ORAN-Based Intelligent Network Optimization

Ram Santhosh Patnaik Belagam
(IEC2021019)

Supervised by
Dr. Suneel Yadav

Department of Electronics and Communication Engineering
Indian Institute of Information Technology, Allahabad

May 9, 2025

Table of Contents I

Table of Contents II

Introduction

5G and RAN: The Changing Landscape

- Unprecedented growth in connected devices and data traffic.
- **5G**: Ultra-reliable, low latency, massive connectivity, diverse services.
- Legacy RAN: Proprietary, inflexible, slow to innovate.
- Need for open, flexible, and intelligent network architectures.

Limitations of Traditional RAN

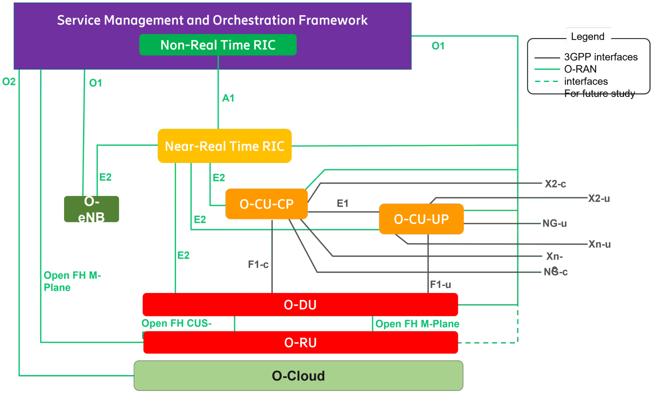
- Closed, single-vendor ecosystems.
- Hard-coded algorithms and static resource allocation.
- Difficult to scale and adapt for new 5G use cases.
- Bottleneck for 5G's full potential.

Open RAN

What is Open RAN?

- **Open RAN (ORAN)** is a new approach to building mobile networks.
- Emphasizes open, standardized interfaces and disaggregated hardware/software.
- Enables multi-vendor deployments, accelerates innovation, and increases flexibility.

Open RAN Architecture



Source: Adapted from [1]

Literature Survey

Open RAN: Literature Survey

Title	Authors	Year	Journal
Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges [2]	Polese et al.	2023	IEEE Comm. Surveys & Tutorials
The economic benefits of open RAN technology [3]	Fetterolf, Peter	2021	ACG/Dell Technologies (Online)
Intelligence and learning in O-RAN for data-driven NextG cellular networks [4]	Bonati et al.	2021	IEEE Comm. Magazine
Resource allocation in an open RAN system using network slicing [5]	Motalleb et al.	2022	IEEE Trans. Netw. Serv. Mgmt.

Problem Statement

Project Goal

- Design and assess an **ML-driven optimization** for ORAN.
- Focus: Combining **data-driven modeling** with **reinforcement learning (RL)**.
- Target functions: **Resource allocation** and **network slicing**.
- Use real-world data for simulation and RL training.

Methodology

Methodology Overview

- A modular pipeline for intelligent network optimization in an Open RAN (ORAN) framework.
- Integrates data-driven network modeling and deep reinforcement learning (RL) for closed-loop control.
- Five modules: Data Acquisition & Processing, Surrogate Environment Model, RL Agent Design, Baseline Evaluation, and Result Visualization.

Dataset Acquisition & Preprocessing

- **Dataset:** O-RAN COMMAG dataset.
- **Samples:** Over 6.9 million snapshots from a 4-gNodeB Colosseum testbed simulating eMBB, URLLC, and MTC slices.
- **Inputs:** slice_id, num_ues, dl_buffer, sum_requested_prbs, ul_rssi, ul_sinr, dl_mcs, slice_prb, scheduling_policy.
- **Targets:** Downlink throughput (tx_brate) and downlink CQI.
- **Preprocessing:**
  - Remove incomplete entries and ensure correct data types.
  - Separate categorical (slice_id, scheduling_policy) from continuous features.
  - Scale numerical features (StandardScaler), encode categorical features (OneHotEncoder).
  - Data split: 81% train, 9% validation, 10% test.

Surrogate Environment Model (DNN)

- **Model:** Deep feedforward neural network using PyTorch.
  - Architecture: Input $\rightarrow$ 512 $\rightarrow$ 512 $\rightarrow$ 512 $\rightarrow$ 256 $\rightarrow$ 128 $\rightarrow$ 64 $\rightarrow$ 32 $\rightarrow$ Output(2).
  - Activation: ReLU; BatchNorm and Dropout (0.3) after each hidden layer.
- **Training:**
  - Batch size: 1024; optimizer: Adam (1×10^{-3}), L2 regularization.
  - Early stopping on validation loss (patience=15).
- **Evaluation:**
  - Evaluate on held-out test set using RMSE and R^2 for both targets.

RL Environment Implementation

- **Custom Environment:** Gym-like interface wraps the trained DNN surrogate for fast, differentiable feedback.
- **Action Space:** Joint PRB allocation and scheduling policy selection.
- **Step Function:**
  - Decodes action, processes input through preprocessor.
  - DNN predicts new throughput and CQI, reward calculated via a custom formula.
  - State updated by sampling or logic; episode ends after max steps.

RL Agent: Dueling Double DQN

- **Agent:** Dueling Double Deep Q-Network (DQN) implemented in PyTorch.
- **Networks:** Online and target networks, dueling architecture separates value and advantage streams.
- **Replay Buffer:** Stores past transitions for off-policy learning; buffer size $= 20{,}000$.
- **Learning:**
  - ϵ-greedy for exploration, Huber loss for stable updates.
  - Target Q-values by Double DQN, periodic target network sync.
- **Training Loop:**
  - Train for multiple episodes, log metrics, and decay ϵ.

Baseline Evaluation

- **Static Baselines:**
  - Allocate min/median/max PRBs.
  - Combined with three scheduling policies (Round Robin, Weighted Fair, Proportional Fair).
- **Comparative Evaluation:**
  - Use fixed initial states for fair comparison.
  - Run 100 test episodes per policy; record rewards, throughput, CQI, buffer.

Visualization and Analysis

- **Training Curves:** Cumulative reward per episode and ϵ decay.
- **Performance Bars:** Compare average reward, throughput, CQI, buffer across policies.
- **Action Histograms:** Show learned PRB allocation and scheduling distributions.
- **KPI Evolution:** Track throughput and CQI during agent training.
- Results quantify RL improvement and interpret agent policy behavior.

Results and Discussion

Surrogate Environment Model Performance

The deep neural network (DNN) trained as a surrogate for the RAN environment is crucial for the RL training phase.
Evaluation Metrics:

- **Root Mean Square Error (RMSE):** Standard deviation of prediction errors; lower is better.
- **Coefficient of Determination (R^2):** Proportion of variance explained; values closer to 1 are better.

DNN Performance on Test Set

- **Downlink Throughput:**
  - RMSE: 0.1069
  - R^2: 0.8809
- **Downlink CQI:**
  - RMSE: 2.2479
  - R^2: 0.3219

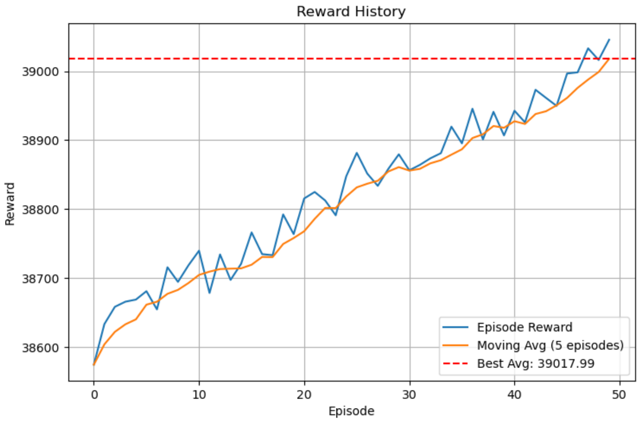
Discussion on DNN Surrogate

High R^2 for downlink throughput indicates good model fit and reliability for RL training.
Lower R^2 for CQI reflects difficulty in capturing channel quality but maintains some predictive ability.
Conclusion: The DNN is suitable as a surrogate environment for RL training.

RL Agent Training Dynamics

Several aspects of Dueling Double DQN training were monitored for learning progress and stability.

Reward Trajectory: Cumulative reward per episode

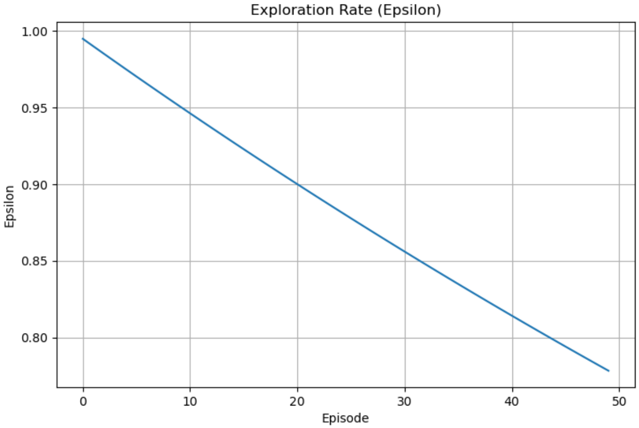


figureCumulative reward per episode during RL agent training. Blue: Raw reward, Orange: Moving average (5 episodes).

Explanation: Reward Trajectory

Figure shows cumulative reward per episode (blue: raw, orange: moving average).
Insight: Upward trend and plateau for moving average indicate stable learning and policy convergence.

Epsilon Decay: Exploration Rate

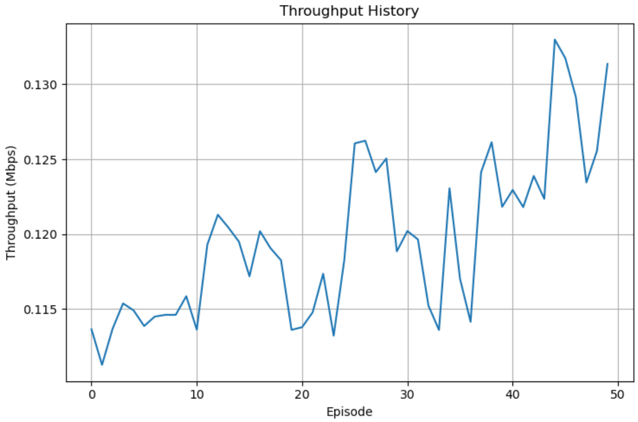


figureExploration rate (ϵ) per training episode.

Explanation: Epsilon Decay Curve

Figure: ϵ starts near 1 (fully exploratory) and decays smoothly as training progresses.
Insight: Proper exploration-exploitation trade-off ensures robust policy learning.

Throughput Evolution

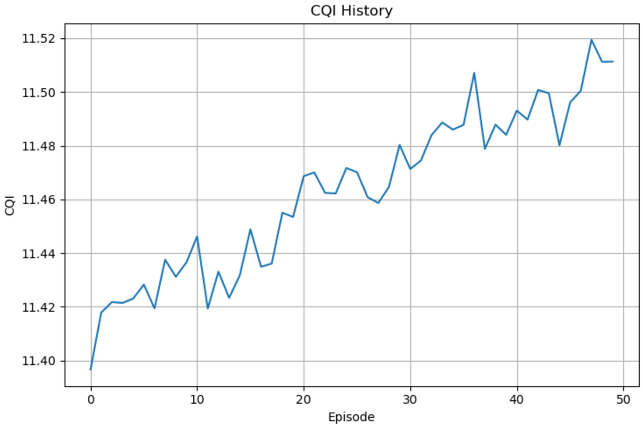


figureAverage episode throughput during RL agent training.

Explanation: Throughput Evolution

Figure shows average throughput per episode.
Insight: Uptrend indicates optimization of downlink rate via RL agent's policy.

CQI Evolution



figureAverage CQI per episode during RL training.

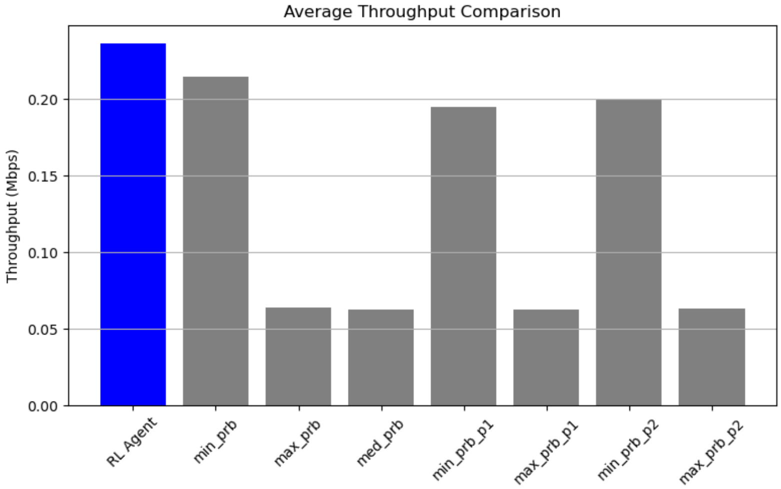
Explanation: CQI Evolution

Figure shows average CQI per episode.
Insight: Increase in CQI indicates better radio link quality optimization.

RL Agent vs. Baseline Policies

Performance was evaluated on 100 independent test episodes, using identical initial states for all policies.

Average Throughput Comparison

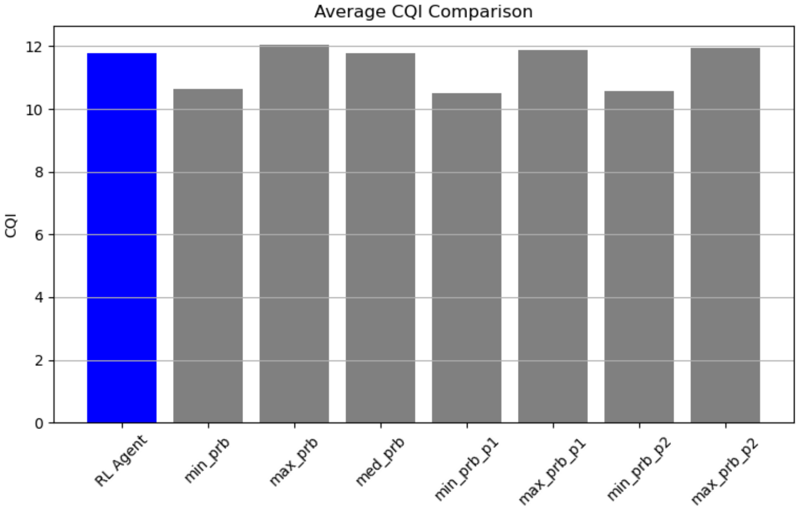


figureComparison of average throughput (Mbps) for RL agent and static baselines.

Explanation: Throughput Comparison

The RL agent outperforms all static baselines in terms of average throughput, optimizing end-user data rates.

Average CQI Comparison

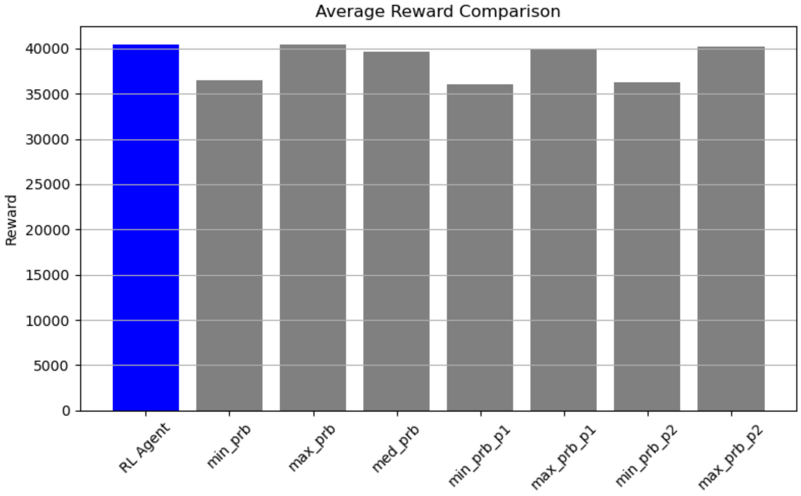


figureComparison of average CQI for RL agent and static baselines.

Explanation: CQI Comparison

RL agent achieves higher average CQI, reflecting effective channel quality optimization.

Average Reward Comparison



figureAverage cumulative reward across policies over test episodes.

Explanation: Reward Comparison

RL agent achieves the highest overall cumulative reward, indicating strong and balanced KPI optimization.

Summary of Average Performance Metrics

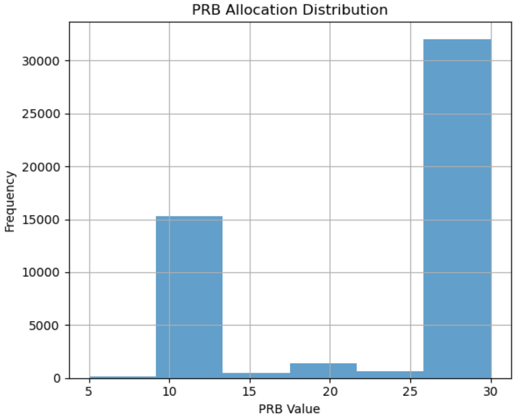
Policy	Avg. Reward	Avg. Throughput (Mbps)	Avg. CQI
RL Agent	**40421.80**	**0.24**	**11.77**
Min PRB + RR	36530.82	0.21	10.63
Max PRB + RR	40465.91	0.06	12.04
Med PRB + RR	39612.83	0.06	11.78
Min PRB + WF	36014.84	0.20	10.51
Max PRB + WF	39941.99	0.06	11.88
Min PRB + PF	36258.17	0.20	10.57
Max PRB + PF	40168.16	0.06	11.95

Performance metrics for RL agent and all baselines.

Explanation: Performance Table

The RL agent achieves the best values in several metrics, confirming it outperforms all baselines when it comes to throughput.

PRB Allocations by RL Agent

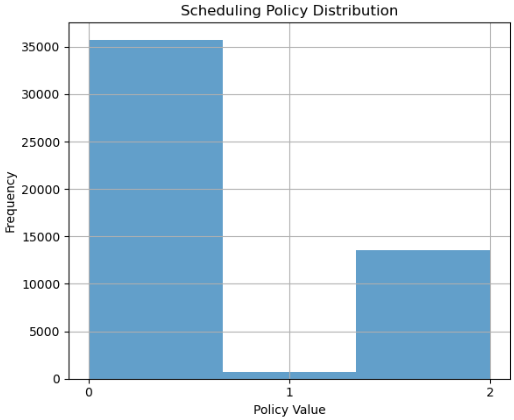


figureHistogram of PRB allocations chosen by RL agent.

Explanation: PRB Allocation Distribution

Histogram shows the agent allocates a diverse range of PRB values, adapting to network conditions in test episodes.

Scheduling Policies by RL Agent



figureHistogram of scheduling policy usage by RL agent.

Explanation: Scheduling Policy Distribution

The RL agent uses all available scheduling algorithms, adjusting dynamically for observed network states.

Conclusions and Future Work

Conclusions and Future Work

- Summary of the thesis and main contributions
- Key conclusions drawn from the results
- Important directions for future research and practical deployment

Summary of Work

Summary of Work

This thesis investigated the application of artificial intelligence (deep learning and reinforcement learning) for intelligent network optimization in Open RAN (ORAN) systems.

1. **Data Acquisition and Preprocessing:** Realistic O-RAN COMMAG dataset from the Colosseum testbed, processed for ML.
2. **Surrogate Environment Modeling:** DNN trained to predict KPIs such as throughput and CQI from state and control action.
3. **RL Agent Design:** Implementation of a Dueling Double DQN to interact with the surrogate model.
4. **Reward Engineering:** Designed a composite reward to balance throughput, CQI, and buffer stability.
5. **Training and Evaluation:** RL agent compared to static baselines on key metrics: reward, throughput, CQI, buffer size.

Used standard ML libraries (PyTorch, scikit-learn) and modern

Conclusions

Conclusions

Key findings from the results:

1. **Feasibility:** DNNs can accurately model RAN KPIs from real-world data.
2. **Efficacy of RL:** Dueling Double DQN learns adaptive, dynamic control policies for ORAN.
3. **Performance Gain:** RL agent achieved a **10.18% increase** in throughput over the best static baseline, and led in CQI and reward metrics.
4. **ORAN Leverage:** Demonstrated how ORAN principles (programmability, data access) enable advanced ML and RL strategies.
5. **Foundation for the Future:** This simulation study shows that integrating AI/ML into RAN management is promising for next-generation networks.

Concluding Statement

Combining data-driven simulation with modern reinforcement learning enables the development and evaluation of intelligent, adaptive network policies, paving the way for high-performance, self-optimizing ORAN deployments.

Future Work

- **Enhanced Modeling**
- **Advancing RL Frameworks**
- **Integration with ORAN Ecosystem**

Final Remark

This project explored practical approaches to ORAN-based intelligent network optimization using data-driven surrogate modeling and reinforcement learning. Our results show measurable gains and adaptivity compared to static baselines, providing a foundation for future research and practical exploitation of AI/ML in open, programmable wireless networks.

References

References I

[1] O-RAN Alliance Working Group 1. *O-RAN Architecture Description*. Technical Specification O-RAN.WG1.O-RAN-Architecture-Description-v06.00: https://www.o-ran.org/blog/o-ran-alliance-introduces-33-new-specifications-released-since-march-2021. Accessed: 2025-05-04. 2023.

[2] Michele Polese et al. "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges". In: *IEEE Communications Surveys & Tutorials* 25.2 (2023), pp. 1376–1411.

References II

[3] Peter Fetterolf. “The economic benefits of open RAN technology”. In: *URL https://infohub. delltechnologies. com/section-assets/acg-the-economic-benefitsof-open-ran-technology* (2021).

[4] Leonardo Bonati et al. “Intelligence and learning in O-RAN for data-driven NextG cellular networks”. In: *IEEE Communications Magazine* 59.10 (2021), pp. 21–27.

[5] Mojdeh Karbalaee Motalleb et al. “Resource allocation in an open RAN system using network slicing”. In: *IEEE Transactions on Network and Service Management* 20.1 (2022), pp. 471–485.

References III

[6] O-RAN Alliance. *O-RAN Alliance White Paper: O-RAN Architecture Overview*. Available online: https://www.o-ran.org/resources. Accessed: 2025-05-04. 2020.

[7] 3GPP. *System architecture for the 5G System (5GS)*. Technical Specification TS 23.501: :https://www.tech-invite.com/3m23/tinv-3gpp-23-501.html. Accessed: 2025-05-04. 2023.

[8] Ziyu Wang et al. "Dueling network architectures for deep reinforcement learning". In: *International conference on machine learning*. PMLR. 2016, pp. 1995–2003.

Appendix

Code Availability

To promote transparency, reproducibility, and further research in this area, all code and resources related to the surrogate model and experimentation have been made publicly available. Interested readers and researchers can access the complete implementation, datasets, and detailed instructions in the following GitHub repository:

`https://github.com/Ram5anthosh/ORAN`