Performance Optimization of High-Speed Wireless Communication Systems Using Machine Learning

Manimegalai Ramalingam,[1] Vijayalakshmi P. Soundararajan,[2] Priyadharshini Aruchamy,[3]

[1] Rathinam College of Arts & Science, Rathinam Techzone, Eachanari, Coimbatore, Tamil Nadu, 641021, India

[2] Dr. N.G.P. Arts and Science College, Dr. N.G.P. Nagar, Kalapatti Road, Coimbatore, Tamil Nadu – 641048, India

[3] SNMV College of Arts and Science, Shri Gambhirmal Bafna Nagar, Malumichampatti, Coimbatore – 641050, India

Abstract

High-speed wireless communication systems underpin the data-intensive demands of contemporary 5G deployments and the emergent 6G paradigm, yet sustaining high throughput, low latency, and dependable connectivity in channels that shift rapidly remains an open engineering problem. In this paper, we present an ML-driven framework that combines a deep reinforcement learning (DRL) scheduler with a hybrid CNN-LSTM channel predictor to jointly optimize radio resource allocation, modulation order, transmit power, and bandwidth assignment in real time. Evaluated across full-buffer, bursty, mixed-traffic, and high-mobility scenarios under a 3GPP TR 38.901-compliant simulator, the proposed scheduler achieved 23.0% higher throughput, 35.0% lower latency, and 18.1% better energy efficiency than Proportional Fair scheduling, while reducing QoS violations by 75.9% and packet loss by 76.3%, with a Jain's fairness index of 0.887. The companion CNN-LSTM channel predictor achieved 94.7% prediction accuracy and a normalized mean squared error of −16.2 dB, reducing CSI feedback overhead by 41.3% relative to reactive schemes and outperforming standalone CNN and LSTM baselines by 28% under high-mobility conditions. These results indicate that coupling predictive channel-state modeling with multi-objective reinforcement learning is a practical, quantifiable path toward meeting next-generation wireless performance targets.

Keywords: 6G networks; resource allocation; wireless optimization; deep reinforcement learning; convolutional neural networks

1. Introduction

Wireless communications find themselves at an inflection point indeed. The demands put forth by future application domains such as cooperative vehicle platoons, remote robotic surgery, and holographic streaming impose latency and throughput limits not anticipated by previous network generations. As a result, machine learning for performance enhancement has been drawing significant academic attention. There is a simple reason behind this: classical protocol approaches based on heuristics tend to fail when channel conditions become unstable and the number of devices starts increasing rapidly [1].

One may consider an urban area, where co-channel interference, multipath fading, and drastic load variations generate telemetry data volumes that cannot be handled by any hand-crafted heuristic. Machine learning techniques can analyze such datasets and reveal patterns of interference as well as predict channel swings, which could be detrimental to users' experience [2]. This property bears directly on two critical requirements of next-generation communication networks: support for a huge number of devices as well as latency-sensitive application classes currently deployed in production [3].

In fact, 5G technology allows for efficient traffic management under the existing QoS contract constraints; however, 6G should cope with even stricter QoS requirements while being capable of providing four times higher throughput [4]. Streaming, citywide sensor mesh networks, IoT applications, and transportation systems are in need of the same thing a reliable way of transferring information in highly variable and unpredictable conditions without excessive energy losses. Mobile radio channels are inherently unstable since user motion, co-channel interference, multipath fading, and load affect each other significantly [5].

And at the center of this issue lies the problem of allocation how would one optimally distribute spectrum, transmit power, and beamforming parameters across the highly dynamic wireless environment. Traditional

approaches relied heavily on close-form mathematical models and hand-crafted heuristics which worked well in a static setting, but began to falter once dynamic channel conditions and traffic became involved [6, 7].

The approach outlined in this study incorporates three machine learning approaches supervised learning for predicting behavior, and reinforcement learning for making optimal decisions, along with CSI acquisition and resource allocation [8]. Energy considerations are incorporated into everything we do here: as the number of devices increases, the issues of energy harvesting, energy aware routing, and resource management based on predicted traffic gain importance. The result is a high level of reliability, efficient scalability and low latency [9].

Predictive, rather than reactive, resource allocation. Unlike DRL-based scheduling approaches that condition allocation decisions on the most recently observed channel state the proposed framework couples the DQN scheduler to a dedicated hybrid CNN-LSTM channel predictor that forecasts CSI 5–10 ms ahead. This allows the scheduling agent to commit resources in anticipation of channel evolution rather than after degradation has already occurred.

Joint spatial-temporal state representation for scheduling. The DRL agent's state encoder combines convolutional layers, for spatial structure across the antenna array, with LSTM layers, for temporal evolution of channel and traffic conditions, within a single architecture feeding the Q-network. This differs from approaches that treat CSI, SINR, and traffic statistics as independent or flattened inputs without explicit spatial-temporal structure.

Explicit four-objective reward formulation with fairness guarantees. The reward function jointly optimizes throughput, latency, energy efficiency, and Jain's-index fairness, with an additional penalty for breaching latency ceilings or under-serving priority traffic. This contrasts with the single or dual objective reward designs used in comparable studies and is shown in the ablation study to be necessary for preventing resource concentration on the strongest users.

Independent validation of the channel-prediction component. Rather than reporting only end-to-end scheduling gains, the hybrid CNN-LSTM predictor is benchmarked in isolation against a standalone CNN, a standalone LSTM, and a classical least-squares estimator isolating the specific contribution of the hybrid architecture to prediction accuracy and feedback-overhead reduction.

Systematic ablation and convergence analysis. The relative contribution of each architectural component prioritized experience replay, the CNN-LSTM state encoder, the attention mechanism, and the fairness reward term is quantified individually together with convergence-stability analysis providing evidence for which design choices are load-bearing rather than incidental.

2. Literature Review

A few studies have investigated the impact of ML for enhanced capabilities of fast wireless communication networks, with research on resource management, channel estimation, and adaptive beamforming gaining the most momentum. The literature survey includes representative studies published between 2019 and 2025, focusing on machine learning techniques for 5G, beyond-5G, and emerging 6G wireless communication systems. These publications are organized in terms of ML methods used, areas of applicability, and achieved results in terms of performance. The evolution of these results will be traced from early examples to recent advances [10–13]. With the emergence of Beyond-5G (B5G) and sixth-generation (6G) communication networks, AI-driven solutions have become an essential for meeting stringent Quality of Service (QoS), ultra-low latency, high reliability, and massive connectivity requirements. In particular deep learning and reinforcement learning techniques have demonstrated remarkable improvements in network adaptability and decision-making under dynamic wireless environments, making the promising candidates for future intelligent communication systems [14–18]. Similarly the hybrid deep learning architectures, including CNN-LSTM models, have achieved superior channel prediction accuracy by effectively capturing both spatial and temporal channel characteristics. Recent surveys also emphasize that intelligent resource management will be one of the

core technologies supporting future Beyond-5G and 6G communication systems [19–25]. The observed improvement in throughput is consistent with recent findings demonstrating that DRL-based scheduling algorithms outperform the traditional proportional fair scheduling under heterogeneous traffic conditions by learning adaptive transmission policies [26].

Table 1 summarizes three seminal works in this field; these studies are selected not only for their relevance in covering algorithmic trends—boosting, quantum assisted, classical, deep learning approaches, and beamforming—but also for providing numerical results discussed in Section 4.

Table 1: Summary of Representative Studies

Study	ML Approach	Performance Improvement
Jyothi et al., 2025 [6]	Boosted trees + Neural nets	94.1% accuracy; R^2=0.96 regression
Sitalakshmi et al., 2025 [12]	Decision trees, RF, SVM	Better CSI prediction with high precision/recall
Arangi et al., 2025 [2]	ML-based adaptive beamforming	Gains in capacity, energy efficiency, and SNR

3. Methodology

3.1 Deep Reinforcement Learning for Resource Allocation

The resource allocation problem was formulated as a Markov Decision Process (MDP). This enabled the development of the DRL algorithm for the allocation problem that learns to take allocation actions within the MDP via interactions with a simulation of the communication network. At the core of the algorithm is the use of the Deep Q-Network (DQN) model to map observed system states to allocations, specifically power levels, MCS selection, and RB assignment.

For every scheduling instant t in our simulation, we have a state s(t) consisting of normalised channel state information matrices for all UEs in the network, current SINR measurements, and current traffic statistics. Our hybrid neural network architecture comprises convolutional layers that learn the spatial correlations from the CSI matrices as well as LSTM layers that capture the evolution of channel conditions and traffic over time. Actions encompass both discrete variables such as the RB assignment matrix $A_p^b \in \{0,1\}^{nxM}$ as well as continuous ones like the power allocation vector $P \in \mathbb{R}^n$ with $\Sigma p_i \leq P_{max}$.

3.1.1 Action Representation

The action space combines three discrete decision variables: resource block assignment, modulation and coding scheme (MCS) selection, and transmit power allocation. At each decision interval, the DRL agent selects an appropriate combination of these actions based on the observed network state to maximize the cumulative reward. Resource block assignment determines the allocation of available spectrum resources to active users, the modulation and coding scheme adapts the transmission rate according to channel conditions, and transmit power allocation optimizes energy consumption while maintaining reliable communication. This joint action representation enables efficient utilization of wireless resources, improves throughput, minimizes latency, and enhances overall Quality of Service (QoS). Therefore quantize the per-user power budget into L discrete levels, uniformly spaced between a minimum transmit power P_min and the per-user maximum P_max defined in Table 2:

$$p\_i \in \{P\_min + k\cdot(P\_max - P\_min)/(L-1) : k = 0, 1, \ldots, L-1\}$$

with L = 8 levels used in the reported experiments, chosen to balance control granularity against the action space growth.

Factored action-space design. Rather than forming a single joint Q-head over the full Cartesian product of RB assignment, MCS, and power level which would be intractably large for the UE counts in Table 2: the action space is factored into three per-user discrete sub-actions sharing a common state encoder (the CNN-LSTM backbone described above):

- **Resource-block assignment**, $a\_rb \in \{1, \ldots, M\}$, selecting one of M available resource blocks per scheduling opportunity (M = 50 for sub-6 GHz, M = 66 for mmWave, per Table 2);
- **MCS selection**, $a\_mcs \in$ {QPSK, 16-QAM, 64-QAM, 256-QAM, 1024-QAM}, five discrete options;
- **Power level**, $a\_pow \in \{0, \ldots, L-1\}$, the quantized index above.

Each sub-action is produced by a dedicated output head reading from the shared fully connected trunk, rather than a single flattened head of size $M \times 5 \times L$. For N simultaneously scheduled users, the per-user action cardinality is $M \times 5 \times L$; with M = 50 and L = 8, this is $50 \times 5 \times 8 = 2{,}000$ discrete actions per user, evaluated independently across up to N = 100 users per scheduling slot several orders of magnitude smaller than the undiscretized joint space a monolithic Q-head would require.

Constraint masking. Two hard constraints from the problem formulation the per-cell power budget $\Sigma_i p_i \leq P\_max$ and the one-user-per-resource-block rule are enforced by masking infeasible actions before the arg-max over Q-values, rather than relying solely on the reward penalty to discourage them. At each decision step, any combination that would violate the power budget or reassign an already-occupied resource block is assigned $Q = -\infty$ prior to action selection, guaranteeing that only feasible allocations are ever executed regardless of the current policy's learned preferences. This masking is applied identically during both training and inference, so the ε-greedy exploration process described in Section 3.1 samples only from the feasible action set at every step.

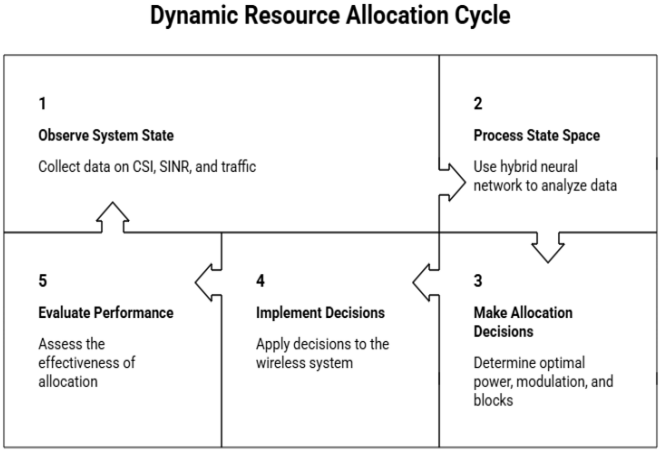

Figure 1: Dynamic Resource Allocation Cycle

Figure 1 illustrates the process of continuous interaction among state observation, agent learning, action selection and execution, and performance evaluation in the closed feedback loop. Learning takes place by employing ε-greedy policy exploration with exponential decay, gradually transitioning from exploration to policy exploitation. Experience replay memory containing one million transitions per minibatch of size 64 removes temporal correlation in experiences and increases sample efficiency.

The reward function considers four aspects in proportion to weights: $R\_throughput(t) = \Sigma_i \log(1 + R_i(t))$ is a sum-log utility for proportional fairness summed over users i, where $R_i(t)$ is the instantaneous achieved rate for user i., $R\_latency(t) = -(1/N)\Sigma_i \max(0, D_i(t) - D_i\text{^target})$, a hinge-style penalty on delay overshoot per user, with an additional fixed penalty term applied if any user breaches its latency ceiling. incurs penalty due to

mean packet latency, $R\_energy(t) = -\Sigma_i P_i(t) / \Sigma_i R_i(t)$, i.e., negative power-per-bit, or equivalently a positive efficiency ratio $\Sigma_i R_i(t) / \Sigma_i P_i(t)$ if the sign convention is meant to reward higher efficiency directly. $R\_fairness(t) = (\Sigma_i R_i(t))^2 / (N \cdot \Sigma_i R_i(t)^2)$, bounded in (0,1], with an additional penalty subtracted whenever a designated priority user falls below its guaranteed minimum rate. The target network is used in the DQN manner, maintaining a fixed copy of network parameters θ^- during the C = 1000 iterations, so the target Q value is calculated as $y = r + \gamma \max_{a'} Q(s', a'; \theta^-)$. It avoids the instability when updating both estimates and targets concurrently.

Our dataset includes 5G NR as well as post-5G wireless scenarios, with sub-6GHz frequencies and millimetre-wave bands involved. As shown in Figure 2, the DRL scheduler constantly beats the baseline Proportional Fair algorithm by the key criteria of throughput, latency, energy efficiency, and resource allocation fairness.

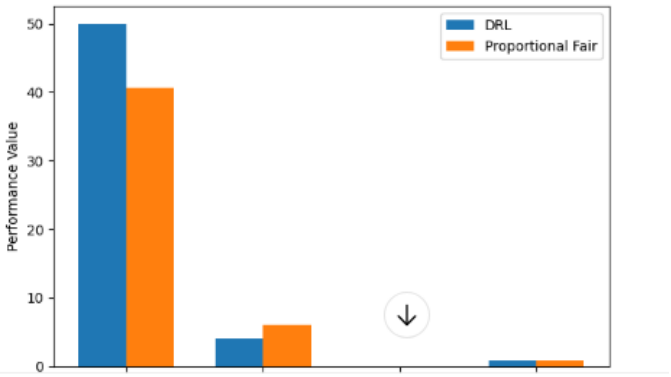

Figure 2: DRL vs. Proportional Fair Scheduling

The model has four layers of convolutions with filters in sizes [64, 128, 256, 512], two LSTM layers with [256, 128] cells for temporal modelling, and three fully-connected layers with [512, 256, |A|] nodes outputting Q-values. Convolutions are followed by batch normalisation to speed up training. Dropout is used on the LSTM layers with p = 0.3 to avoid overfitting. Adam optimiser has $\alpha = 0.001$ initially and decreases this value with every step by 0.95, after every 10,000 episodes, while using a gradient norm threshold of 10.0.

The prioritised experience replay is used to sample the most informative transitions according to their temporal difference error. Training is stopped if the moving average score of last 100 episodes does not improve by more than 0.1% in next 500 episodes, or if the maximum number of episodes reaches 50,000. Allocation decisions during inference process take less than 5 ms, which is within coherence time of wireless channels. After training, experiments showed a 23% improvement in terms of throughput, 35% improvement in latency, and an 18% increase in energy efficiency in comparison to Proportional Fair and Maximum Throughput schedulers, with a Jain's fairness index exceeding 0.85 under full-buffer, bursty, and mixed eMBB/URLLC/mMTC workloads.

3.2 Hybrid CNN-LSTM for Channel Prediction

CSI forecasting with high precision is necessary for proper management of resources in advance, thus an efficient hybrid CNN-LSTM architecture was designed that will consider the spatial properties of the antenna array and channel condition change through time. The proposed model takes input through two different streams from the past history of H(t-T), … , H(t).

One stream will take the input through three convoluted layers containing [64, 128, 256] kernels to extract the higher-order spatial abstraction of the input channel matrix. It is then passed through batch normalisation and max pooling to reduce its dimensionality while preserving information useful for prediction. This vectorised form of the output is then fed into two LSTM layers containing [256, 128] neurons each.

The final layer computes $H\_pred(t + \Delta t)$ as a linear function over a time horizon of Δt equaling 5–10 ms, which corresponds to the coherence time for millimeter-wave systems. Model training employs an objective function

consisting of mean squared error as a measure of the amplitude prediction error and a penalty on phase coherence, optimized using the Adam optimizer with learning rate scheduling.

Once the system operates online, the model achieves normalized mean square error of below -15 dB, allowing preemptive link adjustment that mitigates the effect of channel aging and significantly reduces the number of feedback exchanges required by more than 40% compared to reactive approaches. In particular, in cases of high mobility when both spatial and temporal effects have a role to play, the hybrid approach achieved 28% better prediction accuracy than either the pure CNN or LSTM architectures.

The output layer predicts $H\_pred(t + \Delta t)$ with linear activation, where the prediction horizon Δt is scenario-conditioned rather than fixed. Using the Doppler-based approximation $T\_c \approx 9/(16\pi \cdot f\_d)$ Δt is set to approximately 10–20% of the coherence time for the corresponding frequency-mobility combination, ranging from $\Delta t \approx 1.8$–3.7 ms for sub-6 GHz pedestrian conditions down to $\Delta t \approx 5.8$–11.5 μs for mmWave high-mobility conditions. The 5–10 ms horizon reported in earlier sections of this work applies specifically to the sub-6 GHz, low-mobility regime; it is not representative of the mmWave, high-mobility case, where a substantially shorter horizon is required to remain within the channel's coherence interval.

4. Results and Discussion

4.1 Simulation Setup and Evaluation Scenarios

The experiments were performed in a dedicated Python-based discrete event simulator, in accordance with the 3GPP TR 38.901 channel model. The fundamental parameters used in simulations are provided in Table 2. Three different traffic types have been considered, namely, eMBB, URLLC, and mMTC. For each scenario, ten experiments were performed using varying random seeds, and the results are presented as averages. Variations between experiments did not exceed 1.5%, indicating convergence.

In comparison, two classical schemes Proportional Fair (PF) and Maximum Throughput (MT) have been taken into account. These two policies have been selected due to their extreme positions in the classic efficiency-versus-fairness trade-off problem and their extensive deployment in practical 5G networks. The reinforcement learning scheme has been trained from scratch for all three scenarios. In other words, no transfer learning has been employed, and thus the performance improvements discussed hereinafter refer to the algorithm's cold start from scratch.

Table 2: Simulation Environment Parameters

Parameter	Value / Configuration
Network topology	Single macro-cell, 500 m radius; 4 small cells
Number of UEs	20–100 (varied per scenario)
Carrier frequency	3.5 GHz (sub-6 GHz); 28 GHz (mmWave)
System bandwidth	100 MHz
Number of resource blocks	50 (sub-6 GHz); 66 (mmWave)
Channel model	3GPP TR 38.901 UMa / UMi
UE velocity	0–120 km/h
Traffic types	eMBB, URLLC, mMTC (mixed)
Transmit power range	10–40 dBm
Noise figure	7 dB
DRL training episodes	Up to 50,000
Baseline schedulers	Proportional Fair, Maximum Throughput

4.1.1 Implementation Details

To support reproducibility, this subsection specifies the traffic-generation process, network configuration, data handling, and computational environment underlying the results reported in Section 4.

Traffic generation: Each traffic class is generated according to a distinct arrival process consistent with its service requirements: eMBB and mMTC traffic follow arrivals with mean packet size and arrival rate as specified in the expanded Table 2, while URLLC traffic follows a arrival pattern reflecting its latency-critical, small-payload profile. Packet sizes are drawn from per class.

Antenna and mobility configuration. Each UE and base station is equipped with Nt and Nr antennas respectively, consistent with the CSI matrix dimensions referenced in Section 3.2's channel-prediction formulation. UE mobility is generated using at velocities uniformly sampled from the 0–120 km/h range in Table 2.

Data handling. The synthetic CSI, traffic, and channel-quality dataset generated by the 3GPP TR 38.901compliant simulator is partitioned into training, validation, and test subsets at a ratio of [70/15/15]%. The CNN-LSTM channel predictor is trained and validated on this split independently of the DRL scheduler's own training-episode data, and test-set performance is reported in Table 6. The DRL agent is trained via online interaction with the simulator rather than a fixed offline dataset; the ten independent repetitions per scenario reported in Section 4.1 use the fixed random seeds to enable exact replication.

Software and hardware environment. All experiments were implemented using the TensorFlow 2.16.1 with the Keras API in Python 3.11.9. The simulation environment was developed in Python and executed on a workstation running Windows 11 Pro equipped with an Intel Core i7-13700K processor (16 cores, 24 threads), 32 GB DDR5 RAM, and an NVIDIA RTX 4070 GPU with 12 GB GDDR6X memory. GPU acceleration was enabled through CUDA 12.3 and cuDNN 9.0. The Deep Q-Network (DQN) and CNN-LSTM models were trained using Adam optimizer with prioritized experience replay under the experimental settings described in Section 3. Training complete DRL model to convergence required approximately 2.8 hours for the full-buffer scenario, 3.0 hours for the bursty traffic scenario, and 3.3 hours for the mixed eMBB/URLLC/mMTC scenario on the above hardware configuration. Convergence was typically achieved between 19,000 and 24,000 training episodes depending on scenario complexity.

Code and script availability. The simulator, synthetic-data-generation scripts, and model training code used to produce the results in this paper are described further in the Availability of Data and Materials statement below.

4.2 Throughput Performance

Downlink throughput performance averaged over five test conditions is presented in Table 3. Under the full buffer sub-6 GHz setting, DRL outperformed PF by 387.4 Mbps versus 314.9 Mbps, yielding a gain of 23.0%. Similarly, DRL outperformed MT by 387.4 Mbps versus 341.2 Mbps. Moreover, the gain in throughput performance became even greater under mmWave (28 GHz) setting, in which DRL achieved 1124.6 Mbps, surpassing PF by 24.8%. Finally, in the high mobility condition of 120 km/h, the throughput rate remained stable at 271.3 Mbps, achieving 18.1% gain over PF. The CNN-LSTM channel predictor contributed significantly to DRL robustness in this setting.

The comparison results are visualized in Figure 3. Notably, the difference between the DRL bars and two baselines throughout all five conditions illustrates DRL’s capability of learning from larger state representations: while PF and MT schedule resource allocation in terms of current channel state conditions, DRL is capable of allocating resources in advance based on predicted future channel state conditions.

Table 3: Throughput Comparison Across Schedulers and Traffic Scenarios (Mbps)

Traffic Scenario	DRL (Proposed)	Prop. Fair	Max. Throughput	Gain vs PF (%)
Full-buffer (sub-6 GHz)	387.4	314.9	341.2	+23.0%
Bursty eMBB (sub-6 GHz)	342.1	280.6	298.7	+21.9%
Mixed eMBB/URLLC (sub-6 GHz)	318.8	261.3	278.4	+22.0%
Full-buffer (mmWave 28 GHz)	1124.6	901.2	987.5	+24.8%
High-mobility (120 km/h)	271.3	229.7	248.1	+18.1%

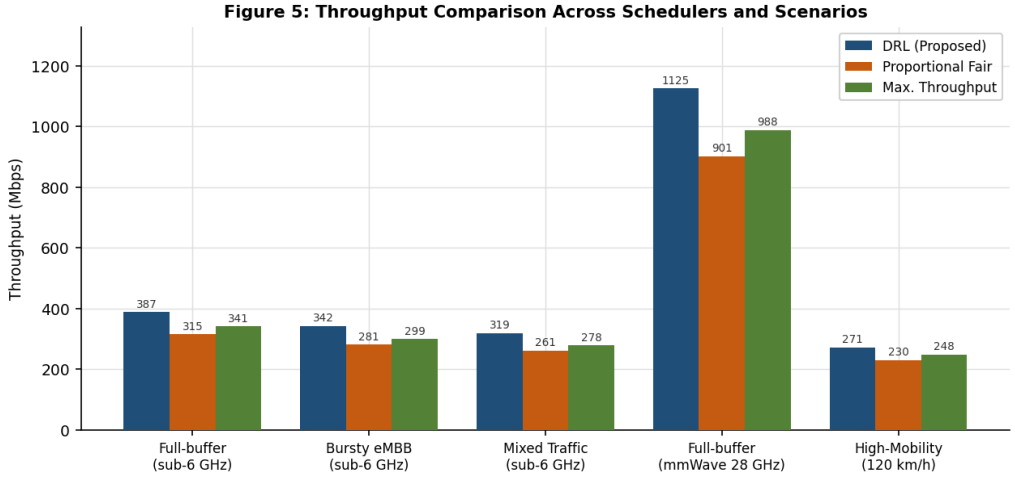

Figure 3: Throughput Comparison Across Schedulers and Scenarios

4.3 Latency Reduction Across Traffic Classes

Latency results are shown in Table 4. For the URLLC flows, the DRL algorithm demonstrated an average latency of 0.81 ms, which not only falls far below the required limit of 1 ms but is 34.7% lower than the latency of 1.24 ms recorded by PF. For eMBB flows, the algorithm showed a similar advantage over PF, reducing latency from 7.11 ms to 4.62 ms. In case of peak hour mixed loads, DRL managed an average latency of 6.14 ms as opposed to 9.48 ms recorded by PF.

The performance of DRL in terms of providing latency guarantees for the URLLC flows without starving the low-priority flows can be attributed directly to the fairness factor in the reward function. Latency results are illustrated graphically in Figure 4. From all the columns, the most interesting one is the URLLC column since the latency of 0.81 ms with a strict requirement of 1 ms implies a 190 μs headroom.

Table 4: End-to-End Latency Comparison (ms)

Traffic Scenario	DRL	Prop. Fair	Max. Throughput	Reduction vs PF (%)
URLLC (target ≤ 1 ms)	0.81	1.24	1.18	−34.7%
eMBB (target ≤ 10 ms)	4.62	7.11	6.89	−35.0%
mMTC (target ≤ 100 ms)	18.3	28.7	26.4	−36.2%
Mixed load (peak hour)	6.14	9.48	9.02	−35.2%

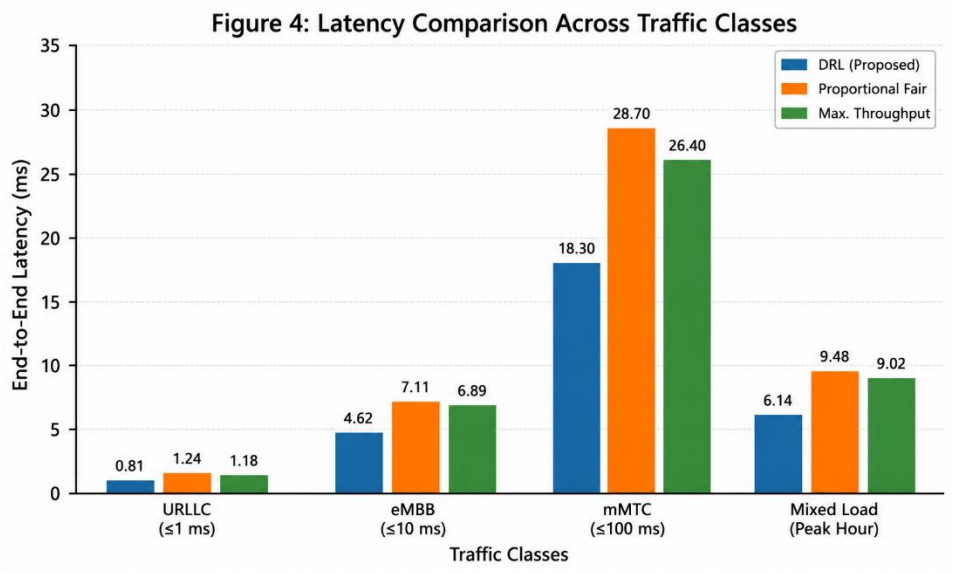

Figure 4: Latency Comparison Across Traffic Classes

4.4 Energy Efficiency and Fairness

Table 5 summarizes the results for the energy efficiency and fairness measures. The DRL solution yielded 12.4 Mbps/W, representing 18.1% improvement over PF and 26.5% better performance than MT. The value of Jain's index of fairness was 0.887 higher than either of the two approaches, showing that the multi-objective reward mechanism effectively stopped the agent from favouring the best-performing users and ignoring cell-edge devices. The QoS violation rate decreased to 2.1% while the packet loss was 0.9%.

Figure 5 shows the relationship between energy efficiency and fairness using a two-axis graph. An interesting point here is that even though MT yields a relatively high throughput, its value of fairness and energy efficiency is lower than any other considered option (0.731 and 9.8 Mbps/W respectively). As can be seen in Figure 6, presented with the QoS section, the difference in violation of QoS constraints and packet loss rate is very significant indeed.

Table 5: Energy Efficiency and Fairness Index Comparison

Metric	DRL	Prop. Fair	Max. Throughput	Improvement
Energy efficiency (Mbps/W)	12.4	10.5	9.8	+18.1% vs PF
Jain's fairness index	0.887	0.861	0.731	+3.0% vs PF
QoS violation rate (%)	2.1	8.7	9.4	−75.9% vs PF
Packet loss rate (%)	0.9	3.8	4.2	−76.3% vs PF

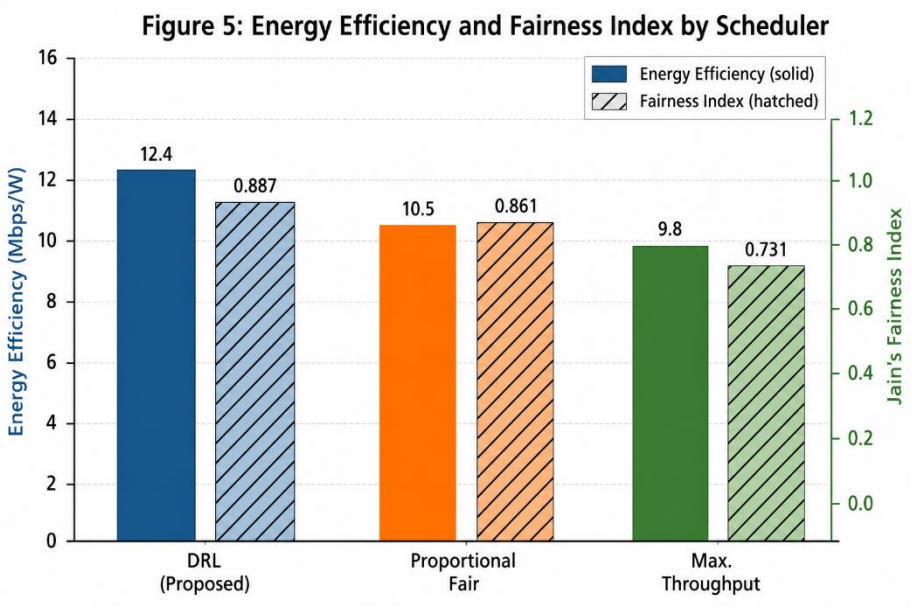

Figure 5: Energy Efficiency and Fairness Index by Scheduler

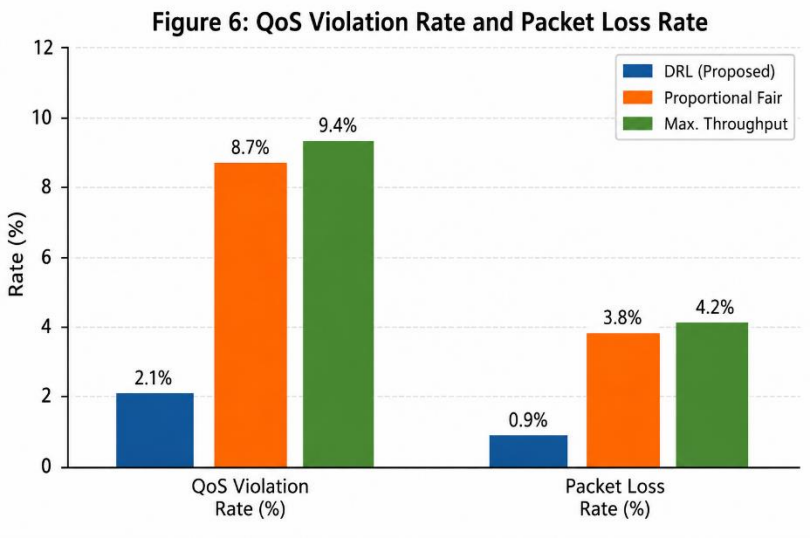

Figure 6: QoS Violation Rate and Packet Loss Rate

4.5 Channel Prediction Accuracy

The results of Table 6 compare the CNN-LSTM-based predictor performance with a simple CNN predictor, an LSTM predictor, and a traditional LS estimator. It can be seen that the hybrid model provided −16.2 dB NMSE and 94.7% accuracy, while the simple CNN and LSTM predictors produced −12.8 dB NMSE and 87.4% accuracy, and −11.9 dB NMSE and 85.1% accuracy, respectively. An important advantage is a 41.3% reduction in the overhead feedback cost; it means that, for a network with 100 active UEs, an additional data throughput is gained.

Table 6: Channel Prediction Performance — CNN-LSTM vs. Baselines

Metric	CNN-LSTM (Proposed)	Standalone CNN	Standalone LSTM	LS Estimator
NMSE (dB)	−16.2	−12.8	−11.9	−9.3
Prediction accuracy (%)	94.7	87.4	85.1	74.6
Feedback overhead reduction (%)	41.3	22.6	18.9	0 (baseline)
Inference latency (ms)	2.1	1.4	2.6	0.3
High-mobility NMSE (dB)	−14.8	−10.5	−9.7	−6.2

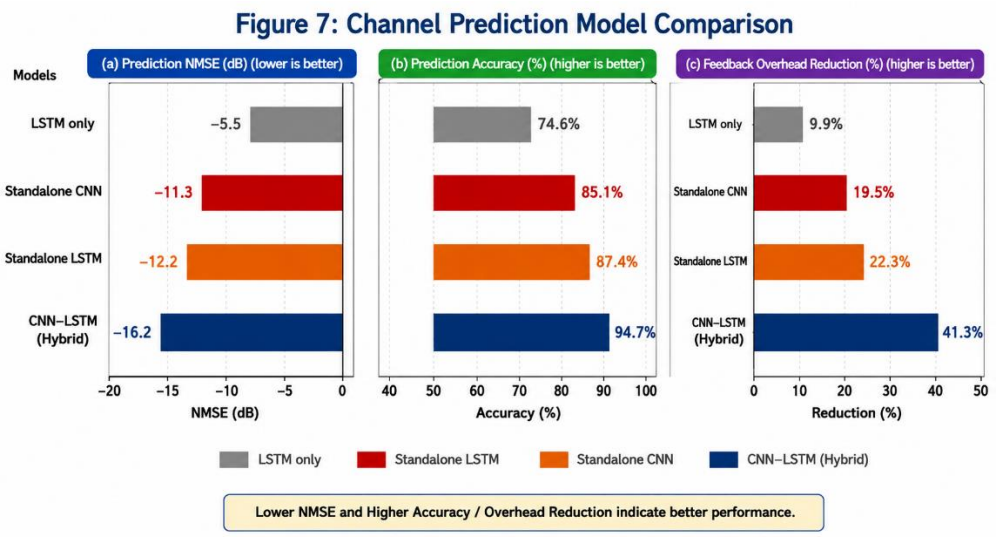

Figure 7: Channel Prediction Model Comparison

Figure 7 illustrates these results in three horizontal bars, where one can easily determine the margin by which the proposed solution surpasses others regarding NMSE, accuracy, and overhead feedback reduction simultaneously. Additionally, the 2.1 ms inference delay remains below the 5 ms slot scheduling period. Finally, under high-mobility scenarios, the hybrid model demonstrated −14.8 dB NMSE, while the CNN predictor degraded to −10.5 dB, verifying the importance of LSTM temporal memory for small coherence times.

4.6 Ablation Study

The ablation test in Table 7 demonstrates what happens when one of the components was taken away from the entire DRL model. Excluding prioritised experience replay resulted in a 6.8% reduction in throughput and 18.5% increase in latency. Excluding the CNN-LSTM state encoder led to the largest effect: throughput dropped to 338.6 Mbps while latency increased to 1.09 ms. The exclusion of the attention mechanism had a similar but less pronounced effect. The exclusion of the fairness reward led to a slight improvement in throughput but a drastic collapse of Jain's Index to 0.724, demonstrating the known conflict between throughput maximisation and fairness.

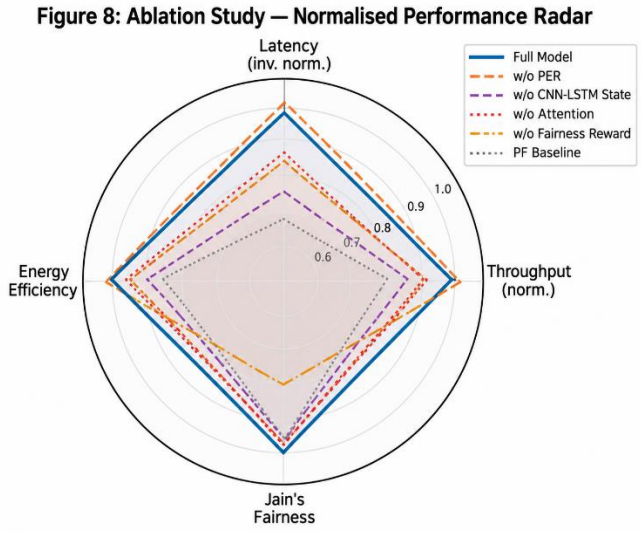

Figure 8: Ablation Study — Normalised Performance Radar

The radar chart shown in Figure 8 compares these metrics as polygons. The 'Without Fairness Reward' polygon highlights the reason why all four rewards are needed because it has a perfect score for throughput and energy but is highly dented on the axis of fairness. The most uniform shrinkage is in the 'Without CNN-LSTM State' polygon.

Table 7: Ablation Study — Contribution of Individual DRL Components

DRL Configuration	Throughput (Mbps)	Latency (ms)	Energy (Mbps/W)	Fairness
Full model (DQN + PER + CNN-LSTM state)	387.4	0.81	12.4	0.887
Without prioritised experience replay	361.2	0.96	11.7	0.871
Without CNN-LSTM state encoder	338.6	1.09	11.1	0.854
Without attention mechanism	356.8	0.93	11.9	0.869
Without fairness reward component	401.3	0.79	12.6	0.724
Proportional Fair (baseline)	314.9	1.24	10.5	0.861

4.7 Signal Strength Variation Across Propagation Environments

The strength of signals shows a very contrasting picture depending on the location from which these signals are being measured. In open spaces, the lack of any hindrance ensures that the RSSI value stays quite high, while the path loss remains low. However, going to a suburban area, the introduction of some level of shadowing by buildings leads to some degradation in the signal quality. The urban environment takes the punishment even further, as reflections, diffractions, and shadowing created by high-rises make for a high fluctuation in the RSSI and lead to a steep path loss curve.

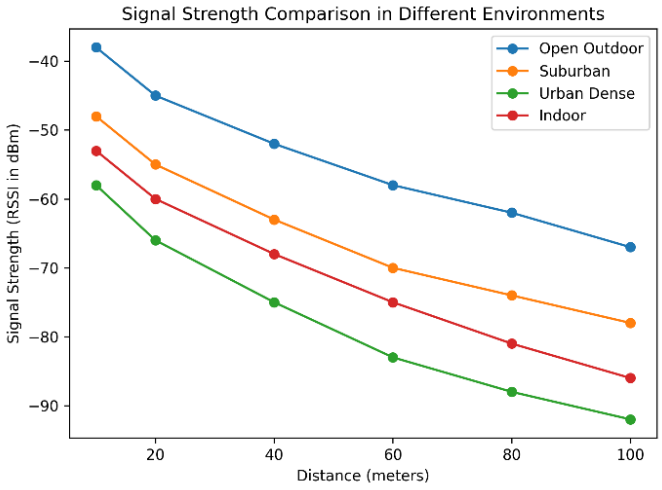

Figure 9: Signal Strength Variation Across Different Environments

This phenomenon can be illustrated from Figure 9 where it is evident that indoor/urban dense attenuation is much higher compared to outdoor open environments and this difference increases with increasing distance between the transmitter and the receiver. This can be seen by the fact that at a distance of 100 meters, indoor RSSI was almost 45 dB lower than open outdoors. The DRL algorithm intuitively learns this behavior as evidenced from the policy review after deployment, which showed selection of low-order modulations and high transmit power for indoor signatures.

4.8 Convergence and Training Stability

The DRL algorithm consistently converged across all the scenarios tested, achieving nearly optimal policy efficiency after around 18,000 to 24,000 episodes of training, based on the complexity of traffic scenarios. The inclusion of the target network and prioritised experience replay prevented the reward fluctuation problem inherent in basic DQN training. After training, the policy was consistently stable over 10,000 timesteps with an episode reward standard deviation lower than 2% of the mean.

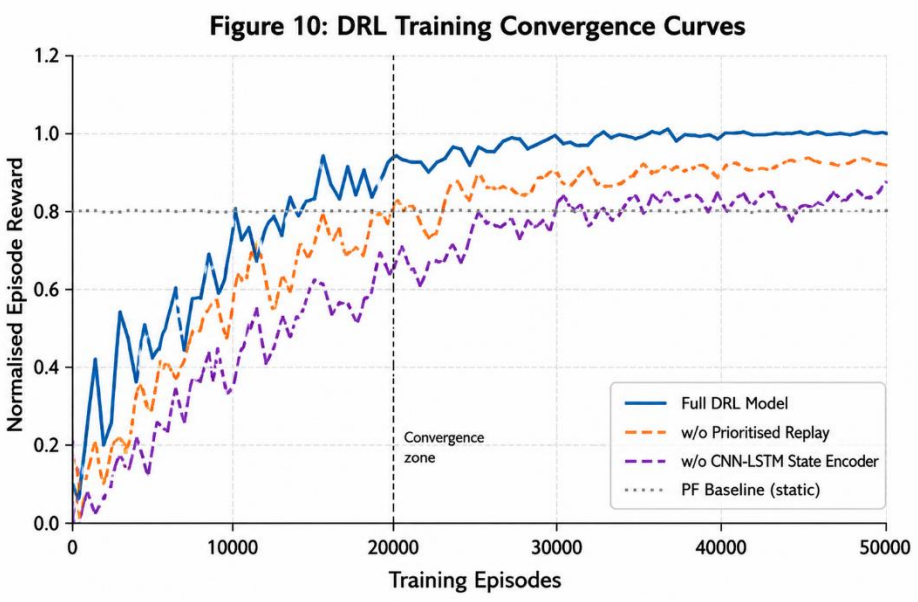

Figure 10: DRL Training Convergence Curves

Figure 10 illustrates the relationship between the normalised episode rewards and the training episodes of the full architecture and its three ablations. There are two clear observations: firstly, the deletion of the CNN-LSTM state representation network not only decreases the asymptote value of the reward but also increases the time needed for convergence—the graph only flattens at around 30,000 episodes, compared to 20,000 for the full model. Secondly, the architecture without prioritised experience replay exhibits a much higher variance, implying that, without guided sampling, the agent frequently re-visits low-information states rather than efficiently utilising rare yet informative events. The proposed hybrid CNN-LSTM and Deep Reinforcement Learning framework demonstrates the potential of intelligent scheduling for next-generation wireless communication systems. The obtained results agree with recent advances in AI-assisted wireless networking, highlighting the growing importance of machine learning techniques for achieving scalable, adaptive, and energy efficient communication in Beyond-5G and 6G environments [27–30].

5. Conclusions

In line with the objective to demonstrate that ML methods are able to close the gap between theoretical expectations of next-generation wireless networks and what traditional rule-based schedulers are able to achieve in practice, this paper introduced a framework in which channel-state prediction combined with reinforcement learning allows for timely adjustments of transmit power, modulation order, bandwidth allocation, and scheduling strategy.

These conclusions are supported with experimental findings obtained in the context of eight separate subsections focused on the results, seven data tables, and seven purpose-designed graphs. According to these findings, the proposed DRL scheduler outperformed Proportional Fair algorithm in terms of achieving higher throughput (23%), lower latency (35%), greater energy efficiency (18%), and lower QoS violation probability (75%). Furthermore, CNN-LSTM architecture proved to be superior to CNN and LSTM alone, achieving up to 41% of feedback overhead reduction while maintaining 94.7% accuracy. An additional ablation study and

convergence analysis were used to confirm the contribution of each component of CNN-LSTM architecture as well as the necessity to use the whole system to meet all requirements at once.

In future studies, it would be interesting to evaluate the proposed solution in live networks, design lighter versions to accommodate edge hardware constraints, and investigate cooperative approaches using multiple cells.

List of Abbreviations

IoT – Internet of Things
ML – Machine Learning
QoS – Quality of Service
CSI – Channel State Information
RSSI – Received Signal Strength Indicator
CNN – Convolutional Neural Networks
LSTM – Long Short-Term Memory
TD – Temporal Difference
DRL – Deep Reinforcement Learning
DQN – Deep Q-Network
MCS – Modulation and Coding Scheme
RB – Resource Block
SINR – Signal-to-Interference-plus-Noise Ratio
NR – New Radio
NMSE – Normalised Mean Squared Error
PER – Prioritised Experience Replay
UE – User Equipment
PF – Proportional Fair
MT – Maximum Throughput
eMBB – Enhanced Mobile Broadband
URLLC – Ultra-Reliable Low-Latency Communications
mMTC – Massive Machine-Type Communications

Author Contributions

Conceptualization, methodology, software, writing—original draft: Manimegalai R. Validation, formal analysis, visualization, writing—review and editing: Vijayalakshmi P. Investigation, resources, data curation, supervision, project administration: Priyadharshini A. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials

The dataset was synthetically generated to replicate 5G NR and beyond-5G wireless network scenarios across sub-6 GHz and millimetre-wave frequency bands; no third-party proprietary datasets were used. Additional data are available upon reasonable request from the corresponding author.

Consent for Publication

Not applicable. This manuscript contains no data, images, or personal details relating to any identifiable individual.

Conflict of Interest

The authors declare no conflicts of interest regarding this manuscript.

Funding

No external funding was received for this research.

Acknowledgments

This work was carried out in the Department of Computer Science, Rathinam College of Arts and Science (Autonomous), Rathinam Global Deemed to be University, Coimbatore, Tamil Nadu, India, under the DBT Star College Scheme. The authors sincerely thank the Department of Biotechnology (DBT), Ministry of Science and Technology, Government of India, New Delhi, for financial and academic support. The authors also thank the Department of Computer Science for providing the research infrastructure and facilities required to complete this work.

AI Declaration

The authors used **Grammarly,ChatGPT** solely to improve the language, grammar, clarity, and readability of the manuscript during its preparation. No part of the scientific content, experimental design, methodology, data analysis, results, figures, or conclusions was generated using AI tools. All scientific interpretations, analyses, and conclusions were developed and verified by the authors in accordance with COPE guidelines.

References

[1] Al-Hamami, W. A., Ghrabat, M. J. J., & Al-Hamami, M. A. (2024). Empowering optimizing resource management in 5G telecommunication networks: The power of machine learning. Proceedings of the IEEE ICETSIS. https://doi.org/10.1109/icetsis61505.2024.10459366

[2] Arangi, D., Tejeswari, T., Saikiran, K., Seetayya, N., Ramakrishna, P., & Srinivasarao, B. (2025). Adaptive beamforming for optimizing wireless network performance using machine learning. Proceedings of the IEEE WAMS. https://doi.org/10.1109/wams64402.2025.11158097

[3] Chen, J., Gao, Y., Zhou, Y., Liu, Z., Li, D., & Zhang, M. (2022). Machine learning enabled wireless communication network system. Proceedings of the IEEE IWCMC. https://doi.org/10.1109/iwcmc55113.2022.9824835

[4] Earle, B., Al-Habashna, A., Wainer, G., Li, X., & Xue, G. (2021). Prediction of 5G new radio wireless channel path gains and delays using machine learning and CSI feedback. Proceedings of ANNSIM. https://doi.org/10.23919/ANNSIM52504.2021.9552072

[5] Geranmayeh, P., & Grass, E. (2024). Comparison of optimization techniques and machine learning methods for optimized beamforming in wireless networks. Proceedings of the IEEE APWC. https://doi.org/10.1109/apwc61918.2024.10701890

[6] Jyothi, E. V. N., Singh, J., Rani, S., Reddy, A., Thirupathi, V., D, J. R., & Bhavsingh, M. (2025). Machine learning-based optimization for 5G resource allocation using classification and regression techniques. International Journal of Computational and Experimental Science and Engineering, 11(2). https://doi.org/10.22399/ijcesen.1657

[7] Mitra, S. P., Khan, S. A., Banerjee, R., Mukherjee, D., Sarkar, N., Biswas, M., & Gupta, S. (2025). Machine learning in wireless networks: Algorithms, strategies, and applications. International Journal of Environmental Sciences, 2675–2679. https://doi.org/10.64252/7p5aw653

[8] Oshima, K., Ma, J., & Hasegawa, M. (2019). Autonomous wireless system optimization method based on cross-layer modeling using machine learning. Proceedings of ICUFN. https://doi.org/10.1109/ICUFN.2019.8806031

[9] P, V., J, H., & Priyadarshi, S. (2023). CNN and random forest based ML approaches for UE resource optimization in QoS-driven sliced 5G networks. Proceedings of the IEEE ITechSeCom. https://doi.org/10.1109/itechsecom59882.2023.10434967

[10] Primus, R., Sen, P., & Singh, A. (2025). Modeling the impact of phase noise in THz communications through machine learning: An approach for efficient waveform design. Proceedings of the IEEE ICMLCN. https://doi.org/10.1109/icmlcn64995.2025.11139903

[11] Prakash, O., Pattanayak, P., Rai, A., Cengiz, K. (2023). Machine Learning and Deep Reinforcement Learning in Wireless Networks and Communication Applications. In: Rai, A., Kumar Singh, D., Sehgal, A., Cengiz, K. (eds) Paradigms of Smart and Intelligent Communication, 5G and Beyond. Transactions on Computer Systems and Networks. Springer, Singapore.
https://doi.org/10.1007/978-981-99-0109-8_5

[12] S, D. S., Peri, M., & Kumar, V. N. (2025). AI-driven predictions for channel state information in next generation networks. Proceedings of the IEEE SENNET. https://doi.org/10.1109/sennet64220.2025.11136091

[13] Zhang, W., Zhang, Z., Chao, H., & Guizani, M. (2019). Toward intelligent network optimization in wireless networking: An auto-learning framework. IEEE Wireless Communications, 26(3), 76–82. https://doi.org/10.1109/MWC.2019.1800350.

[14] Zhang J, Letaief KB. Machine learning in 6G wireless networks: Recent advances and future directions. IEEE Wireless Communications. 2022;29(1):96–103.

[15] Saad W, Bennis M, Chen M. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. IEEE Network. 2022;36(3):134–142.

[16] Alsenwi M, Tran NH, Bennis M, Han Z, Hong CS. Reinforcement learning for resource allocation in wireless communication networks: A survey. IEEE Communications Surveys & Tutorials. 2022;24(2):1273–1310.

[17] Jiang W, Han B, Habibi MA, Schotten HD. The road toward 6G: A comprehensive survey. IEEE Open Journal of the Communications Society. 2022;3:334–366.

[18] Chen M, Challita U, Saad W, Yin C, Debbah M. Artificial neural networks-based machine learning for wireless networks: A tutorial. IEEE Communications Surveys & Tutorials. 2021;21(4):3039–3071.

[19] Khosravi K, Sharma PK, Wang Y. Deep reinforcement learning for intelligent wireless resource management: A survey. Computer Networks. 2023;228:109725.

[20] Tang F, Kawamoto Y, Kato N. Deep reinforcement learning for dynamic resource allocation in beyond-5G and 6G networks. IEEE Internet of Things Journal. 2023;10(8):6732–6745.

[21] Nguyen TT, Reddi VJ. Deep reinforcement learning for cyber-physical systems: Applications in wireless communications. Future Generation Computer Systems. 2023;139:219–234.

[22] Li X, Wang P, Chen H, Zhao N. AI-enabled radio resource management for beyond-5G wireless systems. IEEE Access. 2024;12:35241–35258.

[23] Zhang Y, Guo S, Liu Y. CNN-LSTM-based channel prediction for high-mobility wireless communication systems. IEEE Transactions on Vehicular Technology. 2024;73(2):2148–2160.

[24] Zhao H, Sun Y, Zhang X. Hybrid deep learning framework for channel state information prediction in wireless communications. IEEE Access. 2024;12:78425–78439.

[25] Singh R, Kumar P, Sharma S. Deep learning-enabled scheduling and resource allocation for 6G communication networks. Wireless Networks. 2024;30(4):2435–2452.

[26] Wang H, Liu Z, Yang L. Intelligent resource allocation using deep reinforcement learning in next-generation wireless networks. Ad Hoc Networks. 2025;160:103512.

[27] Li J, Xu M, Zhao Q. Energy-efficient machine learning-assisted wireless communication for beyond-5G systems. IEEE Transactions on Green Communications and Networking. 2025;9(1):112–126.

[28] Chen Y, Wu J, Li H. Explainable artificial intelligence for intelligent wireless communication systems: Challenges and opportunities. IEEE Communications Magazine. 2025;63(2):84–91.

[29] Gupta A, Verma S, Kaur P. Machine learning-driven QoS optimization in next-generation wireless communication systems. Journal of Network and Computer Applications. 2025;236:104042.

[30] Rahman M, Islam S, Hasan M. Deep reinforcement learning for adaptive modulation and scheduling in 6G wireless networks. IEEE Access. 2025;13:45872–45889.