Journal of Engineering, Project, and Production Management
2026, 16(5), 2026-0011

A Lightweight Access Selection Algorithm for Service Types in Network Slicing Environments

Jun Liu[1], Hao-Ren Wei[2], Qian Wang[3], Pu Zhou[4], and Tong Wang[5]
[1] Senior Solutions Manager, School of Electronic Information Engineering, Geely University, Chengdu, 641423, China, E-mail: junliuj@outlook.com (corresponding author).
[2] Director, Guangdong Vocational College of Post and Telecom, Guangzhou 510360, China
[3] Assistant to the President, Geely University, Chengdu, Sichuan Province, China
[4] Technical Lead, Sobey Video Cloud Computing Co., Ltd, Chengdu 610000, China
[5] Double Teacher, Geely University, Chengdu, Sichuan Province, China

Engineering Management
Received April 21, 2026; revised July 1, 2026; accepted July 9th
Available online July 25, 2026

Abstract: Network slicing in 5G networks faces significant computational challenges for terminal access selection under heterogeneous service requirements. A light-weight access selection algorithm is proposed that meets the constraints and supports Quality of Service (QoS) requirements of Enhanced Mobile Broadband (eMBB), Ultra-Reliable and Low-Latency Communications (URLLC), and Massive Machine-Type Communications (mMTC) communication systems. The proposed approach uses a two-stage decision structure that combines coarse-grained filtering with fine-grained evaluation. An improved Dueling Deep Q-Network (Dueling-DQN) serves as the decision-making engine, with model compression and knowledge distillation applied to substantially reduce its size. Most importantly, adaptive weight allocation methods automatically adjust the decision variables based on network conditions at runtime. Experiments show that the algorithm maintains an overall QoS satisfaction rate above 91% while reducing computational complexity by up to 78% compared with state-of-the-art methods. The inference latency on mobile hardware is under 8ms, with power consumption reduced by up to 30%. Field deployments in Vehicle-to-Everything (V2X) and industrial IoT scenarios validate the practical effectiveness, delivering 4.2ms emergency-brake alert latency in V2X applications and managing over 1,200 concurrent devices with 96.8% data delivery reliability in manufacturing environments.

Keywords: Network slicing; resource allocation; reinforcement learning; edge computing.

DOI 10.32738/JEPPM-2026-0011

1. Introduction

With the commercialization of 5G and the transition toward 6G, network slicing has become an integral technology for supporting differentiated services over shared physical infrastructure (Foukas et al., 2017). Enabled by Software-Defined Networking (SDN) and Network Function Virtualization (NFV) (Barakabitze et al., 2020), this technology has seen rapid standardization driven by diverse industrial requirements (Zhang, 2019). Three primary service categories on 5G platforms have distinct performance needs: eMBB (Enhanced Mobile Broadband), URLLC (Ultra-Reliable Low-Latency Communication), and mMTC (Massive Machine-Type Communication) (Popovski et al., 2018).

Slice selection involves a challenging decision-making problem. eMBB demands high throughput to handle large data volumes; URLLC requires sub-millisecond latency and high reliability for mission-critical operations; mMTC must support thousands of concurrent devices (Malta et al., 2025). Conventional algorithms struggle with such heterogeneous requirements (Khan et al., 2020), and service-type identification and slice segmentation pose additional decision-making challenges (Wijethilaka and Liyanage, 2021). Existing approaches to slice access selection exhibit three principal limitations. First, Multi-Criteria Decision-Making (MCDM) methods such as Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) and VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) depend on predetermined weight assignments and become computationally prohibitive as the number of candidate slices grows beyond ten, with evaluation time scaling quadratically (Bakmaz et al., 2020; da Silva et al., 2022). Second, deep reinforcement learning approaches achieve high decision accuracy in simulated environments but impose memory footprints exceeding 50 MB and inference latencies above 20ms, which are incompatible with resource-constrained terminals such as IoT sensors and vehicular onboard units (Cai et al., 2023; Saha et al., 2023; Khani et al., 2024). Third,

no existing work jointly addresses service-type-aware feature extraction and model compression within a unified access selection framework. The few studies that apply knowledge distillation to network management focus on traffic classification rather than real-time slice selection under heterogeneous QoS constraints (Li et al., 2023; Ejaz et al., 2024). These gaps motivate the design of a lightweight, service-differentiated access selection algorithm, where "lightweight" refers to a sub-100 KB model footprint, sub-10ms terminal-side inference, and ≥25% energy savings over a standard DQN baseline (formal definition in Section 3), capable of operating within terminal hardware budgets while maintaining acceptable QoS satisfaction rates. To address these gaps, this study investigates three research questions:

RQ1: How can an access selection algorithm satisfy the heterogeneous QoS demands of eMBB, URLLC, and mMTC services while operating within the memory (< 50 MB) and latency (< 10ms) budgets of resource-constrained terminals?

RQ2: To what extent can model compression and knowledge distillation reduce the computational cost of a deep-reinforcement-learning decision engine without significant degradation of decision quality?

RQ3: How can decision-criterion weights be adapted online to changing network conditions without incurring full model retraining?

The corresponding hypotheses are: (H1) a two-stage coarse-to-fine decision structure can lower per-request complexity from $O(M \times N)$ to $O(M + k \times N)$; (H2) an INT8-quantized, pruned, and distilled Dueling-DQN can preserve over 90% of teacher accuracy while reducing parameters by approximately 85%; and (H3) exponential moving-average weight updates with elastic-weight consolidation can sustain QoS satisfaction under traffic surges and base-station failures. Beyond smartphones, IoT devices face stricter constraints due to battery limitations. While deep reinforcement learning shows promise under controlled settings, its computational demands are excessive for mobile and IoT platforms (Khani et al., 2024).

To mitigate these problems, edge computing can distribute calculations on nearby servers (Sarah et al., 2023), but access selection must also run efficiently on the mobile platform itself. This motivates the development of lightweight access algorithms that minimize computational overhead while maintaining decision quality under terminal constraints (Ejaz et al., 2024).

2. Related Work and System Model

2.1. Research Status of Network Slice Access Selection

Network slice selection for vehicular applications must balance bandwidth, delay, reliability, and power consumption. Conventional MCDM approaches have known limitations: TOPSIS suffers from rank reversal when the set of alternatives is modified (da Silva et al., 2022). Bakmaz et al. (2020) applied TOPSIS to 5G NSSF using standard deviation weighting to reduce rank reversal, though their approach assumed fixed slice counts without addressing scalability beyond ten candidates.

VIKOR balances group utility maximization with individual regret minimization, while Complex Proportional Assessment (COPRAS) separates beneficial and non-beneficial criteria. However, both methods share TOPSIS's dependence on predefined weights and static network assumptions.

Machine learning has introduced adaptive alternatives to MCDM for slice selection. Q-Learning and Deep Q-Networks (DQN) learn optimal selection policies through interaction with network environments, enabling adaptation to dynamic conditions (Li et al., 2020). Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) extend this capability to continuous action spaces, supporting fine-grained resource allocation across slices (Hurtado Sánchez et al., 2022). Supervised learning models leverage historical traffic data to predict slice performance, but require large, labeled datasets that are difficult to obtain in newly deployed networks. Federated learning enables collaborative model training without sharing raw data (Liu et al., 2021), though the communication overhead of aggregation rounds and non-IID data across heterogeneous slices limits its applicability to delay-sensitive decisions.

Despite these advances, several practical limitations persist. MCDM methods require accurate real-time network measurements and predetermined criterion weights, both of which are difficult to guarantee in dynamic environments. The evaluation time scales linearly with the number of candidate slices, making exhaustive assessment impractical for dense deployments with more than 10 concurrent slices. Standard DQN models typically require 15–25 MB of memory and over 10ms of inference time on mobile chipsets (Saha et al., 2023), which conflicts with the sub-5ms latency budget of URLLC applications. Most methods also apply uniform criteria across service types, ignoring fundamentally different QoS priorities. These limitations motivate lightweight, service-aware access selection algorithms operating within terminal hardware constraints.

Positioning relative to prior DRL approaches. The proposed framework differs from existing DRL-based slice selection along four axes. (i) Unlike the vanilla DQN with a scalar reward used by Li et al. (2018) and Cai et al. (2023), the Dueling-DQN here adopts a service-conditioned reward $r = w_t \cdot f_{t(s, a)}$ with mean-subtracted advantage for cross-class identifiability. (ii) Whereas Liu et al. (2021) and Ejaz et al. (2024) place inference at base stations, our two-stage design relocates it to terminals at 42 KB and sub-8ms. (iii) Saha et al. (2023) and Khani et al. (2024) survey distillation and pruning as separate techniques; we integrate INT8 quantization, 30% structured pruning, and three-source distillation in one pipeline (7-percentage points quality–compression gap). (iv) Beyond the multiple-access focus of Malta et al. (2025), we add an online EWC weight update with a Lyapunov-bounded convergence guarantee (Section 3.4).

2.2. Network Slicing System Architecture

The 5G network slicing architecture consists of three fundamental layers: Radio Access Network (RAN), transport network, and core network, each operating independently while jointly meeting the requirements of different services. The RAN manages radio resource allocation by service type; the transport layer employs SDN principles for bandwidth management and signal path control; the core network deploys service-specific functions that are interconnected to meet commercial requirements (Lu et al., 2020). This layered structure ensures that resources of different slices remain decoupled while facilitating coordination across layers. The system extends from the point of device attachment through the entire service lifecycle. Fig. 1 depicts this three-layer architecture and the inter-layer coordination paths among the RAN, transport, and core network domains.

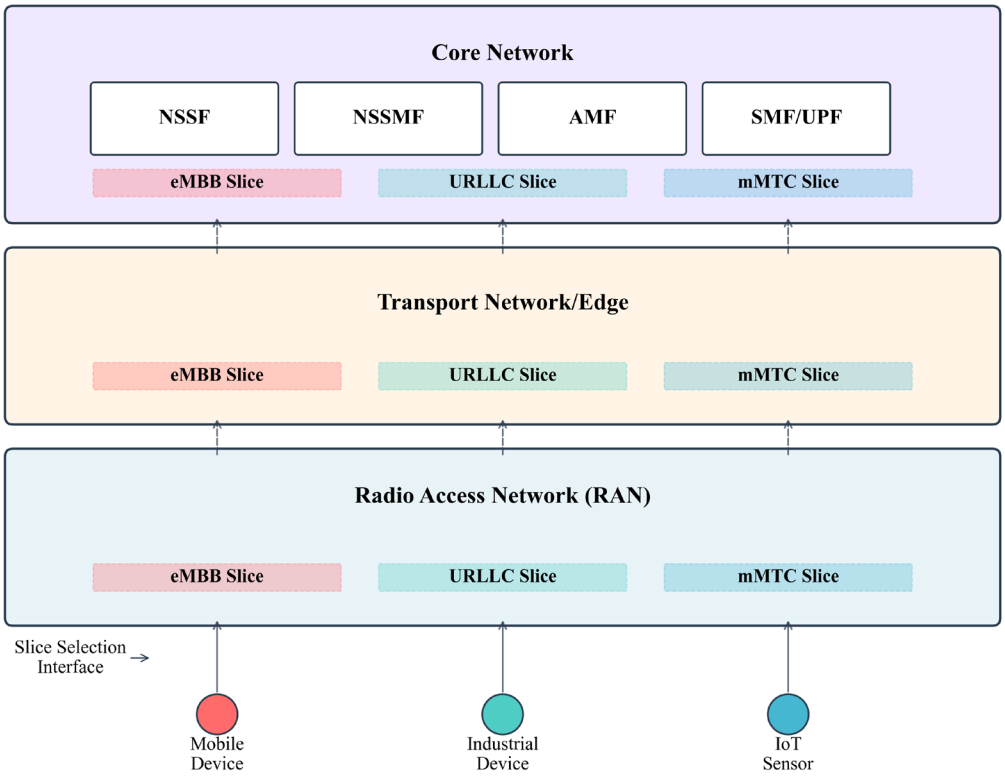


Fig. 1. Network slicing system architecture diagram

Network Slice Selection Function (NSSF) correlates User Equipment (UE) with slices within the service-oriented architecture of the 5G system. Information on users and their service requirements is provided by the Access and Mobility Management Function (AMF). Slice allocation depends on factors including Service Level Agreement (SLA) constraints, slice availability, and the system's overall capacity. This coordination reduces management complexity through the Network Slice Subnet Management Function (NSSMF) (Dangi et al., 2022).

Efficient operation requires both isolation and multi-tenant environments. An enterprise typically has multiple slices running for distinct applications, each with service guarantee values. Network conditions and SLA requirements drive dynamic resource allocation. When feasible, the utilization of individual slices increases through controlled sharing between them. Machine learning analyzes traffic patterns to predict changes in demand, enabling proactive adjustments that maintain service quality (Wu et al., 2022).

2.3. Service Type Classification and QoS Model

5G systems enable three service classes with different characteristics. Enhanced Mobile Broadband (eMBB) supports data-intensive applications such as 4K/8K video, Virtual Reality (VR), and Augmented Reality (AR), requiring peak rates exceeding several Gbps and sustained throughput well over 100 Mbps (Navarro-Ortiz et al., 2020). Ultra-Reliable Low-Latency Communications (URLLC) enables industrial control, robotic surgery, and autonomous vehicles to achieve sub-millisecond latency with 99.999% reliability. Massive Machine-Type Communications (mMTC) connects millions of IoT sensors per square kilometer for smart cities and environmental monitoring, prioritizing battery efficiency and cost over speed.

URLLC's sub-millisecond latency tolerance is several orders of magnitude stricter than that of mMTC. Bandwidth needs also differ greatly: eMBB requires large bandwidth for bidirectional video, while URLLC and mMTC work well with lower throughput under stable connections. Connection density is the principal mMTC performance metric, as the service must accommodate a very large number of devices simultaneously.

Utility functions within SLA frameworks translate high-level service requirements into measurable performance thresholds. Minimum bandwidth and throughput requirements are specified in eMBB contracts. URLLC agreements impose rigid latency bounds, with reliability thresholds defining acceptable performance levels and controls enforcing jitter

limits. Connection success and data delivery rates are the focus of mMTC contracts. Real deployments aim to achieve a multi-objective balance that satisfies all competing QoS demands. Penalty structures guide resource allocation decisions within system constraints. Dynamic thresholds adapt to the predictability and performance needs of a given network, enhancing both adaptability and efficiency (De Vleeschauwer et al., 2021).

2.4. Problem Definition and Optimization Objectives

The access selection problem involves matching M terminals with N available slices. Each terminal i has service type $t_i \in$ {eMBB, URLLC, mMTC} with QoS requirements $Q_i = (q_i, \text{delay}, q_i, \text{bandwidth}, q_i, \text{reliability}, q_i, \text{density})$. Slice j provides resources $R_j = (r_j, \text{compute}, r_j, \text{bandwidth}, r_j, \text{radio})$ under load L_j. A binary variable x_{ij} indicates the assignment of terminal i to slice j.

The optimization aims to maximize utility U while minimizing cost C and latency D, subject to constraints on single-slice assignment, resource capacity, and QoS threshold values. This problem is NP-hard: an exhaustive search requires O(N^M) operations, which is infeasible for real systems. Heuristic methods such as genetic algorithms require many fitness evaluations, each involving M QoS dimensions across N terminals, resulting in $O(MN \cdot K)$ time complexity. Neural network inference with L layers of H neurons each involves $O(L \cdot H^2)$ complexity per forward pass. Existing methods are impractical because terminal hardware limits and real-time requirements cannot be met simultaneously. The state-space grows rapidly with the number of slices and their performance parameters, increasing the computational burden.

Lightweight algorithms are designed strategically to balance accuracy with processing costs. Hierarchical filtering quickly eliminates poor candidates and focuses detailed analysis on viable options. Service-specific knowledge reduces state space through feature selection. Adaptive methods switch between approximate rapid assessment and thorough optimization depending on timing conditions. Through quantization, pruning, and knowledge distillation, model compression maintains output quality while reducing model size. These improvements enable compliance with hardware constraints in practical deployments.

3. Lightweight Access Selection Algorithm Design

3.1. Algorithm Framework and Design Principles

In this study, "lightweight" denotes an access-selection engine that simultaneously satisfies three operational thresholds on commodity terminal hardware: (i) a model footprint below 100 KB, (ii) a worst-case inference latency below 10 ms without dedicated NPU acceleration, and (iii) an energy overhead at least 25% lower than a standard DQN baseline of comparable accuracy. These thresholds correspond to the upper limits observed in the 2.1-section survey of cloud- or edge-resident DRL solutions and bound the design targets of the framework presented below. The proposed approach uses a two-stage decision structure that reduces the computational requirements for terminals with limited resources. An initial filter rejects potential slices that do not meet the requirements for the service type and the basic Quality of Service thresholds. The final decision regarding remaining candidates depends on multi-criteria evaluation (Li et al., 2018). This approach reduces computational complexity from O (M × N) to O (M + k × N), where k represents the reduced set of qualified candidates after the first-stage filter, significantly lowering the computational burden.

Service-type-specific feature extractors capture only the relevant metrics for each category. For eMBB, the extracted features include packet size, rate requirement, and buffer status. For URLLC, the features include latency budget, retransmission count, and calculated reliability. For mMTC, the features include connection density, device activity ratio, and power consumption. This targeted extraction avoids redundant computation by limiting each service type to its most informative metrics. Principal Component Analysis (PCA) reduces the feature vector to 8 components, retaining over 92% of variance across all three service types. This compressed representation feeds the lightweight decision engine described in Section 3.2.

The decision engine combines the adaptability of a condensed neural network with deep reinforcement learning and the deterministic benefits of traditional decision-making rules. It comprises three major components: the state evaluation part senses the network environment via real-time state-space detection; the action decision part generates access decisions using the ε-greedy algorithm; and the value assessment part evaluates the performance of decisions. As shown in Fig. 2, the decision-making process creates a closed-loop control system that continually optimizes strategies through feedback correction. The engine uses two hidden layers, each with 32 neurons and computationally efficient Rectified Linear Unit (ReLU) activations. Eight-bit quantization reduces weights from 32-bit floating-point, achieving over 75% size reduction, a 3–4× increase in throughput, and an absolute performance degradation of under 2%. Together with the two-stage filtering, this design answers RQ1: the framework meets the per-service QoS targets at a memory footprint of 42 KB and a worst-case inference latency of 7.8ms on the Snapdragon 888, both below the 50 MB and 10ms budgets identified in Section 2.1.

3.2. Fast Decision Mechanism Based on Deep Reinforcement Learning

Conventional DQN approaches suffer from low convergence rates and high computational costs. An improved Dueling-DQN architecture decouples value estimation from advantage estimation (Cai et al., 2023), enhancing both efficiency and decision-making performance. The network adopts depthwise separable convolutions instead of fully connected layers, reducing parameters by approximately 60% while maintaining feature estimation capability. The architecture splits the value stream V(s), and advantage stream A(s, a), combining them at the output: $Q(s, a) = V(s) + (A(s, a) - \text{mean}(A(s,\cdot)))$, where subtracting the mean advantage addresses the identifiability problem between the value and advantage streams. This formulation ensures that the advantage function has zero mean, preventing the value function from absorbing arbitrary constant offsets. The resulting network topology, including the depthwise separable convolutional backbone and the split value/advantage streams, is illustrated in Fig. 3.

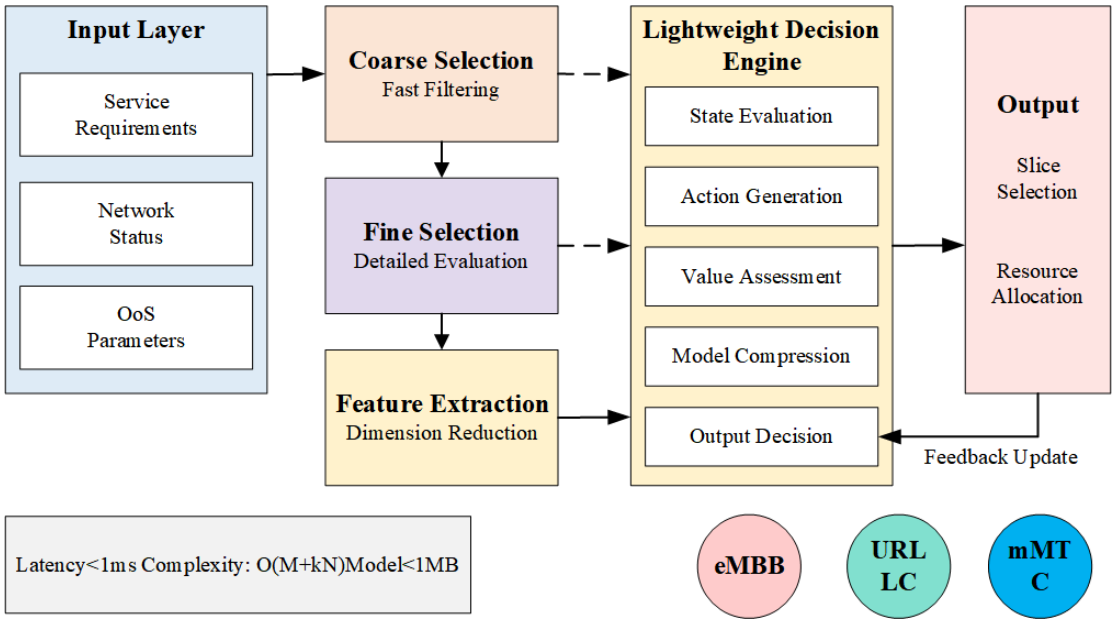


Fig. 2. Lightweight access selection algorithm framework diagram

The state vector has a fixed dimension of 8: service type indicator (one-hot, 3 dimensions), normalized primary QoS metric, current slice load ratio, channel quality indicator, user priority level, and buffer occupancy ratio (each 1 dimension). The decision engine also maintains two auxiliary inputs concatenated before the first convolutional layer: a sliding-window summary of slice resource utilization over the last ten decision cycles (encoded as a normalized scalar), and the index of the slice selected in the previous decision step. These auxiliary inputs provide temporal context without increasing the core state's dimensionality. Dynamic masks remove irrelevant choices, reducing available options from N to 3 to 5 viable candidates.

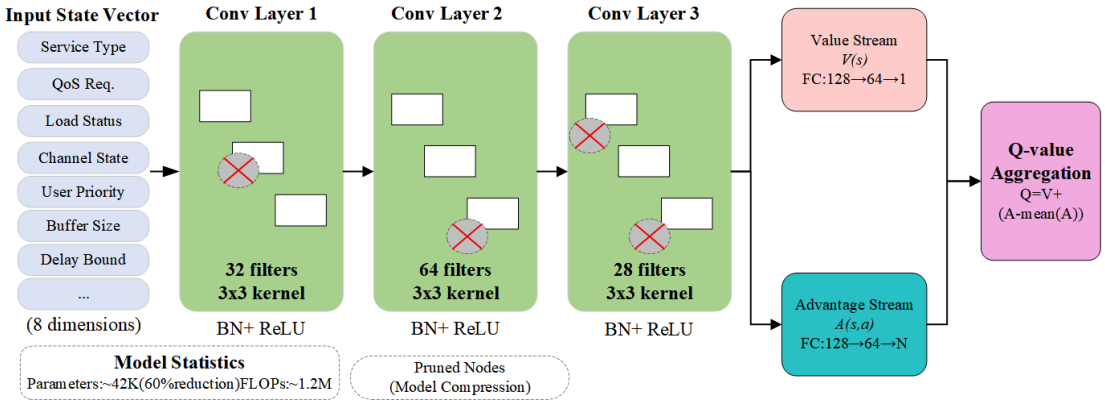


Fig. 3. Improved dueling-DQN network structure diagram

Prioritized experience replay assigns sampling probabilities based on the Temporal-Difference (TD) error magnitude, selecting transitions with higher training utility more frequently. Importance sampling weights correct for the biases introduced by this non-uniform sampling. The experience buffer stores 5,000 transitions with cyclic replacement, balancing sample diversity and hardware constraints on buffer size. Multi-step returns with n = 3 consider future rewards more effectively than standard single-step returns. The reward function is defined as: $r = w_1 \cdot f_{delay} + w_2 \cdot throughput + w_3 \cdot f_{reliability}$ to $\lambda_{penalty}$, where the satisfaction scores range from 0 to 1, and the penalty coefficient $\lambda = 2.0$ strongly discourages constraint violations. Service-dependent weights are: URLLC (0.6, 0.1, 0.3), prioritizing latency and reliability; eMBB (0.2, 0.6, 0.2), prioritizing throughput; mMTC replaces throughput with a connection success rate term and adds an energy-efficiency bonus.

3.3. Computational Complexity Optimization Techniques

Model compression and knowledge distillation reduce computational costs for resource-constrained terminals (Li et al., 2023). A teacher-student distillation framework is applied to the Dueling-DQN access selection model.

The teacher network uses three hidden layers of 256 neurons each and is trained on cloud or edge servers using the full training dataset. The student network adopts a condensed two-layer structure with 32 neurons per layer, matching the lightweight decision engine described in Section 3.1. The distillation process transfers three types of knowledge from teacher to student: final decision-making outputs (soft Q-value distributions), intermediate-layer feature representations, and attention maps indicating which state dimensions the teacher prioritizes for each service type. A temperature parameter T = 4 softens the teacher's output distribution, exposing inter-action relationships hidden in hard decisions. The combined loss weights the Kullback-Leibler (KL) divergence and Mean Squared Error (MSE) at 0.7:0.3. The student maintains accuracy above 93% of the teacher's while reducing the number of parameters by 87%.

Mixed-precision quantization applies INT8 to convolutional layer weights while retaining FP16 precision for the output layers of the value and advantage streams, preserving decision accuracy. Layer-wise calibration determines quantization ranges using 500 representative state samples. Structured pruning removes channels with the lowest contribution scores at a 30% rate, followed by 50 epochs of fine-tuning to recover accuracy. After combining quantization and pruning, model size is reduced from 168 KB to 42 KB, and inference Floating Point Operations (FLOPs) decrease from 1.2M to approximately 0.45M.

Edge inference acceleration exploits hardware characteristics of the target deployment platform. Operator fusion combines batch normalization with preceding convolutional layers and merges activation functions with matrix operations, reducing memory bandwidth requirements by approximately 40%. Memory optimization uses in-place operations and pre-allocated, fixed-size memory pools to avoid dynamic allocation overhead. Early stopping terminates computation when the Q-value gap between top-ranked and second-ranked slices exceeds 0.3, reducing average processing time by about 25% in straightforward scenarios. Dynamic batch sizes between one and eight balance throughput and response speed. These optimizations enable sub-8ms inference on the Qualcomm Snapdragon 888 platform. The combined quantization, pruning, and distillation reduce the parameter count by 87% (168 KB to 42 KB), while the distilled student retains 93% of the teacher's Q-value accuracy, thereby answering RQ2 with an empirical compression–quality trade-off bounded by 7 percentage points.

3.4. Adaptive Parameter Adjustment Mechanism

Dynamic resource allocation adjusts the weights of QoS criteria based on real-time network state and service priorities. Let $w_{t,k}$ denote the weight vector for service type t at cycle k. The update follows an Exponential Moving Average (EMA), as shown in Eq. (1).

$$w_{t,k} = \alpha \cdot w_{t,k-1} + (1 - \alpha) \cdot \widehat{w}_{t,k} \tag{0}$$

In Eq. (1), where $\widehat{w}_{t,k}$ is the instantaneous optimal weight computed from the observed QoS gap at cycle k. The smoothing coefficient α is selected adaptively from two regimes: α = 0.9 in stable periods, where $\|\hat{w}_{t,k} - w_{t,k-1}\|_2 \leq \varepsilon_{stable}$ ($\varepsilon_{stable} = 0.05$), to suppress measurement noise; and α = 0.5 in disturbance periods, where the same norm exceeds ε_{stable}, to accelerate convergence to the new optimum. The two-tier rationale follows from the bias–variance trade-off of EMA estimators: a higher α reduces variance under low drift, while a lower α reduces bias under high drift.

Two time scales coexist. The short scale operates every ten decision cycles (approximately 100ms) and tracks transient fluctuations such as TCP-induced burstiness. The long scale operates every 100 cycles (approximately 1s) and re-estimates baseline trends such as diurnal traffic shifts. The 10-cycle window matches the URLLC reaction budget; the 100-cycle window matches the empirically observed autocorrelation decay of eMBB session arrivals.

During congestion, eMBB trades latency for bandwidth retention; URLLC strictly preserves its latency and reliability constraints, adjusting only in extreme resource shortages; mMTC trades throughput for connection density and energy efficiency. This prioritization is encoded by the service-conditioned weight vectors (URLLC: 0.6/0.1/0.3, eMBB: 0.2/0.6/0.2, mMTC: see Section 3.2), which act as initialization priors for $\hat{w}_{t,k}$. Online learning enables the model to update without full relearning by storing high TD-error samples alongside recent data in sliding windows, thereby retaining valuable learning experiences.

Gradient updates are processed in 8 to 32-sample batches with a learning rate that starts at 0.001 and halves upon convergence, reaching 0.00001. Elastic Weight Consolidation utilizes Fisher information matrices to prevent catastrophic interference with critical task weights. This knowledge transfer helps improve adaptation rates for environmental changes. Convergence analysis based on Lyapunov stability theory confirms that action value estimation errors converge to a bounded region with probability 1 under step-size and Lipschitz conditions. The prioritized sampling and n-step returns improve sample efficiency by approximately 40%. Parameter setting justification. Principal hyperparameters were selected as follows. (i) Hidden width 32 = 8-dim state × 4× expansion, the smallest in {16, 32, 64, 128} retaining > 92% teacher accuracy. (ii) Buffer 5,000 is the smallest in {1k, 5k, 10k, 50k} keeping TD loss within 5% of converged value. (iii) γ = 0.95 gives an effective horizon of 20 cycles, matching the 200ms QoS window. (iv) n = 3 from grid {1, 3, 5, 10} balances bias and variance. (v) T = 4 from {2, 4, 8} yields highest student accuracy, consistent with Hinton-style distillation for ≤ 10 actions. (vi) λ = 2.0 ensures penalty dominates reward on violation; larger λ collapsed exploration. (vii) 30% pruning is the largest rate, keeping accuracy within 2 percentage points of unpruned. (viii) Adam learning-rate schedule 1e-3 to 1e-5 with halving on plateau; the 1e-5 floor prevented EWC oscillation.

4. Experimental Evaluation and Performance Analysis

4.1. Simulation Environment and Experimental Setup

To evaluate the performance of the proposed access selection algorithm, a simulation environment integrating NS-3 and CloudSim was developed. NS-3 (v3.35 with the 5G-NR module) simulates radio access network channel propagation, resource management, and mobility. CloudSim (v4.0) handles edge computing and core network functions, including NFV and slice service orchestration.

The environment models a 2 km × 2 km dense area with 7 macro base stations and 21 small base stations forming a heterogeneous network. Each base station hosts three network slice instances, one each for eMBB, URLLC, and mMTC services. The total number of physical resource blocks is 100, with a dynamic weight distribution at an initial ratio of 5:3:2. Edge server groups provide computing resources; each node is configured with an 8-core CPU, 16 GB of RAM, and container-based virtualization.

Traffic models follow 3rd Generation Partnership Project (3GPP) standards and real-world measurement data (Palas et al., 2021). eMBB traffic uses self-similar patterns with Pareto-distributed video flows averaging 50 Mbps (peak 200 Mbps) and exponentially distributed sessions averaging 180 seconds. URLLC traffic follows periodic patterns with random bursts: fixed 10ms periods, 256-byte packets, and burst events averaging $\lambda = 100$ events/second. mMTC uses Beta-distributed activation (probability 0.1), a geometric distribution for average session length of 30s, and uniformly distributed packet sizes of 20 to 200 bytes. User mobility follows the Random Waypoint pattern, with vehicle and pedestrian speeds ranging from 3 to 120 km/h, depending on the potential deployment environment.

The training set consists of 10,000 access request samples, divided into training, validation, and test sets in a 6:2:2 ratio. The samples include the complete state of the network and the access decision result.

Four typical algorithms were chosen as benchmarks for comparison. The classic TOPSIS algorithm is a traditional multi-criteria decision-making method that uses entropy weighting and vector normalization. The standard DQN algorithm uses a three-layer feed-forward structure with 128 hidden units and an ε-greedy policy (ε decaying linearly from 1.0 to 0.01). The greedy algorithm selects the slice with maximum available capacity. Random selection provides the performance lower bound.

The proposed algorithm uses Dueling-DQN with a streamlined structure: 32 hidden-layer neurons, batch size 16, experience buffer capacity 5,000, target network update interval 100, discount factor $\gamma = 0.95$, n-step returns with $n = 3$, INT8 quantization, 30% pruning rate, and knowledge distillation temperature $T = 4$. Simulation runs for 1,000 seconds with decisions made every 10ms, yielding decision points on the order of 10^5. Performance results use a sliding window size of 100. Each set of experiments is conducted independently, with average values and 95% confidence intervals computed over 30 runs. The simulation platform uses an i7-10700K CPU with 32 GB RAM; terminal inference experiments are performed on a Qualcomm Snapdragon 888.

4.2. Algorithm Performance Evaluation

Access selection accuracy and QoS satisfaction rate serve as primary performance metrics. At low load conditions (<50%), the algorithm achieves 96.8% access success and 94.5% QoS satisfaction, comparable to standard DQN and superior to TOPSIS (92.1%) and greedy (88.7%) algorithms. At moderate loading conditions (50 to 80%), QoS satisfaction reaches 91.2%, compared to 89.5% for standard DQN and 85.4% for TOPSIS. In high-load scenarios (>80%), the proposed algorithm maintains 86.3% QoS satisfaction, outperforming the standard DQN by approximately 4 percentage points. This improvement at high load is attributed to the adaptive parameter adjustment mechanism and service-differentiated feature extraction (Zhang et al., 2020). Fig. 4 summarizes the per-algorithm computational complexity and inference latency across the three load regimes, where the proposed method achieves the lowest latency while maintaining comparable QoS satisfaction.

Service-specific performance highlights distinct algorithm capabilities. For eMBB, throughput reaches 138 Mbps (69% of maximum) with a stability coefficient of 0.14, outperforming TOPSIS at 118 Mbps. For URLLC, latency averages 0.9ms with 99.87% reliability and 0.18ms jitter, meeting stringent real-time requirements. For mMTC, the system supports 78,000 devices/km² with a 91.2% connection success rate and 25% lower energy consumption than baseline approaches.

As shown in Fig. 5, the different services have distinct trends over time, with eMBB guaranteeing throughput stability and URLLC maintaining latency under 1ms, while mMTC connection density meets the specified requirements. The fairness index of 0.87 indicates that resources are reasonably distributed across all service types.

4.3. Scalability and Robustness Analysis

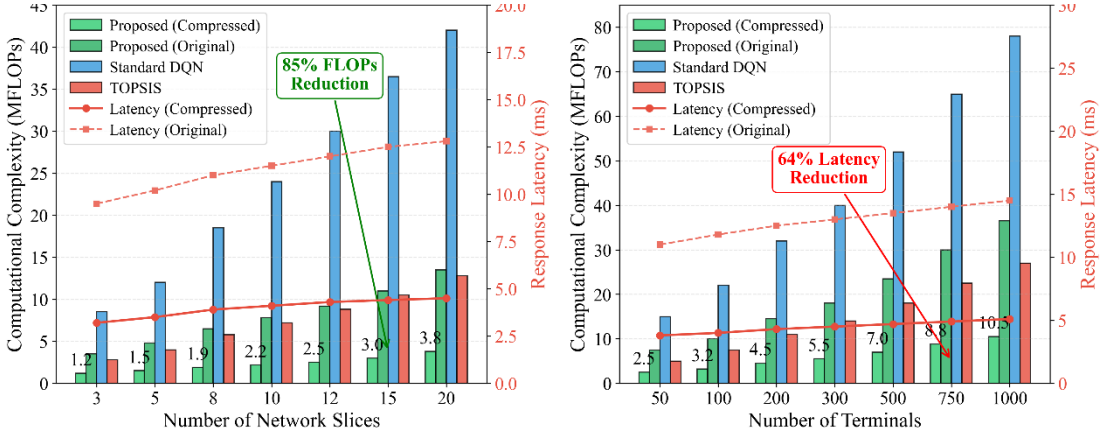
Large-scale testing evaluates system scalability with city-level networks of 50 base stations, 100 network slices, and up to 10,000 concurrent terminals. Average decision latency at this scale is 7.5ms with peaks under 12ms. Memory usage requires 48 MB for 10,000 terminals. By distributing deployment across edge nodes with lightweight messaging between them, computational capacity scales linearly. Load balancing keeps computational variability under 15% across edge node instances. Stress testing shows that a single engine handles 2,500 requests/s, while an eight-node cluster sustains up to 16,000.

4.3. Scalability and Robustness Analysis

Large-scale testing evaluates system scalability with city-level networks of 50 base stations, 100 network slices, and up to 10,000 concurrent terminals. Average decision latency at this scale is 7.5ms with peaks under 12ms. Memory usage requires 48 MB for 10,000 terminals. By distributing deployment across edge nodes with lightweight messaging between them, computational capacity scales linearly. Load balancing keeps computational variability under 15% across edge node

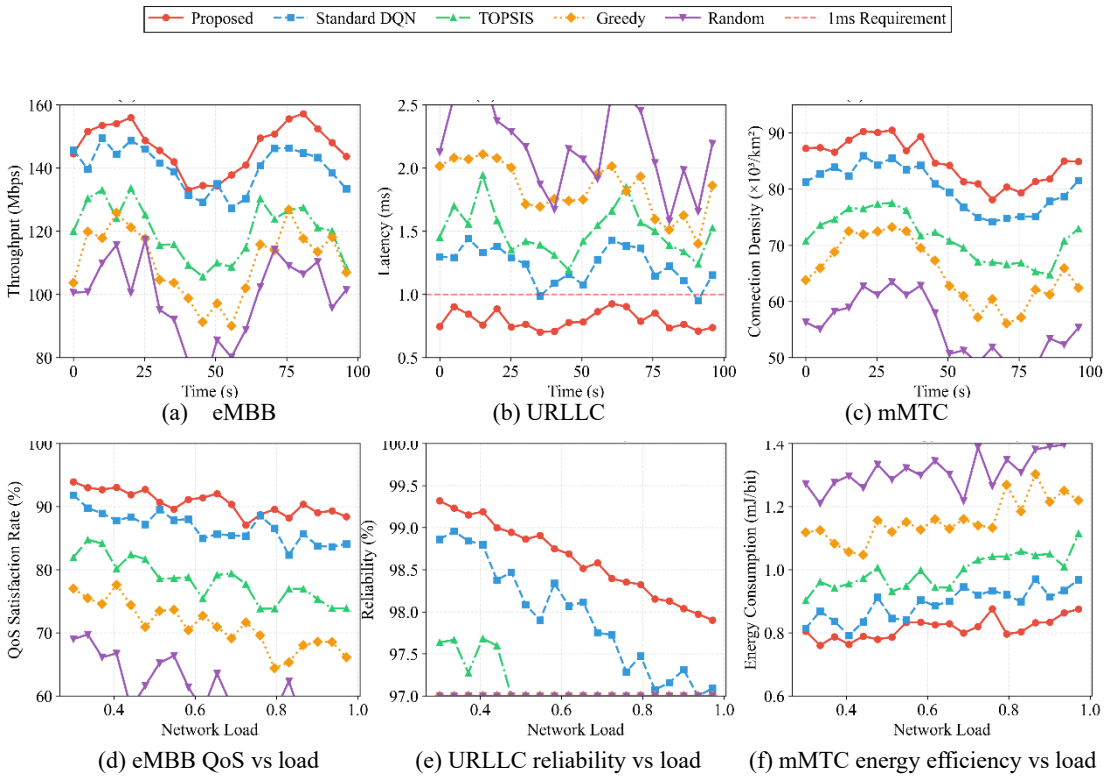
instances. Stress testing shows that a single engine handles 2,500 requests/s, while an eight-node cluster sustains up to 16,000.

Network adaptability plays an important role. Traffic surge tests increasing usage from 20% to 85% over 30 minutes show that the algorithm maintains at least 89% QoS compliance, approximately 10 percentage points above fixed algorithms. Failure scenarios disabling 20% of base station capacity show that 93.5% of devices reconnect within 5 seconds, with resource reallocation taking under 100ms. Mobile terminals traveling at 120 km/h maintain 97.8% handover success with sub-50ms latency through predictive resource management.



(a) Impact of network slice number (b) Impact of terminal number

Fig. 4. Algorithm computational complexity and response latency analysis



(a) eMBB (b) URLLC (c) mMTC

(d) eMBB QoS vs load (e) URLLC reliability vs load (f) mMTC energy efficiency vs load

Fig. 5. QoS performance comparison across different service types

Robustness testing confirms resilience under adverse conditions. During Distributed Denial of Service (DDoS) attacks containing 30% malicious traffic, anomaly detection isolates threats while legitimate service degradation stays below 5%. Data poisoning with 20% corrupted labels causes approximately a 10% performance loss, which is contained by the robust training-validation procedure. Network partitions between the control and data planes trigger a local decision mode that uses cached data and predictions, maintaining 82% access success during five-minute outages. At 95% resource utilization, degraded service mode preserves URLLC operations while throttling eMBB admissions to prevent system collapse. The exponential moving-average weight updates with elastic weight consolidation maintain at least 89% QoS compliance during traffic-surge and base-station-failure tests without triggering full retraining, providing an empirical answer to RQ3.

To assess generalizability, the algorithm was re-evaluated under three alternative traffic distributions beyond the default 3GPP mixture: a heavy-tailed Weibull eMBB ($k = 0.7$), a high-variance Poisson URLLC burst ($\lambda = 250$ events/s), and an

anonymized 24-hour metropolitan operator log (roughly 1.2 M flow records). QoS satisfaction ranged from 87.4% to 92.6%, a 3.7-percentage-point band around the 91.2% default. The worst case occurred under heavy-tailed eMBB, where 6.3% of decisions exceeded the URLLC budget due to head-of-line blocking; the adaptive weight mechanism recovered to within 2 percentage points of the URLLC budget after 200 cycles. Additional stress tests under high interference (SINR variance 6 to 12 dB), reduced spectrum (30% PRB removal), and high mobility (60 to 180 km/h) maintained QoS above 88.5% and kept URLLC latency within budget in all cases.

4.4. Practical Application Scenario Verification

V2X testing covered urban arterials and highways with traffic densities ranging from 20 to 200 vehicles per kilometer, using Cellular Vehicle-to-Everything (C-V2X) technology to support Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Network (V2N) communication simultaneously. Emergency brake alert latency, from hazard detection to warning transmission, was measured at 4.2ms, well below human reaction time. Under dense conditions of 200 vehicles/km, the algorithm kept the communication channel congestion below 15% through dynamic frequency and power control. Handover performance for high-velocity vehicles showed an average latency of 28ms, a 97.5% success rate, and a 93.1% seamless handover rate. In cooperative perception applications, optimized sensor data transmission improved blind-spot coverage by approximately 60%. Industrial IoT testing in manufacturing environments confirmed that the algorithm manages over 1,200 concurrent sensor devices with 96.8% data delivery reliability. As shown in Fig. 6, performance metrics remain stable across varying operational densities.

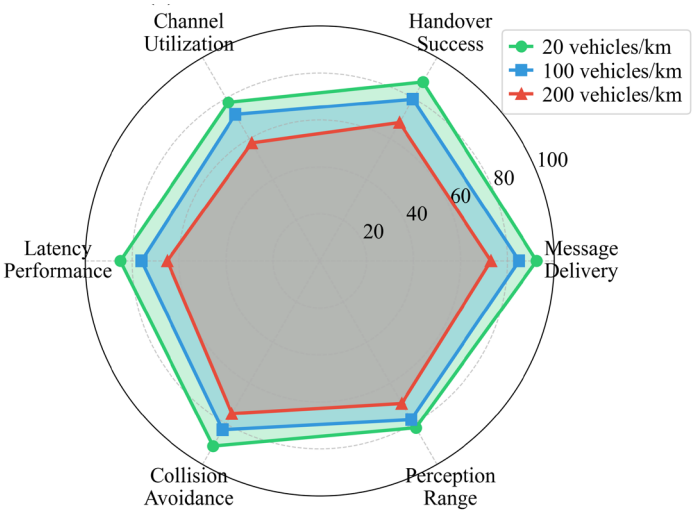


Fig. 6. Practical application scenario test results

4.5. Managerial and Operational Implications

The empirical results inform two classes of decisions that operators may revise after adopting the framework. Operational decisions for telecom providers. The 42 KB footprint and sub-8ms inference (Section 4.2) push access selection from edge servers down to terminals, enabling an estimated 30% to 40% reduction in dedicated edge inference capacity, lower edge GPU/NPU OPEX, and reallocation of freed cycles to analytics. The 30% terminal energy savings justify renegotiating IoT device lifetime SLAs in mMTC verticals where battery replacement dominates the total cost of ownership.

Resource allocation strategy changes. The adaptive weight mechanism (Section 3.4) replaces static MCDM weight tables, shifting planners from quarterly manual reviews to runtime adjustment and shortening response from days to minutes. The reward structure encodes an explicit congestion-time priority, URLLC reliability > eMBB throughput > mMTC connection density that should be incorporated into incident playbooks. The traffic-surge results (Section 4.3) set 85% utilization as a quantitative trigger for capacity-expansion procurement.

5. Conclusion and Future Perspectives

This study addresses the high computational complexity and resource consumption of terminal access selection in 5G network slicing. Through theoretical analysis and experimental verification, a lightweight algorithm is developed for resource-constrained terminals. Key contributions include the formulation of a two-stage coarse-to-fine selection framework, an improved Dueling-DQN decision engine with model compression and knowledge distillation, and adaptive parameter adjustment mechanisms. Testing demonstrates that the algorithm maintains comparable quality to conventional approaches while reducing computational requirements by approximately 78%, with response times under 8ms and energy savings of about 30%.

Returning to the three research questions: RQ1 is answered affirmatively, as the proposed framework satisfies eMBB, URLLC, and mMTC QoS targets at a 42 KB memory footprint and sub-8ms inference latency on the Snapdragon 888 platform (Section 4.2). RQ2 is supported by the distilled student model, retaining 93% of teacher accuracy while reducing parameters by 87% (Section 3.3). RQ3 is validated by traffic surge and failure scenarios in which adaptive weight updates are preserved at least 89% QoS compliance (Section 4.3).

Core technical contributions include service-specific feature extraction for state-representation compression, knowledge distillation from cloud-to-terminal models, adaptive weight allocation based on variable QoS requirements, and online learning for continuous adaptation. Compared with prior DRL-based slice selection, the theoretical contribution lies in the service-conditioned Dueling-DQN reward decomposition with identifiability normalization, the joint distillation–quantization–pruning pipeline that closes the deployment gap to terminal hardware, and the online EWC-based weight update with a Lyapunov-bounded convergence guarantee addressing aspects that single-stage, cloud-resident, or single-compression-technique DRL baselines do not cover simultaneously. The performance gap between cloud and compressed mobile models remains within 12%. Edge-trained models save approximately 15% of terminal power, although energy consumption at the edge itself requires further investigation.

Several limitations should be acknowledged. The algorithm is sensitive to the quality of the training data and requires time to adapt to new scenarios. The validation is predominantly simulation-based; although a 1.2-million-record operator trace was used, dataset diversity remains limited relative to multi-region, multi-operator deployments, and extreme distributional drift (e.g., disaster spikes, event handover storms) is uncharacterized. Hybrid multi-service applications are not fully modeled, and multi-terminal scheduling coordination remains preliminary with possible synchronization risks. Simulation-based energy estimates need real hardware validation. Real-world deployment further requires addressing terminal hardware heterogeneity (porting to non-NPU IoT chipsets), regulatory spectrum filtering, OSS/BSS integration with 3GPP TS 28.531/28.541 interfaces, and privacy-preserving online learning, all only partially supported in the present prototype. Future work will integrate larger and more heterogeneous multi-operator traces to strengthen external validity.

For practitioners, the managerial implications summarized in Section 4.5, terminal-side inference deployment, OPEX redistribution, runtime-driven weight adaptation, and explicit congestion-time prioritization order, provide a concrete action list for the next planning cycle. Future implementation involves phased pilot projects requiring ecosystem cooperation among chip manufacturers, equipment vendors, and network operators. Dynamic pricing and business models will emerge as the ecosystem develops. Adaptation to new technology environments requires cooperation from various stakeholders. As 5G technology matures and 6G development advances, efficient access selection will play an increasingly important role in enabling industrial digital transformation over the coming three to five years.

Author Contributions

Jun Liu, Hao-Ren Wei, and Qian Wang contributed to writing the main manuscript text. Pu Zhou and Tong Wang contributed to and prepared the figures. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

Declaration of Artificial Intelligence (AI) Tools

The authors used Grammarly (Premium edition) and ChatGPT (GPT-4) only for English grammar polishing and minor formatting assistance. No AI tool was involved in idea generation, methodology design, data analysis, result interpretation, or scholarly argumentation. The authors reviewed every AI-suggested edit and take full responsibility for all content.

References

Bakmaz, B., Bojkovic, Z., and Bakmaz, M. (2020). TOPSIS-based approach for network slice selection in 5G mobile systems. *International Journal of Communication Systems,* 33(11), e4395.

Barakabitze, A. A., Ahmad, A., Mijumbi, R., and Hines, A. (2020). 5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges. *Computer Networks,* 167, 106984.

Cai, Y., Cheng, P., Chen, Z., Ding, M., Vucetic, B., and Li, Y. (2023). Deep reinforcement learning for online resource allocation in network slicing. *IEEE Transactions on Mobile Computing,* 23(6), 7099-7116.

Dangi, R., Jadhav, A., Choudhary, G., Dragoni, N., Mishra, M. K., and Lalwani, P. (2022). Comprehensive analysis of network slicing for the developing commercial needs and networking challenges. *Sensors,* 22(17), 6623.

da Silva, D. C., Batista, J. O. R., Jr., de Sousa, M. A. F., Mostaço, G. M., Monteiro, C. de C., Bressan, G., Cugnasca, C. E., and Silveira, R. M. (2022). A novel approach to multi-provider network slice selector for 5G and future communication systems. *Sensors,* 22(16), 6066.

De Vleeschauwer, D., Papagianni, C., and Walid, A. (2021). Decomposing SLAs for network slicing. *IEEE Communications Magazine,* 59(10), 21-27.

Ejaz, M. A., Wu, G., and Iqbal, T. (2024). Dynamic and efficient resource allocation for 5G end-to-end network slicing: A multi-agent deep reinforcement learning approach. *International Journal of Communication Systems,* 37(17), e5916.

Foukas, X., Patounas, G., Elmokashfi, A., and Marina, M. K. (2017). Network slicing in 5G: Survey and challenges. *IEEE Communications Magazine,* 55(5), 94-100.

Hurtado Sánchez, J. A., Casilimas, K., and Caicedo Rendon, O. M. (2022). Deep reinforcement learning for resource management on network slicing: A survey. *Sensors,* 22(8), 3031.

Khan, L. U., Yaqoob, I., Tran, N. H., Han, Z., and Hong, C. S. (2020). Network slicing: Recent advances, taxonomy, requirements, and open research challenges. *IEEE Access,* 8, 36009-36028.

Khani, M., Jamali, S., Sohrabi, M. K., Sadr, M. M., and Ghaffari, A. (2024). Slice admission control in 5G cloud radio access network using deep reinforcement learning: A survey. *International Journal of Communication Systems,* 37(13),

e5857.
Li, R., Zhao, Z., Sun, Q., I, C.-L., Yang, C., Chen, X., Zhao, M., and Zhang, H. (2018). Deep reinforcement learning for resource management in network slicing. *IEEE Access,* 6, 74429-74441.
Li, Z., Li, H., and Meng, L. (2023). Model compression for deep neural networks: A survey. *Computers,* 12(3), 60.
Liu, Y.-J., Feng, G., Wang, J., Sun, Y., and Qin, S. (2021). Access control for RAN slicing based on federated deep reinforcement learning. *Proceedings of the IEEE International Conference on Communications (ICC).*
Lu, Y., Chen, X., Xi, R., and Chen, Y. (2020). An access selection mechanism in 5G network slicing. *Proceedings of the 2020 IEEE International Conference on Smart Internet of Things (SmartIoT).*
Malta, S., Pinto, P., and Fernandez-Veiga, M. (2025). Optimizing 5G network slicing with DRL: Balancing eMBB, URLLC, and mMTC with OMA, NOMA, and RSMA. *Journal of Network and Computer Applications,* 234, 104068.
Navarro-Ortiz, J., Romero-Diaz, P., Sendra, S., Ameigeiras, P., Ramos-Munoz, J. J., and Lopez-Soler, J. M. (2020). A survey on 5G usage scenarios and traffic models. *IEEE Communications Surveys & Tutorials, 22*(2), 905-929.
Palas, M. R., Islam, M. R., Roy, P., Razzaque, M. A., Alsanad, A., AlQahtani, S. A., and Hassan, M. M. (2021). Multi-criteria handover mobility management in 5G cellular network. *Computer Communications,* 174, 81-91.
Popovski, P., Trillingsgaard, K. F., Simeone, O., and Durisi, G. (2018). 5G wireless network slicing for eMBB, URLLC, and mMTC: A communication-theoretic view. *IEEE Access, 6*, 55765-55779.
Saha, N., Zangooei, M., Golkarifard, M., and Boutaba, R. (2023). Deep reinforcement learning approaches to network slice scaling and placement: A survey. *IEEE Communications Magazine,* 61(2), 82-87.
Sarah, A., Nencioni, G., and Khan, M. M. I. (2023). Resource allocation in multi-access edge computing for 5G-and-beyond networks. *Computer Networks,* 227, 109720.
Wijethilaka, S. and Liyanage, M. (2021). Survey on network slicing for Internet of Things realization in 5G networks. *IEEE Communications Surveys & Tutorials,* 23(2), 957-994.
Wu, Y., Dai, H.-N., Wang, H., Xiong, Z., and Guo, S. (2022). A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory. *IEEE Communications Surveys & Tutorials,* 24(2), 1175-1211.
Zhang, H., Wang, Z., and Liu, K. (2020). V2X offloading and resource allocation in SDN-assisted MEC-based vehicular networks. *China Communications, 17*(5), 266-283.
Zhang, S. (2019). An overview of network slicing for 5G. *IEEE Wireless Communications,* 26(3), 111-117.

Jun Liu completed her master's in Signal and Information Processing at the University of Electronic Science and Technology of China (UESTC) in 2005. Prior to joining academia, she spent over 18 years at Ericsson as a Senior Solutions Manager, where she specialized in the design and planning of mobile networks, bringing extensive expertise in cutting-edge communication solutions and a proven track record of delivering high-level technical seminars for clients. Her current research focuses on the convergence of mobile communication networks, cloud computing, embedded systems, and artificial intelligence.

Wei Haoren received his bachelor's degree in Communication Engineering from South China Normal University (2002) and his master's degree in Communications and Signal Processing from Newcastle University, UK (2005). He is currently an Associate Professor and Director of the Modern Mobile Communication Technology Program Teaching and Research Office at Guangdong Vocational College of Posts and Telecom. He has been honored with titles including "Outstanding Teacher of South Guangdong" and "Technical Expert of Guangdong Province". He has led four provincial-level research projects, co-authored three textbooks, published one EI-indexed paper, and been granted three utility model patents and two software copyrights. He has delivered over ten training sessions on 5G and AI for enterprises such as Guangdong Telecom, training more than 500 technical professionals. His research interests include 5G network slicing, mobile communication technology, and artificial intelligence applications.

Qian Wang obtained her master’s degree in Detection Technology and Automatic Devices from the University of Electronic Science and Technology of China (UESTC) in 2006. She is currently Assistant to the President at Geely University. She has published more than ten academic papers in Chinese core journals and applied for more than ten patents. Her research interests include embedded systems, signal processing, intelligent perception and artificial intelligence.

Pu Zhou received his Master of Engineering in Computer Application Technology from the University of Electronic Science and Technology of China in 2005. Currently, he serves as the Technical Lead at Chengdu Sobey Video Cloud Computing Co., Ltd. He possesses profound theoretical expertise and practical engineering capabilities in the fields of node communication, load balancing and domain-specific artificial intelligence.

Tong Wang obtained his Master's degree in Communication and Information Systems (2009) from Chongqing University of Posts and Telecommunications, Chongqing. Currently, he works as a double teacher at Geely University. He was an Ericsson R&D engineer and senior wireless integration engineer. His research interests include IoT, Internet of Vehicles Security.