Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Deep Reinforcement Learning-Based Dynamic Channel Reservation for SDN-Enabled Cognitive Radio Networks

Mai Abdelgalel

mabdelgalel@aiu.edu.eg

Article

Keywords: cognitive radio networks, software-defined network, deep reinforcement learning, dynamic spectrum access, quality of service, channel reservation

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11266514/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Deep Reinforcement Learning-Based Dynamic Channel Reservation for SDN-Enabled Cognitive Radio Networks

Mai Abdelgalel[1*]

1 Alamein International University, New Alamein City, Matrouh Governorate, Egypt.

*Corresponding author. E-mail: mabdelgalel@aiu.edu.eg

Abstract

This paper proposes a deep reinforcement learning (DRL)-based dynamic channel reservation scheme for SDN-enabled cognitive radio networks (CRNs). Unlike traditional supervised-learning approaches that rely on pre-labeled data, the proposed DRL agent learns channel reservation policies through interaction with the network environment. The objective of the proposed scheme is to minimize primary users (PUs) blocking probability while maintaining acceptable quality of service (QoS) for secondary users (SUs). The model is evaluated based on key metrics, including PU/SU blocking probability, forced termination probability, and system cost. Under the evaluated simulation scenarios, the DRL-based approach achieves lower values for several blocking and cost metrics than the supervised AI-driven SDN baseline at the reported traffic points, while dynamically adjusting channel reservations as traffic conditions vary.

Keywords: cognitive radio networks, software-defined network, deep reinforcement learning, dynamic spectrum access, quality of service, channel reservation.

1 Introduction

The rapid growth of wireless communication technologies has resulted in an increased demand for radio spectrum, leading to spectrum congestion and inefficient usage. With the transition to 5G networks, the need for flexible and reliable spectrum availability has become even more critical [1]. Traditional cellular networks, which rely on fixed spectrum allocations, fail to meet these requirements.

Cognitive Radio (CR) offers a solution by enabling Secondary Users (SUs) to dynamically access unused portions of the spectrum reserved for Primary Users (PUs). Through Dynamic Spectrum Access (DSA), CR systems identify "spectrum holes" or idle spectrum while minimizing interference with PUs. The overlay method in Cognitive Radio Networks (CRNs) ensures that SUs can access vacant spectrum without compromising PU priority [2],[3]. When a PU requires its channel, the allocated SU must vacate it, often leading to a spectrum handover process [3],[4].

To improve system performance and prevent service interruptions, Dynamic Channel Reservation (DCR) solutions are implemented, reserving specific channels for SUs based on traffic load and network conditions. This increases SU availability while maintaining Quality of Service (QoS) for PUs, enhancing spectrum efficiency and addressing spectrum scarcity.

Traditional systems require both PUs and SUs to search for and allocate spectrum, adding complexity and load. To address this challenge, the proposed system introduces a centralized SDN controller utilizing Artificial Intelligence (AI). This approach reduces the load on equipment, supports centralized system management, and coordinates spectrum allocation for both PUs and SUs [4],[5],[6].

The proposed system seeks to limit PU blocking associated with R-CRN allocations, because PU blocking can reduce service quality and increase operational costs. The AI-enhanced SDN controller monitors PU arrivals and blocking events and reallocates channels from R-CRN to N-CRN to maintain QoS. The supervised-learning component predicts the number of R-CRN channels and estimates costs for PUs and SUs. This integration of AI with SDN supports dynamic decision-making in the modeled system.

The Deep Reinforcement Learning (DRL) agent is evaluated under similar traffic conditions as in [5], with model sizes of M=50 and M=600 channels, and PU arrival rates (λ_p) ranging from 0.7 to 1.0, with low failure rates (λ_f). Key performance metrics such as PU and SU blocking probabilities, forced termination probability, and system cost are evaluated. Simulation results show lower values for several evaluated blocking and cost metrics under the DRL model than under the previous supervised model, while the learned reservation decision also changes with the evaluated traffic conditions.

Traditional spectrum allocation methods are static [7], leading to underutilized frequency bands and congestion. CRNs address this issue by allowing SUs to access idle PU channels, improving spectrum utilization. In SDN-enabled CRNs, a centralized SDN controller dynamically adjusts channel reservations based on current PU and SU activity. The total number of channels is split into reserved R-CRN and non-reserved N-CRN pools, with the SDN controller prioritizing PU traffic to avoid blocking while allocating remaining channels to SUs. The CTMC model represents these dynamics under the stated Poisson-arrival and exponential-service assumptions.

In [5], the reservation count R_t was predicted using a supervised neural network model based on historical data. While this method reduced PU blocking, it had key limitations, such as reliance on labeled training data, offline tuning, and lack of adaptability to unforeseen traffic changes. Moreover, trade-offs were observed: PU protection strategies often degraded SU performance or increased system costs. The supervised model was fixed for two scenarios (PU arrival or PU-block-based), limiting flexibility.

To overcome these challenges, a DRL-based approach is proposed, where a Deep Q-Network (DQN) agent learns reservation policies through interaction with the network environment [8]. The DQN agent learns without pre-labeled training data, using observations of channel occupancy and traffic conditions. Additionally, Double DQN is employed to mitigate overestimation bias in Q-values [9]. The Markov Decision Process (MDP) formulation remains consistent with the prior framework [5]: the state is represented by the current number of occupied PU/SU channels, reserved channel count, and traffic parameters, while the action involves adjusting the reservation count by -1, 0, or +1. The reward function penalizes PU blocking, SU blocking, and system cost using the weights from [5], which can be tuned for optimization.

This study extends the prior SDN-enabled CRN framework with a DRL-based dynamic channel reservation approach, where reservation policies are learned through interaction rather than from pre-labeled offline data. By incorporating Double DQN, the controller is designed to balance PU protection and SU QoS while adapting reservation decisions to changing simulated traffic conditions.

2 Methods

The previous SDN-based CRN framework [5] is extended by replacing the supervised predictor with a DRL agent. The DRL agent observes the same network state, including channel occupancies, PU/SU arrival rates,

and failure rate, and dynamically adjusts the number of reserved channels to optimize a cumulative reward. This reward function penalizes PU blocking, SU blocking, and system cost. The system is modeled as a Markov Decision Process (MDP) with a continuous-time Markov chain (CTMC) model for PU and SU arrivals.

The state space, action set, and reward function are aligned with the prior work [5] to ensure a fair comparison. A Deep Q-Network (DQN) with a target network and experience replay [8] is used to maximize the expected discounted reward over multiple episodes. The model is trained using an epsilon-greedy exploration method to improve the agent's decision-making over time.

The trained DRL model is evaluated using the same traffic scenarios and metrics as in previous work, including PU and SU blocking probabilities, forced termination, and system cost. The results are compared directly with the supervised AI model from the prior work, and percentage differences between the models are reported.

2.1 System and CTMC model

The same CRN system model is adopted as in [5] . A total of Mchannels (here $M = 50$ or 600) are divided into reserved (R-CRN) and non-reserved (N-CRN) pools. Primary users (PUs) and secondary users (SUs) arrive as Poisson processes with rates $\lambda_p$and λ_s(exponential service rates μ_p, μ_s). Channels may fail (rate λ_f) and be repaired (exponential service). At each decision epoch, the SDN controller (our agent) adjusts the number of reserved channels R_t. New PU arrivals beyond $R_t$are blocked. This matches the CTMC assumptions of [5] with low λ_f. The same parameter values are used to enable fair comparison.

2.2 DRL problem formulation

The reservation control is defined as a Markov decision process (MDP):

State

The state includes occupied PU channels n_{PU}, occupied SU channels n_{SU}, and reserved count R_t. Current arrival rates $(\lambda_p, \lambda_s, \lambda_f)\ are\ included$ in s_t. For example, $s_t = [n_{PU}, n_{SU}, R_t, \lambda_p, \lambda_s, \lambda_f\ ]$.

Action

$a_t \in \{-1,0,+1\}$, is allowed corresponding to decreasing, maintaining, or increasing $R_t$by one. Actions are clipped so $0 \le R_t \le M$. This matches the R-CRN adjustment in the prior supervised scheme.

Reward

The blocking and cost are inserted. A typical reward is the negative weighted sum:

$$r_t = -(\alpha P_{\text{PU-block}}(s_t, a_t) + \beta P_{\text{SU-block}}(s_t, a_t) + \gamma\ \text{Cost}(s_t, a_t)) \quad \text{..... (1)}$$

Here $P_{\text{PU-block}}$and $P_{\text{SU-block}}$are the PU/SU blocking probabilities (or counts) induced by the action, and Cost may include channel-switching or energy costs. The same weights $(\alpha, \beta, \gamma)\ are\ used$ from [5] or tune them. This reward encourages minimizing PU-blocking first (high α) while also limiting SU-blocking and cost.

Each episode runs for a fixed horizon (e.g. T time slots). This MDP formulation enables model-free DRL algorithms to learn a channel-reservation policy from interaction with the environment.

2.3 DQN agent architecture

A Deep Q-Network (DQN) agent is implemented as:

Network:

A multi-layer perceptron takes s_t as input and outputs Q-values for each action. For instance, a network of two hidden layers (128 units each, ReLU activation) mapping $\mathbb{R}^{n_s} \rightarrow \mathbb{R}^3$ (for actions $\{-1, 0, +1\}$).

Target network

A separate target network (periodically updated) is used for stable DQN training.

Replay buffer

Experiences (s, a, r, s') are stored in a replay buffer and sampled in batches.

Double DQN

The Double-DQN variant is used to reduce Q-value overestimation during updates.

Loss

The MSE loss is minimized between the predicted Q-value and the target:

$$\text{Loss} = \mathbb{E}\left[\left(Q(s_t, a_t) - \left(r_t + \gamma \max_{a'} Q(s_{t+1}, a')\right)\right)^2\right] \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2)$$

Hyperparameters:

The network and training hyperparameters (learning rate, γ, etc.) are provided below.

Table 1 below presents the key hyperparameters used in the training of the DRL agent. These parameters define the training configuration used in the simulation:

Table 1. Key hyperparameters used to train the DRL agent.

Hyperparameter	Value / Range	Notes
Network Layers	[128, 128]	Two hidden layers with 128 units each
Activation	ReLU	Non-linear activation function
Learning Rate	10^{-3}	Adam optimizer
Discount Factor	$\gamma = 0.99$	Future reward discount
Replay Buffer Size	10,000 transitions	Memory for experience replay
Batch Size	64	Mini-batch update size
ϵ(exploration start)	1.0	100% *random actions initially*
ϵ(final)	0.01	Greedy action eventually
ϵdecay schedule	Linear over 2000 episodes	From 1.0 to 0.01
Target Update Frequency	1000 steps	Update target network weights
Training Episodes	5000+	Sufficient for convergence

2.4 Training procedure

The agent with ϵ-greedy exploration is trained:

Initialize policy network and target network.

For each episode:

Reset environment ($R = 0$, empty channels).

For each time step:

With probability ϵ, select a random action, else $a_t = \arg\max_a Q(s_t, a)$.

Execute a_t, observe next state s_{t+1} and reward r_t.

Store (s_t, a_t, r_t, s_{t+1}) in replay buffer.

Sample a minibatch and perform a DQN gradient step.

Update $s_t \leftarrow s_{t+1}$.

Decay ϵ(e.g. linearly to $\epsilon_{\min}$).

Every N_{target}steps, update the target network.

Repeat for the configured number of training episodes.

This procedure is summarized in the pseudocode section below.

2.5 Evaluation methodology

The DRL controller is compared with the prior supervised model. Evaluation is performed under the following conditions:

PU arrival rates:

0.7, 0.8, 0.9, and 1.0 (Poisson arrival process).

Channel counts: M=50 and M=600.

Fixed low channel-failure rate (0.01).

Relevant SU arrival rates as defined in [5].

Key metrics evaluated include:

PU Blocking Probability

SU Blocking Probability

Forced Termination Probability

Total Cost

The reported metrics are computed for the evaluation settings listed below.

Table 2 shows key simulation parameters for the evaluation. Values in the table represent the settings used in the experiment.

Table 2. Simulation parameters.

Parameter	Symbol	Value / Range
Total Channels	M	600 ,50
PU Arrival Rate	λ_p	(Poisson) 1.0 - 0.7
SU Arrival Rate	λ_s	as in [5], e.g., 0.5
PU Service Rate	μ_p	as in [5], e.g., 1.0
SU Service Rate	μ_s	e.g., 1.0
Channel Failure Rate	λ_f	Very low (0.01)
Simulation Time Per Episode	T	e.g., 10,000 time slots
Reward Weights	α, β, γ	from [5] if any

Table 3 summarizes the scalar evaluation results for the supervised model and the DRL model at M=600 and M=50. Relative reduction is calculated as (Supervised - DRL) / Supervised x 100; positive values indicate a lower value under DRL, whereas negative values indicate an increase.

The PU/SU blocking trade-off is presented separately in Fig. 3 because it is a two-dimensional curve rather than a single scalar metric.

Metric	Supervised (M=600)	DRL (M=600)	Relative reduction (%)	Supervised (M=50)	DRL (M=50)	Relative reduction (%)
PU Blocking Probability	0.09	0.06	33.33	0.09	0.09	0.00
SU Blocking Probability	0.05	0.035	30.00	0.05	0.05	0.00
Forced Termination	0.02	0.015	25.00	0.05	0.075	-50.00
PU Cost	9.5	8.5	10.53	10.0	11.1	-11.00
SU Cost	6.0	5.0	16.67	5.5	4.5	18.18
Reserved Channels (R)	2.5	2.0	20.00	22.5	20.0	11.11
Relative reduction = (Supervised - DRL) / Supervised x 100. Negative values indicate an increase under DRL.						

Table 3. Evaluation results for the supervised and DRL models. Negative relative-reduction values indicate an increase under DRL.

2.6 Use of generative AI

ChatGPT (OpenAI) was used to assist with manuscript formatting and language editing. I reviewed the manuscript and remains responsible for the scientific content, analyses, interpretations and final text.

3 Results

3.1 Evaluation summary

Performance comparisons are reported descriptively using the simulation outputs for the evaluated scenarios. Percentage changes are used to quantify differences between the DRL and supervised models. No inferential statistical significance is claimed in the absence of reported confidence intervals or paired-test statistics.

3.2 Performance comparison

The results show that, in the evaluated simulation scenarios, the DRL model reduces several PU/SU blocking and cost metrics relative to the supervised model. The main observed patterns include:

Adaptability: The DRL model changes its channel-reservation decision as the simulated traffic conditions vary.

PU Protection: Across the PU-arrival-rate experiments in Fig. 1a, the DRL model yields lower PU blocking probabilities than the supervised model.

Cost Efficiency: In the traffic sweep shown in Fig. 2, the DRL model yields lower reported PU and SU costs at the evaluated points.

However, training overhead remains a concern, as the DRL model requires time to learn a useful policy. Future work could explore multi-agent DRL and online adaptation to further improve performance in dynamic environments.

In Fig. 1a, the proposed DRL-based model yields lower PU blocking probability than the supervised baseline at all plotted traffic points. For instance, at a PU arrival rate of 0.8, the blocking probability decreases from 0.072 in the supervised model to 0.050 in the DRL model, corresponding to a reduction of approximately 31%. At a PU arrival rate of 1.0, the blocking probability decreases from 0.098 to 0.065, corresponding to a reduction of approximately 34%.

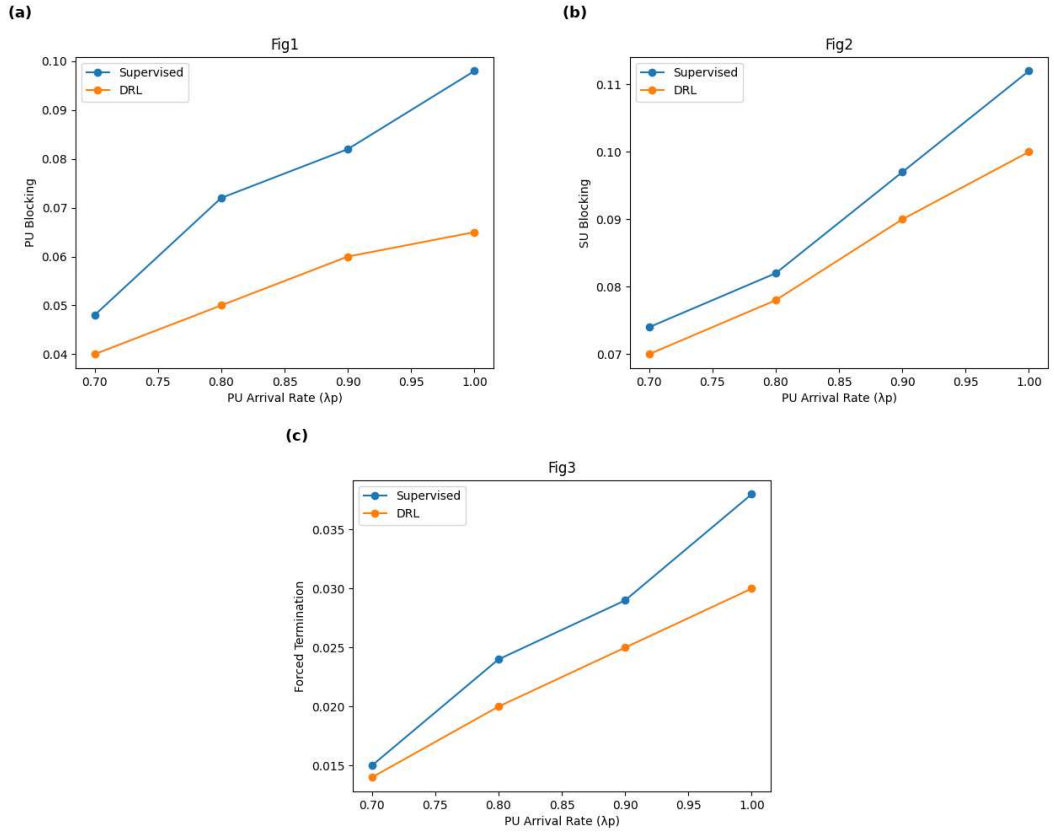

Figure 1. Performance comparison between the supervised and DRL models across PU arrival rates: (a) PU blocking probability; (b) SU blocking probability; and (c) forced termination probability.

These results indicate that the DRL controller adapts its reservation decisions across the simulated traffic conditions and is associated with lower PU blocking in Fig. 1a than the supervised baseline.

In Fig. 1b, the DRL model yields lower SU blocking probability at the reported points. At a PU arrival rate of 0.9, the SU blocking probability decreases from 0.097 to 0.090 (approximately 7% reduction), and at a PU arrival rate of 1.0, it decreases from 0.112 to 0.100 (approximately 11% reduction).

The lower SU blocking values observed in Fig. 1b suggest a more balanced allocation under these simulation settings.

In Fig. 1c, the DRL-based approach yields lower forced termination probability across the evaluated PU arrival rates. At a PU arrival rate of 1.0, the forced termination probability decreases from 0.038 to 0.030, corresponding to a reduction of approximately 21%.

The lower forced-termination values indicate fewer simulated connection interruptions under the evaluated conditions.

In Fig. 2a, the DRL model yields lower PU cost across the reported traffic levels. For example, at a PU arrival rate of 1.0, the PU cost decreases from 9.6 in the supervised model to 7.8 in the DRL model, corresponding to a reduction of approximately 19%.

This cost reduction is consistent with the lower PU blocking observed in the same evaluated scenarios; both quantities are lower in the DRL results.

In Fig. 2b, the DRL model maintains lower SU cost at the reported high-traffic points. At a PU arrival rate of 1.0, the SU cost decreases from 6.2 to 5.5, corresponding to a reduction of approximately 11%.

At the reported points, the lower SU cost is observed alongside the lower SU blocking values shown in Fig. 1b.

In Fig. 2c, the DRL model changes the number of reserved channels (R) as the simulated PU arrival rate varies. For example, at a PU arrival rate of 1.0, the number of reserved channels is reduced from 20 in the supervised model to approximately 5 in the DRL model.

The reduction in reserved channels coincides with lower PU blocking in the corresponding traffic scenarios, although this relationship is demonstrated only within the present simulation setup.

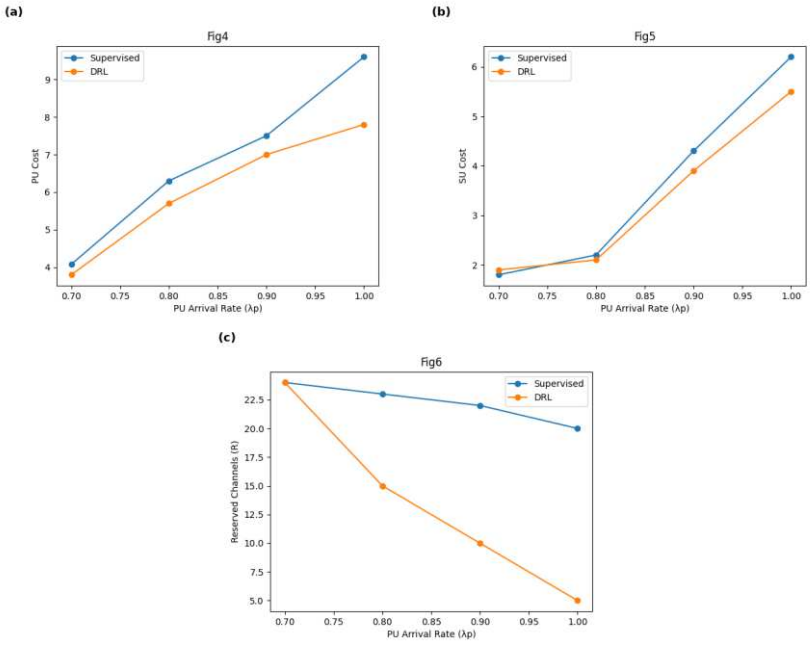

Figure 2. Cost and reservation comparison between the supervised and DRL models across PU arrival rates: (a) PU cost; (b) SU cost; and (c) number of reserved channels.

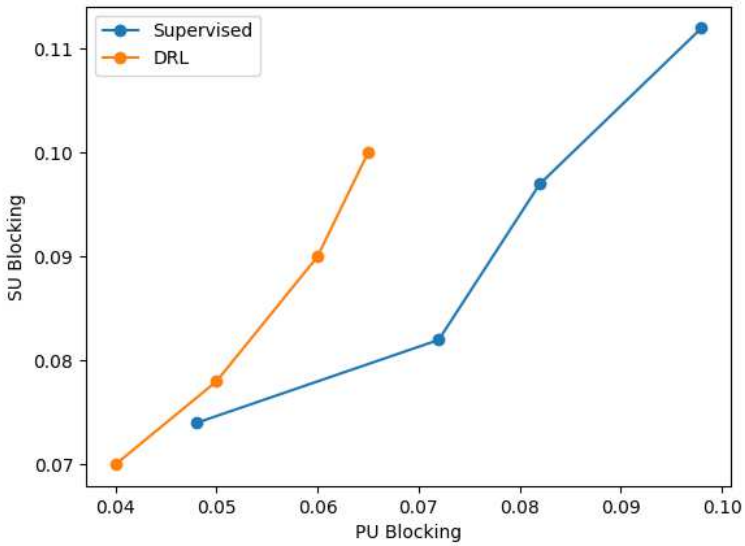

Figure 3. Trade-off between PU blocking and SU blocking for the supervised and DRL models.

In Fig. 3, the DRL trade-off curve occupies lower-PU-blocking values across the plotted points, while SU blocking remains in a comparable range. This visual comparison suggests a favorable PU/SU blocking trade-off for the DRL controller under the evaluated scenarios.

These plotted results suggest that the DRL approach can shift the PU/SU blocking trade-off under the evaluated scenarios; they do not establish a globally optimal trade-off.

Overall, the simulations indicate that the DRL-based model can provide a more adaptive channel-reservation strategy and lower values on several blocking and cost metrics relative to the supervised AI-SDN model under the evaluated conditions.

4 Discussion

In this paper, we proposed a DRL-based approach for dynamic channel reservation in Software-Defined Networking (SDN)-enabled Cognitive Radio Networks (CRNs). Unlike the supervised baseline, the DRL agent learns a reservation policy through interaction with the network environment and does not require pre-labeled targets for the reservation decision.

Our approach was evaluated under selected traffic conditions. Across the PU-arrival-rate sweeps in Figs. 1 and 2, the DRL model produced lower reported blocking and cost values than the supervised baseline at the plotted points. The summary results in Table 3 also show that the gains are scenario-dependent: at M=50, forced termination and PU cost increase relative to the supervised values, while SU cost and reserved-channel count decrease and PU/SU blocking remain unchanged.

Within these simulation settings, the framework shows potential for adaptive spectrum management. Double DQN and experience replay are used to support learning stability; however, the present results do not isolate their individual contributions or establish deployment performance in a real network.

Limitations

The findings should be interpreted within several limitations. First, the evaluation is simulation-based and assumes Poisson arrivals, exponential service/repair times, a low channel-failure rate, selected channel counts,

and parameter settings inherited from [5]; performance may differ under bursty or non-stationary traffic, correlated failures, heterogeneous QoS requirements, or different network sizes. Second, the learned policy depends on the selected reward weights and training hyperparameters, and training introduces computational overhead. Third, the main benchmark is the prior supervised AI-SDN model, so broader comparisons with heuristic, optimization-based, and other DRL baselines are needed. Fourth, no real SDN/CRN testbed validation is presented. Finally, the reported comparisons are descriptive; inferential statistical uncertainty is not established in the present results. Future work should address these limitations through broader baselines, repeated-seed uncertainty analysis, adaptive or multi-agent DRL methods, and real-world/testbed validation.

In conclusion, under the evaluated simulation scenarios, the DRL-based controller provides an adaptive alternative to the supervised baseline and achieves lower values on several blocking and cost metrics. The observed benefits are scenario-dependent, and real-world validation together with broader statistical and algorithmic comparisons is required before drawing general conclusions about deployment performance.

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Author Contributions

I conceived and designed the study, developed the methodology, performed the simulations and analysis, interpreted the results, and wrote and revised the manuscript. I approved the final version of the manuscript.

Competing Interests

I declares no competing interests.

References

[1] K. S. Yuvaraj and P. Priya, "Common control channel-based spectrum handoff framework for cognitive radio network", Proc. Int. Conf. Current Trends Towards Converging Technol. (ICCTCT), pp. 1-5, Mar. 2018.

[2] Namal S., Ahmad I., Saud S. et al.: ‘Implementation of OpenFlow based cognitive radio network architecture: SDN&R’, Wirel. Netw., 2016, 22, pp. 663–677

[3] M. Cicioğlu, S. Cicioğlu, and A. Çalhan, "SDN-enabled Cognitive Radio Network Architecture," IET Communications, vol. 14, no. 18, Nov. 2020, doi: 10.1049/iet-com.2019.1346.

[4] Mai M. Abdelgalel, H. N. Kheirallah, M. R. M. Rizk, and N. M. El Azaly, "Performance Analysis in the Presence of Channel Failure in Cognitive Radio Networks with Dynamic Spectrum Reservation," IEEE Access, vol. 12, pp. 37483-37492, Feb. 2024. doi: 10.1109/ACCESS.2024.3371884.

[5] M. M. Abdelgalel, H. N. Kheirallah, and M. R. M. Rizk, "Dynamic Channel Reservation in Cognitive Radio Networks Using AI Driven SDN Approach," in Proc. 2024 Int. Conf. Future Telecommunications and Artificial Intelligence (IC-FTAI), Alexandria, Egypt, 2024, pp. 205-212, doi: 10.1109/IC-FTAI62324.2024.10950028.

[6] I. A. M. Balapuwaduge and F. Y. Li, “System Times and Channel Availability Analyses in Multi-channel Cognitive Radio Networks”, in Proc. IEEE ICC, pp. 1320-1325, Sydney, Australia, 10-14 Jun. 2014.

[7] W. Yafeng, L. Chao, W. Tianwei and W. Xiang, "Dynamic channel reservation for cognitive radio networks", Proc. IEEE Int. Conf. Comput. Intell. Commun. Technol., pp. 339-343, Feb. 2015.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. A. Riedmiller, "Playing Atari with Deep Reinforcement Learning," arXiv:1312.5602, 2013.

[9] H. van Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," Proc. AAAI Conf. Artif. Intell., vol. 30, no. 1, pp. 2094-2100, 2016, doi: 10.1609/aaai.v30i1.10295.