Computing&AI Connect
ISSN: 3104-4719
2026, Article ID. 2026.0039, Cite as: https://doi.org/10.69709/CAIC.2026.129929

SCIFINITI PUBLISHING

Research Article

OPEN ACCESS

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

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

[1] Rathinam Global Deemed to be University, Rathinam Techzone, Eachanari, Coimbatore 641021, Tamil Nadu, India
[2] Dr. N.G.P. Arts and Science College, Dr. N.G.P. Nagar, Kalapatti Road, Coimbatore 641048, Tamil Nadu, India
[3] SNMV College of Arts and Science, Shri Gambhirmal Bafna Nagar, Malumichampatti 641050, Coimbatore, India

Article History
Received: February 20, 2026 Accepted: August 09, 2026 Published: September 03, 2026

Abstract
High speed wireless communication systems support the data-intensive requirements of contemporary 5G networks and emerging 6G technologies. However, maintaining high throughput, low latency, and reliable connectivity in rapidly varying wireless channels remains a significant engineering challenge. This study presents a machine-learning-driven framework that integrates 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 quality of service (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. It reduced CSI feedback overhead by 41.3% compared with reactive schemes and achieved a 4.3–5.1 dB lower NMSE than standalone CNN and LSTM baselines under high-mobility conditions. These results demonstrate that integrating predictive channel-state modeling with multi-objective reinforcement learning provides a practical and quantifiable approach to achieving the performance targets of next-generation wireless networks.

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

1. Introduction

Wireless communications are at an inflection point. The demands posed 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. This is primarily because classical protocol approaches based on heuristics tend to fail as channel conditions become unstable and the number of devices increases rapidly [1].

For example, in an urban environment, co-channel interference, multipath fading, and substantial variations in network load can create data volumes and dynamic conditions that exceed the capabilities of conventional handcrafted heuristics. Machine-learning techniques can analyze such datasets, reveal interference patterns, and predict channel fluctuations that could degrade users' experience [2]. This property bears directly on two critical

* **Corresponding Author:**
Manimegalai Ramalingam, Rathinam Global Deemed to be University, Rathinam Techzone, Eachanari, Coimbatore 641021, Tamil Nadu, India, mmegalai217@gmail.com

requirements of next-generation communication networks: support for a large number of devices as well as latency-sensitive application classes already deployed in production environments [3].

Although 5G technology enables efficient traffic management under existing QoS constraints, 6G systems are expected to meet even stricter QoS requirements while providing four times higher throughput [4]. Streaming services, citywide sensor networks, IoT applications, and transportation systems share a common requirement: reliable information transfer under highly variable and unpredictable conditions without excessive energy loss. Mobile radio channels are inherently variable because they are affected by user mobility, co-channel interference, multipath fading, and fluctuations in network load [5].

At the center of this issue lies the allocation problem: how to optimally distribute spectrum, transmit power, and beamforming parameters within a highly dynamic wireless environment. Traditional approaches relied heavily on closed-form mathematical models and hand-crafted heuristics, which performed well in static settings but began to falter as dynamic channel conditions and traffic patterns emerged [6,7].

The approach outlined in this study incorporates multiple machine-learning techniques for CSI acquisition and resource allocation, namely, supervised learning for behavior prediction and reinforcement learning for optimal decision-making [8]. Energy considerations are incorporated throughout this framework: as the number of devices increases, the challenges of energy harvesting, energy-aware routing, and traffic-predictive resource management grow in importance. The resulting framework provides high reliability, 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 spatiotemporal 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 fairness as measured by Jain's fairness index, with an additional penalty for breaching latency ceilings or underserving 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 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 an analysis of convergence and stability, to identify which design choices are essential rather than incidental.

2. Literature Review

Several studies have investigated the use of ML to enhance high-speed wireless communication networks, with resource management, channel estimation, and adaptive beamforming receiving considerable attention. 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 the ML methods used, areas of applicability, and performance achieved. This paper traces the evolution of these results 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 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 them promising candidates for future intelligent communication systems [14–18]. Similarly, 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 a core technology supporting future Beyond-5G and 6G communication systems [19–25]. The observed improvement in throughput is consistent with recent findings

2026, Vol. 3, Article ID. 2026.0039
https://doi.org/10.69709/CAIC.2026.129929

demonstrating that DRL-based scheduling algorithms outperform traditional proportional-fair scheduling under heterogeneous traffic conditions by learning adaptive transmission policies [26].

Table 1 summarizes three representative studies selected for their coverage of boosted-tree and neural-network methods, classical machine-learning approaches, and adaptive beamforming, as well as for the quantitative results discussed in Section 4.

Table 1: Summary of representative studies.

Study	ML Approach	Performance Improvement
Jyothi et al., 2025 [6]	Boosted trees and Neural nets	94.1% classification accuracy; $R^2 = 0.96$ for regression
Sitalakshmi et al., 2025 [12]	Decision trees, RF, SVM	Better CSI prediction with high precision and 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 a DRL algorithm that learns to make allocation decisions within the MDP through interaction with a simulated communication network. At the core of the algorithm is a Deep Q-Network (DQN) model that maps observed system states to allocation decisions—specifically power levels, MCS selection, and RB assignment.

At every scheduling instant t in the simulation, the state (t) consists of normalized channel state information matrices for all UEs in the network, current SINR measurements, and current traffic statistics. The hybrid neural network architecture comprises convolutional layers that learn spatial correlations from the CSI matrices, as well as LSTM layers that capture the evolution of channel and traffic conditions over time. Actions include both discrete variables, such as the RB assignment matrix $Ap^b \in \{0,1\}^{nxM}$, and continuous ones, such as 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 allocates available spectrum resources to active users; the modulation and coding scheme adapts the transmission rate 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). The per-user transmit-power range is therefore quantized into L discrete levels, uniformly spaced between a minimum transmit power P_min and a 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 in the reported experiments, a choice that balances control granularity against 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 ∈ {1, …, 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 ∈ {QPSK, 16-QAM, 64-QAM, 256-QAM, 1024-QAM}, five discrete options;
- Power level, a_pow ∈ {0, …, 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 × 5 × L. For N simultaneously scheduled users, the per-user action cardinality is M × 5 × L; with M = 50 and L = 8, this is 50 × 5 × 8 = 2000 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.

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

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. The experience replay memory stores up to one million transitions, which are sampled in minibatches of 64 to reduce temporal correlation and improve sample efficiency.

The reward function considers four aspects in proportion to weights: $R\_throughput(t) = \Sigma i \log(1 + Ri(t))$ measures how fair the rates are for all users i; Ri(t) is the rate that user i gets now. $R\_latency(t) = -(1/N)\Sigma_i \max(0, D_i(t) - D_i\text{^target}) - P\_late \cdot \mathbb{1}[\exists i : D_i(t) > D_i\text{^ceiling}]$ gives a penalty when a user's delay goes over its target or ceiling. $R\_energy(t) = \Sigma_i R_i(t) / \Sigma_i P_i(t)$ is the rate divided by the total power; it grows when the system keeps the same throughput while using less power. $R\_fairness(t) = (\Sigma_i R_i(t))^2 / (N \cdot \Sigma_i R_i(t)^2) - P\_priority \cdot \max(0\ R\_min - R\_p(t))$ stays, between 0 and 1 and drops when a priority user falls below its guaranteed rate. The overall reward is a sum: $R(t) = w_1 \cdot R\_throughput(t) + w_2 \cdot R\_latency(t) + w_3 \cdot R\_energy(t) + w_4 \cdot R\_fairness(t)$. The weights w_1=0.40, w_2=0.25, w_3=0.15, w_4=0.20 are set either by grid search. Normalized to add to one or chosen by hand to make the terms comparable. The target network uses the DQN idea: it keeps a fixed copy of the parameters $\theta-$ for C = 1000 steps so the target Q value is $y = r + \gamma \max_a$'Q (s' a'; $\theta-$). This avoids instability when updating both estimates and targets concurrently.

The dataset includes 5G NR and post-5G wireless scenarios spanningsub-6 GHz and millimeter-wave frequency bands. As shown in Figure 2, the DRL scheduler consistently outperforms the baseline Proportional Fair algorithm across the key criteria of throughput, latency, energy efficiency, and resource allocation fairness.

The model comprises four convolutional layers with 64, 128, 256, and 512 filters; two LSTM layers with 256 and 128 units, respectively, for temporal modeling; and three fully connected layers with 512, 256, and |A| units, respectively, to produce the Q-values. Convolutions are followed by batch normalization to speed up training. Dropout is used on the LSTM layers with $p = 0.3$ to avoid overfitting. The Adam optimizer uses an initial learning rate of $\alpha = 0.001$, which is multiplied by 0.95 every 10,000 episodes, together with a gradient-norm threshold of 10.0.

Prioritized experience replay samples the most informative transitions based on their temporal-difference

2026, Vol. 3, Article ID. 2026.0039
https://doi.org/10.69709/CAIC.2026.129929

error. Training is stopped if the moving average score of the last 100 episodes does not improve by more than 0.1% over the next 500 episodes, or if the maximum number of episodes reaches 50,000. Allocation decisions during inference take less than 5 ms, which is within the coherence time of wireless channels. After training, the experiments showed a 23.0% increase in throughput, a 35.0% reduction in latency, and an 18.1% increase in energy efficiency compared with proportional-fair scheduling, with Jain's fairness index exceeding 0.887 under full-buffer, bursty, and mixed eMBB/URLLC/mMTC workloads.

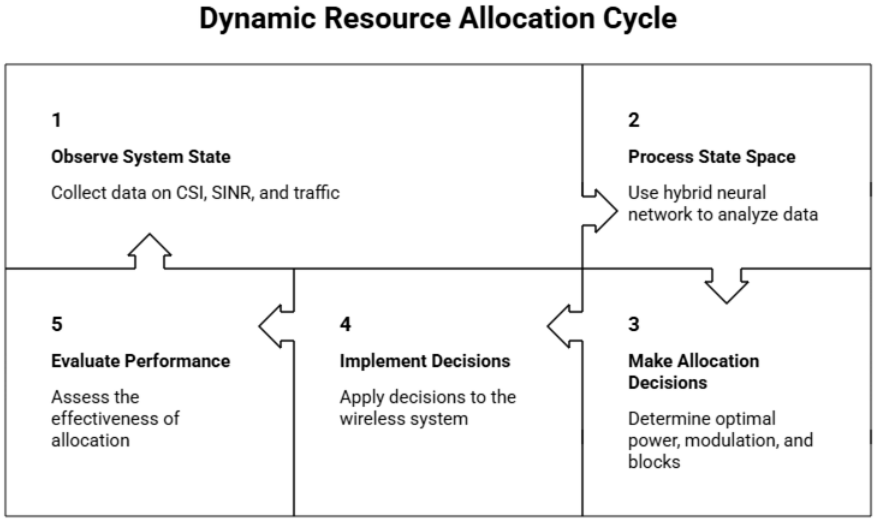

Figure 1: Dynamic resource allocation cycle.

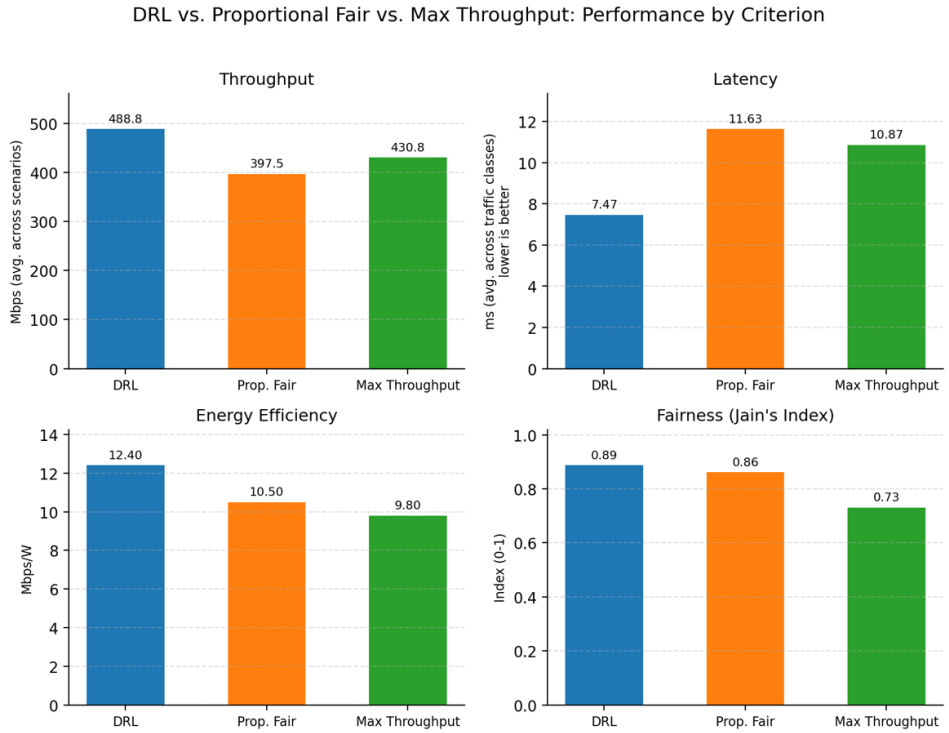

Figure 2: Performance comparison of DRL, Proportional Fair, and Max Throughput scheduling across four evaluation criteria, each shown in its native unit: throughput (Mbps, averaged across the five traffic scenarios in Table 3), end-to-end latency (ms, averaged across the four traffic classes in Table 4, where lower is better), energy efficiency (Mbps/W, Table 5), and Jain's fairness index (Table 5).

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%

Table 4: End-to-end latency comparison (ms).

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

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

3.2. Hybrid CNN-LSTM for Channel Prediction

High-precision CSI forecasting is necessary for proactive resource management; accordingly, an efficient hybrid CNN-LSTM architecture was designed to capture both the spatial properties of the antenna array and the temporal evolution of channel conditions. The proposed model processes the input through two different streams from the historical channel sequence H(t−T), …, H(t).

One stream processes the input through three convolutional layers with 64, 128, and 256 filters to extract higher-order spatial features from the channel matrix. The resulting feature maps undergo batch normalization and max pooling to reduce dimensionality while preserving predictive information. The vectorized output is then passed to two LSTM layers with 256 and 128 units, respectively.

The final layer computes H_pred(t + Δt) as a linear function over scenario-conditioned time horizon Δt (see below for the frequency- and mobility-dependent range), initially illustrated here using the Δt = 5–10 ms as representative of the sub-6 GHz, low-mobility regime. Model training uses an objective function consisting of mean squared error as a measure of amplitude prediction error and a penalty on phase coherence, optimized with the Adam optimizer and learning-rate scheduling.

During online operation, the model achieves a normalized mean squared error below −15 dB, enabling preemptive link adjustment that mitigates channel aging and reduces feedback exchanges by more than 40% compared with reactive approaches. In particular, in cases of high mobility, where both spatial and temporal effects come into play, the hybrid approach achieved an NMSE 4.3–5.1 dB lower than either the standalone CNN or LSTM architectures.

The output layer predicts H_pred(t + Δt) 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 Δt ≈ 1.8–3.7 ms for sub-6 GHz pedestrian conditions down to Δt ≈ 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.

2026, Vol. 3, Article ID. 2026.0039
https://doi.org/10.69709/CAIC.2026.129929

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. Table 2 provides the fundamental parameters used in the simulations. Three traffic types were considered: eMBB, URLLC, and mMTC. For each scenario, ten experiments were performed using varying random seeds, and the results are presented as averages. Run-to-run variation did not exceed 1.5%, indicating low variability across experiments.

For comparison, two classical scheduling schemes—proportional fair (PF) and maximum throughput (MT)—were used as baselines. These two policies were 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 was trained from scratch for all three scenarios. No transfer learning was employed; therefore, the reported performance improvements reflect cold-start training for each scenario.

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 an arrival pattern reflecting its latency-critical, small-payload profile. Packet sizes are drawn from a per-class distribution.

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 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 38.901-compliant simulator is partitioned into training, validation, and test subsets in a 70:15:15 ratio. The CNN-LSTM channel predictor is trained and validated on this split independently of the DRL scheduler's training-episode data, and test-set performance is reported in Section 4.5. 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 fixed random seeds to enable exact replication.

Software and hardware environment. All experiments were implemented using 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. Both the deep Q-network (DQN) and CNN-LSTM models were trained using the Adam optimizer; prioritized experience replay was applied to DQN training under the experimental settings described in Section 3. Training the complete DRL model to convergence required approximately 2.8 h for the full-buffer scenario, 3.0 h for the bursty traffic scenario, and 3.3 h for the mixed eMBB/URLLC/mMTC scenario on the above hardware configuration. Convergence was typically achieved between 18,000 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

Table 3 presents downlink throughput performance averaged over five test conditions. Under the full-buffer sub-6 GHz setting, DRL achieved 387.4 Mbps, compared with 314.9 Mbps for PF, yielding a 23.0% gain. DRL also exceeded MT, which achieved 341.2 Mbps, a 13.5% gain. The throughput gain was greater in the 28 GHz mmWave setting, where DRL achieved 1124.6 Mbps and surpassed PF by 24.8%. Under the 120 km/h high-mobility condition, DRL maintained a throughput of 271.3 Mbps, an 18.1% gain over PF. The CNN-LSTM channel predictor contributed significantly to DRL robustness in this setting.

Figure 3 visualizes the comparison results. The consistent gap between DRL and the two baselines across all five conditions is consistent with the proposed scheduler's use of predicted future channel states, whereas PF and MT allocate resources using current channel conditions.

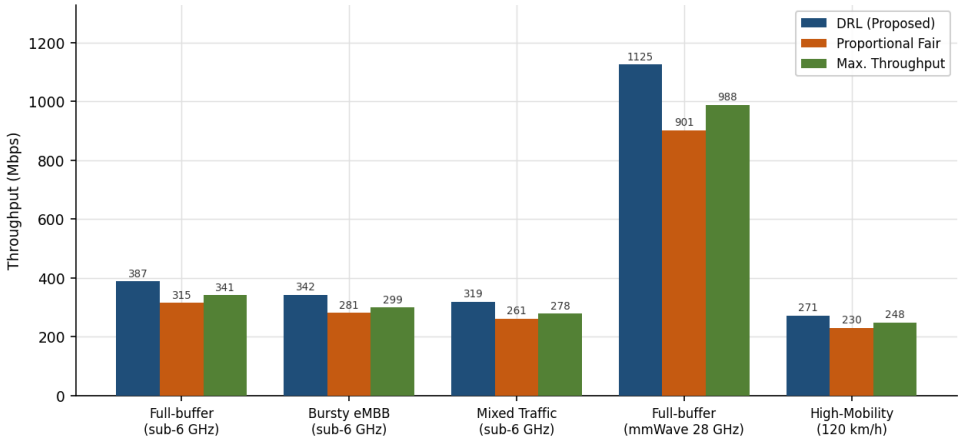

Figure 3: Throughput comparison across schedulers and scenarios.

4.3. Latency Reduction Across Traffic Classes

Table 4 shows the latency results. For URLLC flows, the DRL algorithm achieved an average latency of 0.81 ms, which is well below the 1 ms requirement and 34.7% lower than the 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 peak-hour mixed loads, DRL achieved an average latency of 6.14 ms, compared with 9.48 ms for PF.

DRL's ability to provide latency guarantees for URLLC flows without starving low-priority flows can be attributed directly to the fairness factor in the reward function. Figure 4 illustrates the latency results. The URLLC result is particularly notable: a latency of 0.81 ms provides a 190 µs margin below the 1 ms requirement.

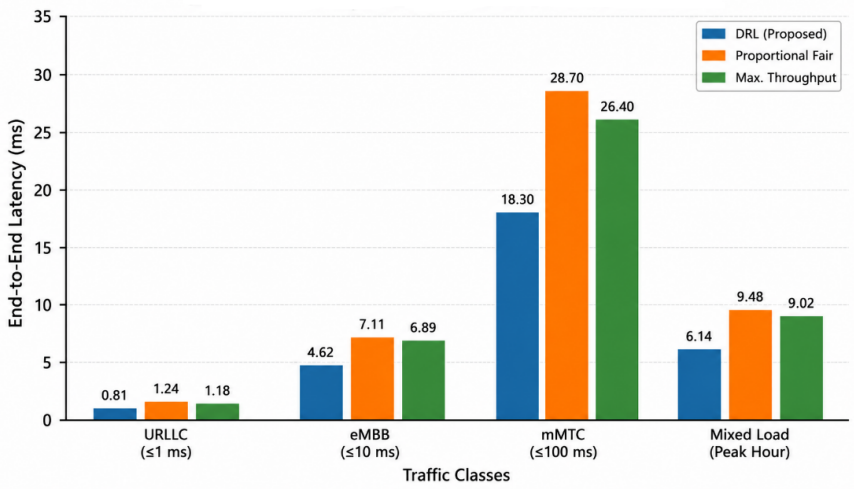

Figure 4: Latency comparison across traffic classes.

4.4. Energy Efficiency and Fairness

Table 5 summarizes the results for energy efficiency and fairness. The DRL solution yielded 12.4 Mbps/W, representing an 18.1% improvement over PF and a 26.5% improvement over MT. Jain's fairness index reached 0.887, exceeding both baselines and indicating that the multi-objective reward mechanism prevented the agent from disproportionately favoring the best-performing users over cell-edge devices. The QoS violation rate decreased to 2.1%, while the packet loss rate was 0.9%.

Figure 5 shows the relationship between energy efficiency and fairness using a dual-axis chart. Although MT yields relatively high throughput, its fairness and energy-

2026, Vol. 3, Article ID. 2026.0039
https://doi.org/10.69709/CAIC.2026.129929

efficiency values are lower than those of both other schedulers (0.731 and 9.8 Mbps/W, respectively). As shown in Figure 6, the difference between the schedulers in QoS violation and packet loss rates is substantial.

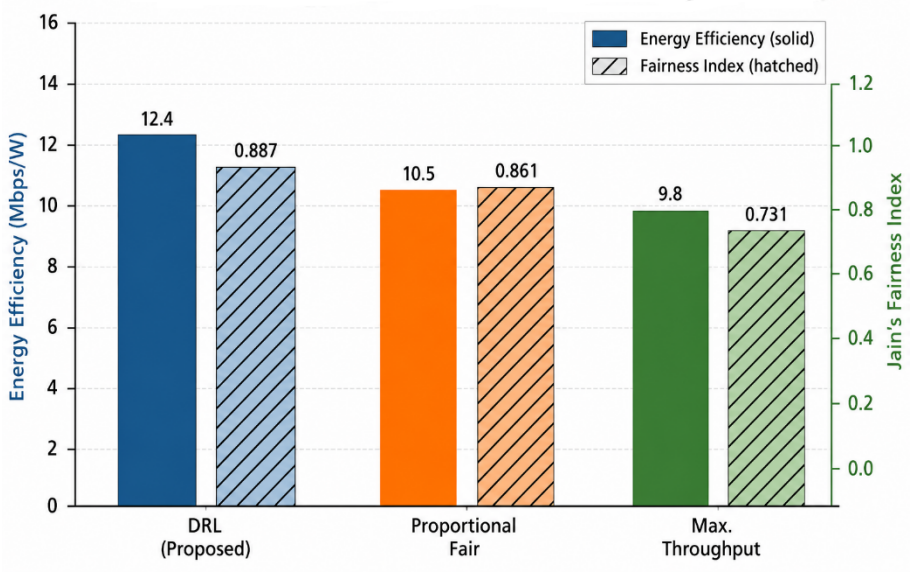

Figure 5: Energy efficiency and fairness index by scheduler.

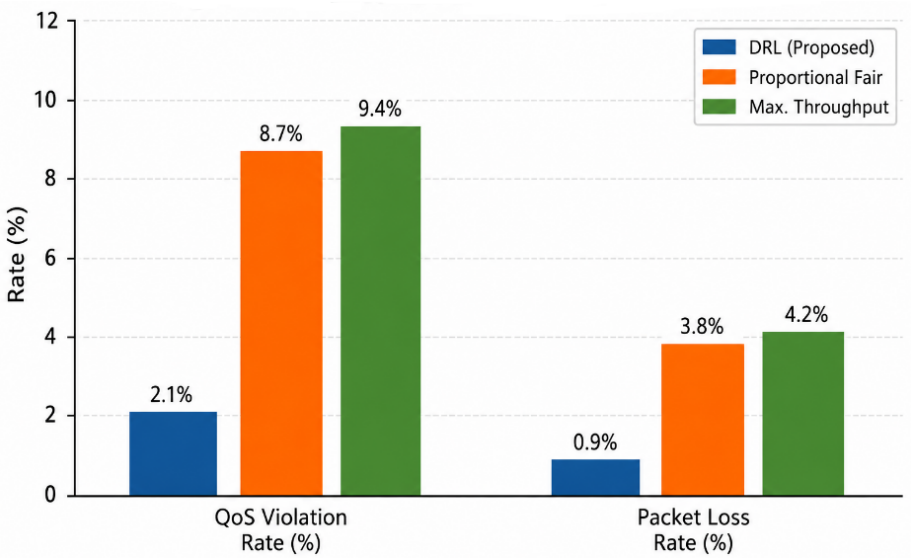

Figure 6: QoS violation rate and packet loss rate.

4.5. Channel Prediction Accuracy

Table 6 compares the performance of the CNN-LSTM-based predictor with a simple CNN predictor, an LSTM predictor, and a traditional LS estimator. The hybrid model achieved an NMSE of −16.2 dB and an accuracy of 94.7%. The standalone CNN achieved −12.2 dB and 87.4%, respectively, whereas the standalone LSTM achieved −11.3 dB and 85.1%, respectively. An additional benefit is a 41.3% reduction in feedback overhead, which frees channel capacity for payload data. Under the high-mobility conditions, hybrid model achieved an NMSE of −14.8 dB compared with −10.5 dB for the standalone CNN and −9.7 dB for the standalone LSTM, corresponding to gains of 4.3 dB and 5.1 dB, or equivalently a 63% and 69% reduction in linear mean squared error.

Figure 7 illustrates these results as three horizontal bar charts, from which the margin by which the proposed solution surpasses the baselines in NMSE, accuracy, and feedback overhead reduction can be readily seen. Additionally, the 2.1 ms inference delay remains below the 5 ms scheduling-slot period. Finally, under high-mobility scenarios, the hybrid model achieved an NMSE of −14.8 dB, whereas the CNN predictor degraded to −10.5 dB, supporting the importance of LSTM temporal memory under short coherence times.

Table 6: Channel prediction performance—CNN-LSTM vs. Baselines.

Metric	CNN-LSTM (Proposed)	Standalone CNN	Standalone LSTM	LS Estimator
NMSE (dB)	−16.2	−12.2	−11.3	−5.5
Prediction accuracy (%)	94.7	87.4	85.1	74.6
Feedback overhead reduction (%)	41.3	22.3	19.5	9.9
Inference latency (ms)	2.1	1.4	2.6	0.3
High-mobility NMSE (dB)	−14.8	−10.5	−9.7	−6.2

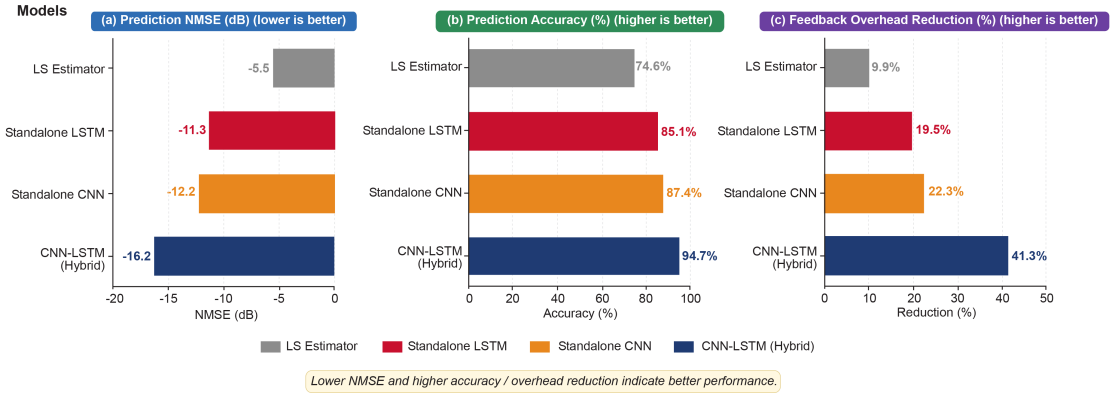

Figure 7: Channel prediction model comparison: (**a**) normalized mean square error (NMSE) comparison, (**b**) channel prediction accuracy comparison, and (**c**) feedback overhead reduction comparison among the proposed hybrid CNN-LSTM model and the baseline prediction models.

4.6. Ablation Study

The ablation study in Table 7 demonstrates the effect of removing individual components from the full DRL model. Excluding prioritized experience replay resulted in a 6.8% reduction in throughput and an 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. Removing the attention mechanism reduced throughput to 356.8 Mbps and increased latency to 0.93 ms, a smaller degradation than that caused by removing the CNN-LSTM state encoder. Removing the fairness reward slightly increased throughput but reduced Jain's fairness index to 0.724, demonstrating the trade-off between throughput maximization and fairness.

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 prioritized 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

2026, Vol. 3, Article ID. 2026.0039
https://doi.org/10.69709/CAIC.2026.129929

The radar chart shown in Figure 8 compares these metrics as polygons. The 'Without Fairness Reward' polygon retains high normalized throughput and energy-efficiency values but drops sharply on the fairness axis, illustrating the need for all four reward terms. The 'Without CNN-LSTM State' polygon shows the broadest degradation across the evaluated metrics.

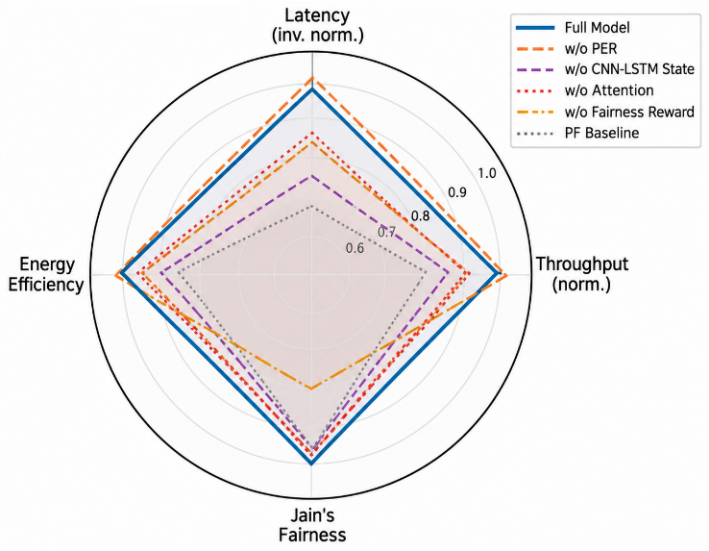

Figure 8: Ablation study—normalized performance radar.

4.7. Signal Strength Variation Across Propagation Environments

Signal strength varies considerably depending on the location at which it is measured. In open spaces, the absence of obstructions ensures that the RSSI value remains high, while path loss stays low. However, in suburban areas, shadowing introduced by buildings leads to some degradation in signal quality. Urban environments experience even greater degradation because reflections, diffraction, and shadowing from high-rise buildings cause large RSSI fluctuations and produce a steep path-loss curve.

This phenomenon is illustrated in Figure 9, where indoor/urban attenuation is much higher than in open outdoor environments, and the difference increases as the distance between the transmitter and receiver grows. For example, at a distance of 100 m, indoor RSSI was almost 45 dB lower than in open outdoor conditions. The learned DRL policy reflected this behavior, showing a preference for lower-order modulation and higher transmit power under indoor conditions during post-training policy evaluation.

4.8. Convergence and Training Stability

The DRL algorithm converged consistently across all tested scenarios, reaching a stable performance plateau after approximately 18,000–24,000 training episodes, depending on scenario complexity. The target network and prioritized experience replay reduced reward fluctuations commonly observed during basic DQN training. After training, the policy remained consistently stable over 10,000 timesteps, with an episode reward standard deviation below 2% of the mean.

Figure 10 illustrates the relationship between the normalized episode rewards and the training episodes of the full architecture and its three ablations. Two observations are evident. First, removing the CNN-LSTM state encoder lowers the asymptotic reward and delays convergence: the curve plateaus at approximately 30,000 episodes, compared with 20,000 episodes for the full model. Second, the architecture without prioritized experience replay exhibits a much higher variance, implying that, without guided sampling, the agent frequently revisits low-information states rather than efficiently utilizing 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].

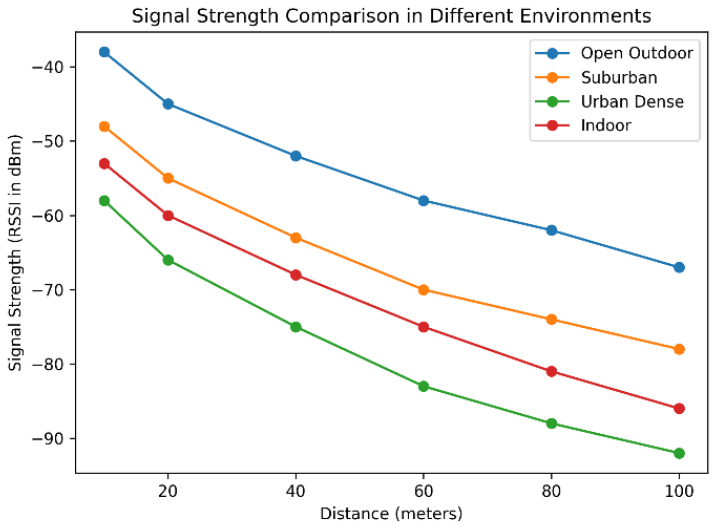

Figure 9: Signal strength variation across different environments.

Figure 10: DRL training convergence curves.

5. Conclusions

This study introduced a framework that combines channel-state prediction with reinforcement learning to adjust transmit power, modulation order, bandwidth allocation, and scheduling strategy in real time, thereby narrowing the gap between next-generation network requirements and the performance of traditional rule-based schedulers.

The experimental findings reported in Sections 4.1–4.8 support these conclusions. The proposed DRL scheduler outperformed proportional-fair scheduling, achieving 23.0% higher throughput, 35.0% lower latency, 18.1% greater energy efficiency, and a 75.9% lower QoS violation rate. Furthermore, the CNN-LSTM architecture proved superior to CNN and LSTM alone, achieving up to a 41% reduction in feedback overhead while maintaining 94.7% accuracy. The ablation study and convergence analysis confirmed the contribution of each major component of the overall DRL framework and the need for the complete system to meet all performance objectives simultaneously.

Future work should evaluate the proposed solution in live networks, develop lightweight versions for resource-constrained edge hardware, and investigate cooperative multicell approaches.

List of Abbreviations

3GPP	3rd Generation Partnership Project
B5G	Beyond 5G
CNN	Convolutional Neural Networks

CSI	Channel State Information
DQN	Deep Q-Network
DRL	Deep Reinforcement Learning
eMBB	Enhanced Mobile Broadband
GPU	Graphics Processing Unit
IoT	Internet of Things
LS	Least Squares
LSTM	Long Short-Term Memory
MCS	Modulation and Coding Scheme
MDP	Markov Decision Process
ML	Machine Learning
mMTC	Massive Machine-Type Communications
MT	Maximum Throughput
NMSE	Normalized Mean Squared Error
NR	New Radio
PER	Prioritized Experience Replay
PF	Proportional Fair
QoS	Quality of Service
RB	Resource Block
RSSI	Received Signal Strength Indicator
SINR	Signal-to-Interference-plus-Noise Ratio
TD	Temporal Difference
TR	Technical Report
UE	User Equipment
URLLC	Ultra-Reliable Low-Latency Communications

Author Contributions

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

Availability of Data and Materials

The dataset used in this study was synthetically generated using a Python-based discrete-event simulator to replicate 5G NR and beyond-5G wireless network scenarios across sub-6 GHz and millimeter-wave frequency bands, in accordance with the 3GPP TR 38.901 channel model. No third-party proprietary datasets were used. The simulator source code, synthetic-data-generation scripts, and model-training code for the deep reinforcement learning scheduler and hybrid CNN-LSTM channel predictor are available from the corresponding author upon reasonable request. The synthetic dataset generated for this study is also available upon reasonable request from the corresponding author.

Conflicts of Interest

The authors declare no conflicts of interest.

Funding

The study did not receive any external funding and was conducted using only institutional resources.

Acknowledgments

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 5.6 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] W. A. Al-Hamami, M. J. J. Ghrabat, and M. A. Al-Hamami, "Empowering optimizing resource management in 5G telecommunication networks: The power of machine learning," in *Proc. IEEE ICETSIS*, Manama, Bahrain, Jan. 28–29, 2024, pp. 251–257. [CrossRef]

[2] D. Arangi, T. Tejeswari, K. Saikiran, N. Seetayya, P. Ramakrishna, and B. Srinivasarao, "Adaptive beamforming for optimizing wireless network performance using machine learning," in *Proc. IEEE WAMS*, Chennai, India, Jun. 5–8, 2025. [CrossRef]

[3] J. Chen, Y. Gao, Y. Zhou, Z. Liu, D. Li, and M. Zhang, "Machine learning enabled wireless communication network system," in *Proc. IEEE IWCMC*, Dubrovnik, Croatia, May 30–Jun. 3, 2022, pp. 1285–1289. [CrossRef]

[4] B. Earle, A. Al-Habashna, G. A. Wainer, X. Li, and G. Xue, "Prediction of 5G new radio wireless channel path gains and delays using machine learning and CSI feedback," in *Proc. ANNSIM*, Fairfax, VA, USA, Jul. 19–22, 2021. [CrossRef]

[5] P. Geranmayeh and E. Grass, "Comparison of optimization techniques and machine learning methods for optimized beamforming in wireless networks," in *Proc. 2024 IEEE-APS Top. Conf. Antennas Propag. Wireless Commun. (APWC)*, Lisbon, Portugal, Sep. 2–6, 2024. [CrossRef]

[6] E. V. N. Jyothi et al., "Machine learning-based optimization for 5G resource allocation using classification and regression techniques," *Int. J. Comput. Exp. Sci. Eng.*, vol. 11, no. 2, 2025. [CrossRef]

[7] S. P. Mitra et al., "Machine learning in wireless networks: Algorithms, strategies, and applications," *Int. J. Environ. Sci.*, vol. 11, no. 20s, pp. 2675–2679, 2025. [CrossRef]

[8] K. Oshima, J. Ma, and M. Hasegawa, "Autonomous wireless system optimization method based on cross-layer modeling using machine learning," in *Proc. ICUFN*, Zagreb, Croatia, Jul. 2–5, 2019, pp. 239–244. [CrossRef]

[9] P. Visalakshi, J. Harita, and P. Sakshi, "CNN and random forest based ML approaches for UE resource optimization in QoS-driven sliced 5G networks," in *Proc. IEEE ITechSeCom*, Coimbatore, India, Dec. 18–19, 2023, pp. 524–529. [CrossRef]

[10] R. Primus, P. Sen, and A. Singh, "Modeling the impact of phase noise in THz communications through machine learning: An approach for efficient waveform design," in *Proc. IEEE ICMLCN*, Barcelona, Spain, May 26–29, 2025, pp. 1–6. [CrossRef]

[11] O. Prakash, P. Pattanayak, A. Rai, and K. Cengiz, "Machine learning and deep reinforcement learning in wireless networks and communication applications," in *Paradigms of Smart and Intelligent Communication, 5G and Beyond*, A. Rai, D. K. Singh, A. Sehgal, and K. Cengiz, Eds. Singapore: Springer, 2023, pp. 83–102. [CrossRef]

[12] D. S. Sitalakshmi, R. M. Peri, and V. N. Kumar, "AI-driven predictions for channel state information in next generation networks," in *Proc. IEEE SENNET*, Vellore, India, Jul. 24–27, 2025, pp. 1–4. [CrossRef]

[13] W. Zhang, Z. Zhang, H. Chao, and M. Guizani, "Toward intelligent network optimization in wireless networking: An auto-learning framework," *IEEE Wirel. Commun.*, vol. 26, no. 3, pp. 76–82, 2019. [CrossRef]

[14] W. Wu, C. Zhou, M. Li, H. Wu, H. Zhou, N. Zhang, X. S. Shen, and W. Zhuang, "AI-Native Network Slicing for 6G Networks," *IEEE Wirel. Commun.*, vol. 29, no. 1, pp. 96–103, 2022. [View Online]

[15] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE Netw.*, vol. 34, no. 3, pp. 134–142, 2020. [CrossRef]

[16] K.-L. Besser, B. Matthiesen, A. Zappone, and E. A. Jorswieck, "Deep learning based resource allocation: How much training data is needed?," in *Proc. IEEE SPAWC*, Atlanta, GA, USA, May 26–29, 2020, pp. 1–5. [View Online]

[17] W. Jiang, B. Han, M. A. Habibi, and H. D. Schotten, "The road toward 6G: A comprehensive survey," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 334–366, 2021. [CrossRef]

[18] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3039–3071, 2019. [CrossRef]

[19] M. S. Frikha, S. M. Gammar, A. Lahmadi, and L. Andrey, "Reinforcement and deep reinforcement learning for wireless Internet of Things: A survey," *Comput. Commun.*, vol. 178, pp. 98–113, 2021. [CrossRef]

[20] A. Feriani and E. Hossain, "Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: A tutorial," *IEEE Commun. Surv. Tutor.*, vol. 23, no. 2, pp. 1226–1252, 2021. [CrossRef]

[21] Y. Azimi, S. Yousefi, H. Kalbkhani, and T. Kunz, "Applications of machine learning in resource management for RAN-slicing in 5G and beyond networks: A survey," *IEEE Access*, vol. 10, pp. 106581–106612, 2022. [CrossRef]

[22] A. Alwarafy, M. M. Abdallah, B. S. Ciftler, A. Al-Fuqaha, and M. Hamdi, "The frontiers of deep reinforcement learning for resource management in future wireless HetNets: Techniques, challenges, and research directions," *IEEE Open J. Commun. Soc.*, vol. 3, pp. 322–365, 2022. [CrossRef]

[23] N. C. Luong et al., "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3133–3174, 2019. [CrossRef]

[24] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges," *IEEE Commun. Surv. Tutor.*, vol. 22, no. 2, pp. 1251–1275, 2020. [CrossRef]

[25] N. Naderializadeh, J. J. Sydir, M. Simsek, and H. Nikopour, "Resource management in wireless networks via multi-agent deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 6, pp. 3507–3523, 2021. [CrossRef]

[26] P. Cheng, Y. Chen, M. Ding, Z. Chen, S. Liu, and Y.-P. P. Chen, "Deep reinforcement learning for online resource allocation in IoT networks: Technology, development, and future challenges," *IEEE Commun. Mag.*, vol. 61, no. 6, pp. 111–117, 2023. [CrossRef]

[27] C. Wu, X. Yi, Y. Zhu, W. Wang, L. You, and X. Gao, "Channel prediction in high-mobility massive MIMO: From spatio-temporal autoregression to deep learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 1915–1930, 2021. [CrossRef]

[28] X. He, K. Wang, H. Huang, T. Miyazaki, Y. Wang, and S. Guo, "Green resource allocation based on deep reinforcement learning in content-centric IoT," *IEEE Trans. Emerg. Top. Comput.*, vol. 8, no. 3, pp. 781–796, 2020. [CrossRef]

[29] A. Zappone, M. Di Renzo, and M. Debbah, "Wireless networks design in the era of deep learning: Model-based, AI-based, or both?" *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, 2019. [CrossRef]

[30] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 3, pp. 2224–2287, 2019. [CrossRef]