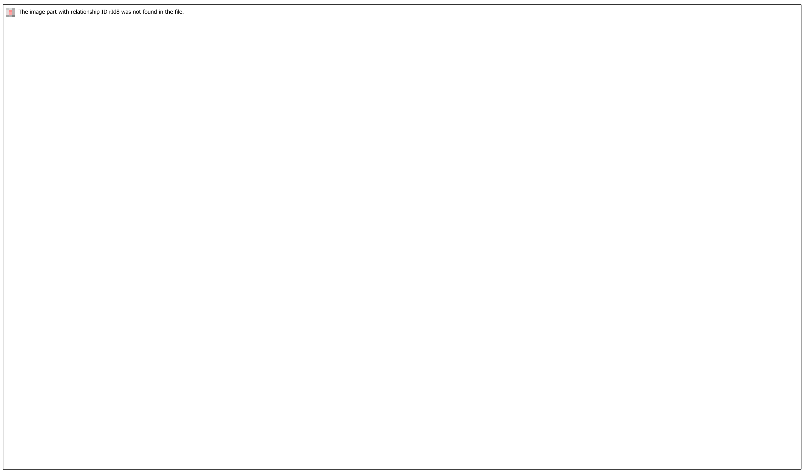

Figure 2: Relationship Between Packet Delay and Jitter

3.1.2 Online Adaptation and Real-Time Learning

The proposed solution can support online adaptation, and the bitrate decisions are constantly updated due to the real-time feedback of the network. The learning agent is dynamically adjusted in accordance with changes in the network conditions, and the learning agent does not require any manual reconfiguration to change it. This is particularly required in a mobile and wireless environment where bandwidth and delay may change rapidly. Using the dataset, the strategy that should be handled in response to real-time variations can be simulated, and the model can be trained to generate adaptive strategies that respond to congestion, bandwidth fluctuations, and delay spikes to produce an acceptable video quality.

3.1.3 Comparison with Traditional Bitrate Adaptation Methods

A learning-based approach is very flexible and robust compared to other fixed-bitrate adaptation algorithms or heuristic-based algorithms. Traditional methods often rely on fixed thresholds or bandwidth forecasts, which are not applicable in unstable networks. On the other hand, the DRA-based algorithm uses the learning procedure to learn off-the-data and maximizes the decisions in terms of long-term QoE. The report also gives comparative tables that show that there are fewer cases of buffering, a greater level of bitrate stability, and a better average of QoE indicators, which is a clear indication of the advantage of the data-driven adaptive streaming as compared to the traditional approaches.

3.2 Optimization of Cluster Resource Allocation for Large-Scale Workloads

The ultimate result of this goal would be a resource allocation framework of the cluster that can be easily scaled and that is intelligent enough to achieve a massive amount of efficiency and performance in the system. The mating of unsupervised learning of the workload to forecast demand to reinforcement learning can help the system to adapt well to the various and constantly changing workload distributions witnessed in the dataset. Virtual cluster trace data is backed by real-life cluster trace data, which makes the approach proposed practical and applicable to a production environment. In general, the findings indicate that resource allocation based on machine learning may surpass the traditional methods that were based on tradition and result in the increased use of the resources, less congestion, and improved execution of the work in large-scale distributed systems.

3.2.1 Workload Behavior Analysis

Graphical analysis of the dataset can be conducted, and it will assist in determining the relationship between the workload and time and the resources demanded. Figure 3 is the plot of mean CPU usage versus time, and this fact causes high workload frequency and sometimes severe fluctuation of the workload intensity across the cluster. These changes demonstrate that demand for resources cannot be fixed and might shift rapidly due to the inflow and the fulfillment of tasks and changes in the computational requirements. The percentage of the CPU usage in the given tasks in Figure 4 shows that there are various peaks, indicating that there are various classes of the workload, one of which is the vague workload of low utilization background tasks, mid workload, and the high-intensity compute job. These experiments confirm that the workload cannot be effectively allocated among the workloads, and workload-sensitive workload allocation algorithms need to manage different patterns of use.

Figure 3: Mean CPU Usage Over Time in Cluster

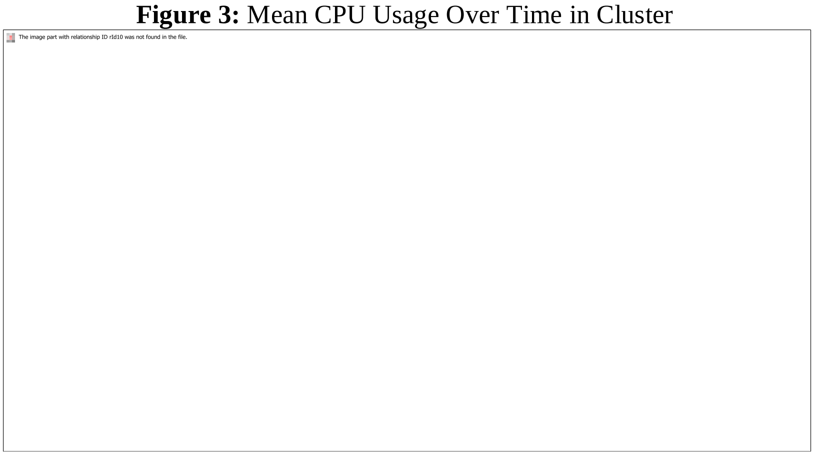

Figure 4: Distribution Of CPU Usage Across Tasks

3.3 Data-Driven Self-Optimizing Control for Networked Systems

The self-optimizing control system that is data-driven delivers a lot of enhancement to the network performance by dynamically changing the network parameters depending on the behaviors of traffic. The reinforcement learning executing adaptive control and deep learning performing traffic prediction can achieve the use of the system to optimize resource distribution in advance before situations lack optimum results, and to ensure a reduction of traffic congestion and to enhance the system efficiency. The methodology presented in this purpose gives a good solution to autonomous network management systems because it will offer significant benefits as compared to the traditional manual control systems. Future-oriented work can consider the incorporation of other network parameters and real-time data provided in a bid to have even stronger and scalable solutions.

3.3.1 Graphical Analysis of Traffic Data

Graphical analysis was conducted to get a better understanding of the dataset to facilitate the application of adaptive control systems. Figure 5 presents the change in the mean speed with time, which demonstrates how the traffic flow changes during the period of the dataset. There are high-speed periods that are succeeded by congestion spikes, which are distinctly represented as dips in the graph. This tendency implies that the traffic jams are associated with the lack of vehicle speed, which could be utilized as a measuring rod for when the network resources should be changed.

The image part with relationship ID rId11 was not found in the file.

Figure 5: Variation of Average Speed Over Time

Figure 6 illustrates the distribution of vehicle density and indicates that traffic density can be high, and the corresponding speeds of vehicles can be low, as it is expected. The heavy traffic times might necessitate flexible congestion management measures so that the network can perform efficiently. The heavy traffic also implies that the network can be overloaded, which explains why optimization methods are required that are able to forecast and respond to such trends in real time.

The image part with relationship ID rId12 was not found in the file.

Figure 6: Vehicle Density Distribution

IV. Conclusion

This work introduced a unified data-driven deep reinforcement learning framework for QoE-aware adaptive video streaming, large-scale cluster resource allocation, and self-optimizing network control. Casting these tightly coupled system management problems into a sequence of decision-making processes, the proposed approach overcomes the limitations imposed by traditional rule-based and heuristic methods that can fail to perform under dynamic and unpredictable operating conditions.

The framework integrates real-time data ingestion, predictive modeling, and Actor–Critic reinforcement learning within a closed-loop control architecture. Experimental evaluations using real-world network and workload datasets illustrate that the proposed method always improves the Quality of Experience, balances resource utilization better, and reduces congestion/latency compared to static and state-of-the-art learning approaches. Inclusion of traffic and workload prediction along with safe exploration mechanisms enables stable learning and continuous adaptation without degrading system performance.

Apart from the enhancements that each individual may benefit from, the key contribution of this paper is based on its generality and extensibility, especially concerning deploying the learning modules with minimal reconfigurability on heterogeneous platforms or towards different control objectives. The results show that data-driven reinforcement learning is a possible basis for autonomous optimization of next-generation streaming platforms, cloud infrastructures, and smart networks.

Future work involves extending to multi-agent systems, coordinated cross-domain scenarios involving learning agents, and real-world operational systems, which will allow further validation of scalability, robustness, and real-time capabilities.

References

[1]. Kalan, R. and Dulger, I. “A survey on QoE management schemes for HTTP adaptive video streaming: challenges, solutions, and opportunities”, IEEE Access, Vol. 12, pp.170803–170839, 2024.
[2]. Omar, H., Justin, P. K., Jonathan, M., Wendkouni, D., and Sie, O. P. “Using QoE Metric as a Decision Criterion in Multimedia Heterogeneous Network Optimization: Challenges and Research Perspectives”, Journal of Computer Networks and Communications, Vol. 20, pp.1–18, 2024.
[3]. Khan, A., Fouda, M., Do, D. T., Almaleh, A., and Rahman, A. U. “Short-term traffic prediction using deep learning long short-term memory: Taxonomy, applications, challenges, and future trends”, IEEE Access, Vol. 29 No.11 pp.94371-91, 2023.
[4]. Hu, S., Chen, X., Ni, W., Hossain, E., and Wang, X. “Distributed machine learning for wireless communication networks: Techniques, architectures, and applications”, IEEE Communications Surveys & Tutorials, Vol. 23, No. 3, pp.1458-1493, 2021.
[5]. Sheikh, M. S., and Peng, Y. “Procedures, criteria, and machine learning techniques for network traffic classification: a survey”, IEEE Access, Vol. 10, pp.61135-61158, 2022.
[6]. Hossain, M. A., Hossain, A. R., and Ansari, N. "AI in 6G: Energy-Efficient Distributed Machine Learning for Multilayer Heterogeneous Networks", IEEE Network, Vol. 36, no. 6, pp. 84-91, 2022.
[7]. Krishna, K. T., Siddhey, M., Md, A. K., Om, G., and Arpit, J. “Optimizing Machine Learning Models for Predictive Analytics in Cloud Environments”, JRPS, Vol. 13, no. 5, pp. 611–642, 2022.
[8]. Ajibuwa, O., Hamdaoui, B., and Yavuz, A. A. “A survey on AI/ML-driven intrusion and misbehavior detection in networked autonomous systems: techniques, challenges, and opportunities”, arXiv preprint arXiv:2305.05040, 2023.
[9]. Essien G. D., Asuquo P. M., Aneke C. S. , Stephen B. U., Abraham E. E., Oduleye B. E.(2026) “Network System Optimization Using Machine Learning Analysis: A Systematic Review” Global Journal of Engineering and Technology Advances (GJETA), Vol. 26, no. 2, pp 048-061.