

A DEEP REINFORCEMENT LEARNING-BASED CONTENTION WINDOW AVOIDANCE SCHEME FOR DISTINGUISHING NETWORK SERVICE PRIORITIES

Xiaowei Nie^{1,2,3}, Demin Wang^{4,5}, Yuqing Zhang⁶, Xiaoduo Pan^{1,2}, Zhibin Zuo^{4,5*}

¹National Tibetan Plateau Data Center (TPDC), State Key Laboratory of Tibetan Plateau Earth System, Environment and Resources (TPESER), Institute of Tibetan Plateau Research, Chinese Academy of Sciences, Beijing, China

²Science Center of Lingshan Forum of Guangdong Province, Guangzhou, China

³School of Ecology and Environment, Tibet University, Lhasa, China

⁴College of Information Science and Engineering, Henan University of Technology, Zhengzhou, China

⁵Henan International Joint Laboratory of Grain Information Processing, Zhengzhou, China

⁶University of Chinese Academy of Sciences, Beijing, China

Email: xwnie@itpcas.ac.cn, 2022930939@stu.haut.edu.cn, zhangyq@ucas.ac.cn, panxd@itpcas.ac.cn, zzbin_2000@haut.edu.cn

Abstract—This study introduces a Deep Reinforcement Learning-based Contention Window Avoidance Scheme aimed at improving service priority differentiation and network throughput under high load conditions within wireless networks that utilize the Network Distributed Coordination Function (DCF) mechanism. This novel approach involves two key strategies: firstly, it enables the differentiation of service priorities within DCF-based channel access by categorizing them in accordance with the varying data types and streams transmitted by different stations; secondly, it leverages Deep Reinforcement Learning to dynamically evaluate the network state through analysis of collision probabilities derived from channel observations. By doing so, the scheme dynamically modulates the ranges of contention windows in alignment with the service priorities, responding to the immediate network circumstances. Empirical evidence indicates that the proposed scheme substantially mitigates collision frequencies and augments data transmission throughput, thus significantly advancing the overall performance of networks that require service priority differentiation.

Keywords—*deep reinforcement learning, quality of service, backoff threshold, network performance optimization*

I. INTRODUCTION

The swift advancement of network communication technology has played a pivotal role in propelling societal progress and enriching people's lives with convenience. Wireless communication technology, in particular, has become a cornerstone for the proliferation of intelligent applications, effecting a transformative impact on daily life. Wi-Fi, by leveraging its advantages of extensive range, high-speed data transfer, and swift connectivity, has emerged as a preeminent wireless communication technology [1]. Nonetheless, the burgeoning number of wireless access devices and the expansion of application scenarios have precipitated significant performance bottlenecks for Wi-Fi networks, particularly in scenarios with high-density device access. Such congestion can lead to diminished data transmission rates and reduced network throughput. Moreover, the increasing diversity and complexity of services over Wi-Fi protocols, including latency-sensitive interactive gaming and augmented/virtual reality applications, demand heightened quality of service (QoS) standards, thereby underscoring the necessity for Wi-Fi networks to deliver reliable and efficient services amidst heavy device traffic.

Machine Learning (ML) [2-4], with its application emerging in various fields, presents innovative avenues to surmount these challenges. Reinforcement Learning (RL) [5] is especially adept at navigating the dynamic and intricate landscape of wireless networks by forging optimal strategies

through environmental interactions. Deep Reinforcement Learning (DRL) [6-8] marries the deep learning's prowess in feature extraction with the decision-making prowess of reinforcement learning, allowing for adaptation to a multitude of tasks and scenarios with robust scalability and applicability. Contemporary strategies employ DRL to optimize contention windows, thereby reducing collisions and enhancing Wi-Fi network performance. However, many of these approaches fall short in prioritizing service data types, which is critical for meeting the diverse QoS requirements of modern wireless networks.

This paper stands out from the existing body of research by proposing a novel adaptive contention window backoff scheme tailored to service priority differentiation, marking the following core contributions:

- (1) Incorporation of deep reinforcement learning to wirelessly sense network states and intelligently modulate contention window ranges, thereby optimizing channel access.
- (2) Establishment of distinct access parameters to prioritize channel access for various service types, thus fostering more efficient network communication through flexible scheduling and resource allocation.
- (3) Introduction of an adaptive backoff algorithm designed to adjust station backoff values in real-time, accommodating fluctuating network loads and enhancing overall network performance.

II. RELATED WORK

Wireless Local Area Networks (WLANs), leveraging Wi-Fi technology, serve to interconnect devices such as laptops, smartphones, and tablets within a confined geographical space. Machine Learning (ML) presents a dynamic solution capable of contending with the unpredictability and complexities of Wi-Fi networks by facilitating the acquisition of knowledge and enabling intelligent, autonomous policy adaptations in real-time to enhance network performance. The traditional IEEE 802.11 standard encounters difficulties in sustaining stable throughput in the face of escalating station numbers within Wi-Fi networks. Sandholm et al. [9] introduced a machine-learning-driven strategy for contention window regulation at access points (APs) in dense settings, markedly bolstering Wi-Fi network throughput. Despite the advancements in Wi-Fi 6 technology, with its adeptness at managing voluminous data, it grapples with maintaining consistent throughput in scenarios populated densely with stations, stemming from the limited scalability of its CSMA/CA mechanism. Chen et al. [10] put forward a deep learning-informed contention window management approach,

which employs deep learning to determine optimal contention window configurations under varied network conditions, thereby significantly enhancing system throughput, reducing average transmission delay, and curtailing packet retransmission rates for Wi-Fi 6, especially pertinent to the influx of Internet of Things (IoT) devices.

Deep Reinforcement Learning (DRL) synergizes the feature extraction abilities of Deep Learning (DL) with the decision-making prowess of Reinforcement Learning (RL), exhibiting remarkable scalability and versatility. Wydmański et al. [11] suggested a DRL-based contention window control strategy, using a Deep Q-Network (DQN) to discern the most suitable contention window (CW) values amidst diverse network states, with the objective of optimizing Wi-Fi network throughput. Ke et al. [12] went a step further by employing DQN to differentiate network conditions and ascertain a CW threshold, thereby refining the CW adjustment strategy based on network load and learning optimal settings tailored to specific scenarios. This nuanced approach to contention window modulation has been shown to not only elevate throughput but also diminish collision rates in Wi-Fi networks.

Current research in WLANs predominantly concentrates on the dynamic modulation of the contention window, transmission rate optimization, and other parameter modifications pertinent to channel access. While these enhancements aid in network performance under standard conditions, real-world applications often necessitate the simultaneous support of multiple services. A homogeneous scheduling strategy can fall short in meeting the diverse requirements of these services, potentially leading to insufficient bandwidth allocation or excessive delays for some services. Moreover, services with stringent performance demands may suffer performance degradation due to interference from other services.

III. DEEP REINFORCEMENT LEARNING-BASED CONTENTION WINDOW AVOIDANCE ALGORITHM FOR DISTINGUISHING NETWORK SERVICE PRIORITIES

The algorithm introduced in this manuscript is predicated on a DRL-enhanced Distributed Coordination Function (DCF) with inherent service priority differentiation. It is specifically designed to augment network throughput and curtail collision incidence in network environments characterized by high traffic loads, while concurrently maintaining differentiated Quality of Service (QoS) for diverse prioritized services. The algorithm encapsulates two principal components: the channel access mechanism and the reinforcement learning framework.

During the channel access phase, service types are differentiated by assigning distinct service priorities and corresponding access parameters. The algorithm employs an adaptive backoff method to compute a backoff threshold, which is predicated on the contention window range allied with the service priority and the extant contention window value of the station. This threshold facilitates differentiation between the number of transmitting stations, allowing for the selection of an appropriate backoff strategy.

The reinforcement learning component of the algorithm employs DRL techniques to ascertain the current network dynamics. It dynamically modifies the CW compensation value in response to network load conditions, consequently updating the minimum CW value for each service priority.

This update facilitates the adjustment of the CW range based on service priority. Through this adaptive approach, the algorithm is adept at accommodating varying network scenarios, thus optimizing network throughput and diminishing collision frequency.

A. The Channel Access Process

According to different service types, the DCF mechanism is differentiated in terms of service priority, and the corresponding priority and access parameters are defined, as shown in TABLE I.

The priority and AIFS time, which are the same settings as in the EDCA mechanism, ensure that stations with high priority service types have less waiting delay and thus seize the channel faster.

CW_{min} and CW_{max} , which indicate the initial contention window range corresponding to different priority levels. Higher priority stations/data streams have a smaller CW_{min} and can choose a smaller backoff value in order to send data faster. In each transmission cycle, the contention window CW_{min} corresponding to the priority level is updated based on the contention window compensation value $CW_{compensate}$ explored by the reinforcement learning process adapted to the current network situation, which is computed as shown in (1). Through this mechanism, the adaptive adjustment of the contention window range corresponding to the service priority is realized.

$$CW_{xmin} = CW_{xmin} + CW_{compensate} \quad (1)$$

The backoff threshold is used to distinguish the number of competing stations and the congestion level of the network. According to the adaptive fair EDCA algorithm in literature [13], the backoff threshold is calculated for each priority level, and its calculation function is shown in (2), where TH_x represents the backoff threshold corresponding to priority X, and CW_{xmin} and CW_{xmax} represent the minimum and maximum contention windows corresponding to priority X. CW_x is the current contention window value of priority X station, and t_s represents a random value in the backoff counter while satisfying a uniform distribution in $(0, CW_x - 1)$, and t_s represents the actual length of the time slot. If the number of stations for data transmission is large, the value of TH_x is large and converges to CW_{xmax} ; if the number of stations for data transmission is small, the value of TH_x is small and converges to CW_{xmin} .

$$TH_x = \frac{CW_{xmax} - CW_x}{CW_{xmax} - CW_{xmin}} \times \frac{bt_x}{CW_x} \times CW_{xmin} \times t_s \quad (2)$$

TABLE I. DEFINITION OF CHANNEL INITIAL ACCESS PARAMETERS

Type	AC VI	AC VO	AC BE	AC BK
priority	1	2	3	4
AIFS	9us	17us	34us	94us
CW_{min}	16	32	64	128
CW_{max}	1024	1024	1024	1024
T_x	8000us	4000us	2000us	1000us

When multiple stations with different service types access the channel for data forwarding, different service priorities and access parameters are first assigned according to the service types of the stations. The wireless access point (AP), as an intelligent body of the reinforcement learning process, periodically broadcasts the contention window compensation

value $CW_{compensate}$ obtained from exploration to all stations via beacon frames (beacon), so as to update the contention window range corresponding to the service priority to better adapt to the current network conditions, and the channel access process mainly consists of three phases.

a) *Backoff stage*: the station with priority X monitors that the channel is idle, assuming that the channel is idle for AIFSX time. The other priority station executes the backoff process, through the competition window size to get the corresponding backoff counter value bt_i and service priority X backoff threshold TH_x , based on the comparison of the values of bt_i and TH_x , different backoff strategies are adopted as shown in (3). If $bt_i > TH_x$, it means that the number of competing station in the current network is more, the value of the backoff counter bt_i will be in the process of linear reduction, which will be subtracted by 1 after an interval of one time slot, if $bt_i < TH_x$, it means that the number of competing station in the current network is less, the value of the backoff counter will be in the process of exponential reduction, which means that the current backoff value will be halved. Until the station's bt_i is 0, wait for a time slot interval and access the channel for data transmission.

$$\begin{cases} bt_{i(old)} > TH_x & bt_{i(new)} = bt_{i(old)} - 1 \\ bt_{i(old)} < TH_x & bt_{i(new)} = bt_{i(old)} / 2 \end{cases} \quad (3)$$

b) *Successful Transmission Stage*: If the station succeeds in accessing the channel and transmits data successfully, the station will count the sum of the time it has occupied the channel and the time of the next data transmission S_x , and then compare S_x and T_x . If $S_x > T_x$, the station no longer occupies the channel and sets the size of the contention window value to $CW_x = CW_{xmin}$ and $T_x = 0$. if $S_x \leq T_x$, the station waits for a time slot interval to try again to access the channel for data transmission. access the channel for data transmission.

c) *Conflict collision stage*: If the site fails to send data. The size of the station's current competition window will be adjusted as shown in (4).

$$CW_x = \min(CW_{xmax}, 2 * CW_x) \quad (4)$$

B. The Reinforcement Learning Process

The reinforcement learning process focuses on predicting and adjusting the contention window compensation value $CW_{compensate}$ by utilizing the DRL technique. The intelligent body is located at the wireless access point (AP) and has a global view of the entire network. By applying the Partially Observable Markov Decision Process (POMDP), the intelligent body (AP) is able to efficiently deal with incomplete observations and make appropriate decisions based on the perceived network conditions in order to determine the best Contention Window compensation value $CW_{compensate}$, which in turn adjusts the contention window ranges corresponding to the different optimization levels to maximize the optimized network performance. Once the appropriate CW compensation value is determined, the AP broadcasts it to all stations via beacon frames to achieve cooperative optimization of the contention window.

POMDP can be defined as a 7-tuple $(S, A, T, R, \Omega, O, \gamma)$, as defined below:

S is the state space set ($s \in S$): it represents the exact state of all devices in the network.

A is the action space set ($a \in A$): it represents the possible actions that can be selected in the current state and is used to adjust the size of the $CW_{compensate}$. Where $a \in [0, 7]$, therefore, the $CW_{compensate}$ ranges from 0 to 896.

$$CW_{compensate} = 2^a \times a \quad (5)$$

T is the state transfer probability: it represents the probability distribution of performing an action (a) in the current state (s) and transferring to another state (s').

R is the reward function: it represents the reward obtained by arriving at the next state after executing an action in the current state, given that throughput is a direct response to the performance of the network, throughput is chosen as the reward function.

Ω is the set of observation space ($o \in \Omega$), in which we define the network historical collision rate (the collision rate of the previous state and the collision rate of the current state) $H(P_{col})$ as the observation o , where P_{col} represents the probability of an unsuccessful transmission, i.e., the collision rate, which is defined as shown in (6), where N_t is the total number of frames sent by the site and N_r is the number of correctly received frames.

$$P_{col} = (N_t - N_r) / N_t \quad (6)$$

$\gamma \in [0, 1]$ is a discount factor, which is used to measure how much the intelligent body values future rewards.

In this paper, by using the DQN algorithm in DRL, the efficiency and stability of learning are further improved through the experience playback mechanism and the target network mechanism. The specific algorithmic process is as follows:

First initialize the experience pool D and some key parameters, the contention window compensation value $CW_{compensate}$ is initialized to 0, and the state is initialized to zero vector. The weight parameters of the evaluation network and the target network should be consistent, i.e., $\theta' = \theta$. After the initialization is completed, the training starts to be executed, and then each time step is executed. For each time step, the intelligent body (AP) first observes the preprocessed historical collision rate $H(P_{col})$ to perceive the current network state (s), and the main network (Q-Network) uses the ϵ -greedy method to select an action in the current state a . Based on the selection of the action, the intelligent body (AP) computes a new $CW_{compensate}$ and broadcasts it to all stations update their contention window ranges based on the $CW_{compensate}$ and the service priority. After a cycle, the intelligent body (AP) evaluates the network performance for that cycle, calculates the reward r (normalized throughput) and the next state s' , and then stores the experience samples (s, a, r, s') for this cycle into the experience pool. A small batch of samples d is also proposed from the experience pool D, and the time-differential objective is computed for each sample. The parameter θ in the loss function $L(\theta)$ is updated using gradient descent. the target network is then updated, keeping the parameters of the evaluation network and the target network consistent every C steps. When the training phase is complete, the second phase, the evaluation phase,

begins, and the intelligent body only needs to observe the state and get the action through the trained model; there is no need for rewards, since the intelligent body is already considered fully trained and has stopped receiving updates.

Algorithm 1 Deep Reinforcement Learning-based Contention Window Avoidance Algorithm for Distinguishing Network Service Priorities

```

1 Initialize  $\theta' \leftarrow \theta$ 
2 Initialize  $s \leftarrow$  vector of zeroes
3 Initialize  $CW_{\text{compensate}} \leftarrow 0$ 
4 for  $Episode = 1$  to  $Episode_{\text{max}}$  do
5   for  $step = 1$  to  $step_{\text{max}}$  do
6      $s \leftarrow$  preprocess ( $H(P_{\text{col}})$ )
7      $a \leftarrow$  Apolicy( $s$ )
8      $CW_{\text{compensate}} \leftarrow 128 \times a$ 
9     if train then
10       $s' \leftarrow$  preprocess ( $H(P_{\text{col}})$ )
11       $r \leftarrow$  normalize (throughput)
12      D.push( $(s, a, r, s)$ )
13      D  $\leftarrow$  d Extracting small batch samples from the
        experience pool
14      Calculate the time-difference target for each
        sample
15      Update the loss function using gradient descent
16      if  $step \% c == 0$  then
17         $q(s', a', \theta') \leftarrow q(s, a, \theta)$ 
18      end if
19   End for
20 End for

```

IV. SIMULATION EXPERIMENTS AND DISCUSSIONS

A. Simulation Parameter Settings

In this study, we employ the Baidu-provided Pral and PaddlePaddle frameworks, alongside the Python programming language, to develop a deep reinforcement learning algorithm for contention window avoidance that prioritizes network service types. Our experimental setup is based on the IEEE 802.11ac standard, utilizing 1024-QAM modulation and a 5/6 coding rate over a 20 MHz channel. The system configuration comprises a single Access Point (AP) managing multiple station transmissions with an 11 Mbps transmission rate from the station to the AP. We make two key assumptions: (1) the AP has the capability to discern the current network state, including the collision rate and network throughput, and (2) the AP periodically disseminates the $CW_{\text{compensate}}$ parameter through beacon frames.

The experiments utilize a Deep Q-Network (DQN) algorithm with essential parameters detailed in TABLE II. The network architecture employed by the algorithm is a DNN consisting of three hidden layers of fully-connected networks, each with an output dimension of 128. The Rectified Linear Unit (ReLU) activation function is employed within the network layers Simulation Experiments and Analysis

The inaugural experiment aimed to evaluate how the contention window compensation value, denoted as

$CW_{\text{compensate}}$, varied with changing site densities. The experiment considered a spectrum of site densities, with the number of competing sites ranging from 20 to 160, distributed evenly across various service types. Over the course of the experiment, the learning algorithm was executed for 20 iterations, with the number of sites incrementing by intervals of 20, to allow for the stabilization of the compensation value. The findings, depicted in Fig. 1, illustrate that the compensation value of the contention window displayed a linear escalation in relation to the rising number of contending sites. This escalation in $CW_{\text{compensate}}$ correspondingly augmented the minimum contention window value for each service priority, which in turn enlarged the contention window for each site associated with a particular priority. Such modulation effectively mitigated potential data transmission conflicts among the sites. Concurrently, the implementation of an adaptive backoff algorithm facilitated the expedited conclusion of the backoff process, thereby curtailing the latency associated with waiting.

TABLE II. DQN ALGORITHM PARAMETERS

parameters	value
Learn frequency	5
Memory size	20.000
Batch size	32
Learning rate α	0.001
Epsilon greedy ϵ	0.1-1
Epsilon greedy ϵ decrement	1e-6
Step _{max}	200
Episode _{max}	20

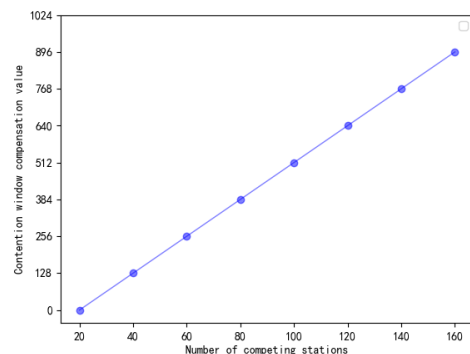


Fig. 1. Contention window compensation values under different number of stations

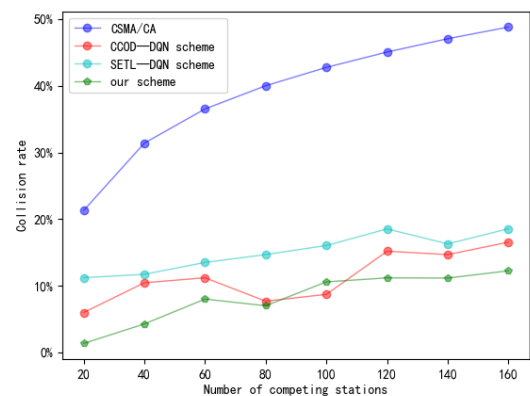


Fig. 2. Collision rates at different number of stations

The second experiment conducted a comparative analysis of collision rate performance by juxtaposing the conventional Carrier Sense Multiple Access (CSMA) mechanism with advanced deep reinforcement learning-based algorithms—specifically, CCOD-DQN [7] and SET-DQN [8]—and the contention window adaptation algorithm presented in this study, which accounts for network service priority differentiation. The experimental results are illustrated in Fig. 2. The findings revealed that the deep reinforcement learning-based contention window optimization schemes outperform the traditional CSMA mechanism by significantly lowering the collision rates across various network load conditions. While the CCOD-DQN and SET-DQN algorithms enhance the conventional Binary Exponential Backoff (BEB) mechanism, they do not differentiate among service types. In marked contrast, the algorithm proposed herein demonstrates a pronounced reduction in collision rates under high load conditions by incorporating service type differentiation, maintaining an equal number of sites per service type. Notably, even with a high site density of 160, the collision rate was curtailed to 11.21%, thus sustaining low collision rates across a spectrum of site densities and consequently bolstering network performance.

B. Discussion

The CSMA/CA protocol employs the BEB algorithm to regulate the CW, which proves effective at mitigating collisions in networks with a small scale and light traffic. However, as network size and load intensify, the collision rate surges, leading to a decrease in throughput. The CCOD-DQN scheme leverages DRL to predict optimal CW values for enhancing Wi-Fi network throughput and reducing collision rates in high-load environments. Nonetheless, its reliance on the BEB algorithm for CW adjustment fails to accommodate rapid network changes, incurring increased transmission delays and reduced throughput.

The SEPL-DQN approach uses DRL to determine an appropriate CW threshold, discerning network load to fine-tune the CW via the SETL algorithm. It employs exponential adjustments for light loads and linear adjustments for heavy loads, aiming to decrease collision rates and elevate throughput. Nevertheless, under light loads, the exponential adjustment may prolong waiting times at stations, and under heavy loads, the lessened increase in the CW range due to linear adjustment may heighten the likelihood of collisions.

While DRL-based solutions significantly enhance Wi-Fi network performance, they generally neglect to differentiate between service priorities. This oversight can compromise the QoS of various service types in scenarios featuring multiple services. Addressing this gap, the present study introduces a contention window backoff scheme that differentiates service priorities. Utilizing the principles of DRL, the proposed scheme adapts the CW range to correspond to service priorities based on current network conditions. An adaptive backoff algorithm is implemented to expedite the backoff process for stations, thereby ensuring QoS and maximizing network performance.

V. CONCLUDE

This study introduces a deep reinforcement learning-based algorithm designed to modulate the CW for effectively differentiating network service priorities. The algorithm

stratifies service types by their respective priority levels for channel access at various sites. Utilizing deep reinforcement learning, the algorithm endeavors to determine a contention window compensation value, denoted as CWcompensate that is congruent with the prevailing network conditions, which, in turn, is used to revise the CW range associated with each service priority. Concurrently, an adaptive backoff algorithm is implemented to select distinct backoff strategies in accordance with the density of competing sites, thereby addressing some of the performance limitations inherent to Wi-Fi technology. Empirical results confirm that the proposed algorithm confers considerable improvements in network throughput and reductions in collision rates. Future work might expand upon this research to encompass more intricate network structures and varied service environments to enhance the algorithm's utility and efficacy.

ACKNOWLEDGMENT

This work obtained the support of Key Science and Technology Program of Henan Province, China (No.232102210132).

REFERENCES

- [1] E. Mozaffariahrar, F. Theoleyre, and M. J. F. I. Menth, "A survey of Wi-Fi 6: Technologies, advances, and challenges," *Future Internet*, vol. 14, no. 10, pp. 293, 2022.
- [2] S. Szott et al., "Wi-Fi meets ML: A survey on improving IEEE 802.11 performance with machine learning," *IEEE Commun. Surv. Tut.* vol. 24, no. 3, pp. 1843-1893, 2022.
- [3] J. Parmar, S. Chouhan, V. Raychoudhury, and S. J. A. C. S. Rathore, "Open-world machine learning: applications, challenges, and opportunities," *ACM Comput. Surv.* vol. 55, no. 10, pp. 1-37, 2023.
- [4] K. Sharifani, M. J. W. I. T. Amini, and E. Journal, "Machine Learning and Deep Learning: A Review of Methods and Applications," *World Information Technology and Engineering Journal*, vol. 10, no. 07, pp. 3897-3904, 2023.
- [5] Y. Huang, K.-W. J. I. T. o. M. L. i. C. Chin, and Networking, "A Three-Tier Deep Learning Based Channel Access Method for WiFi Networks," *IEEE Transactions on Machine Learning in Communications and Networking*, 2023.
- [6] X. Wang et al., "Deep reinforcement learning: a survey," *IEEE T. Neur. Net Lear.* 2022.
- [7] A. Oroojlooy and D. J. A. I. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *Appl. Intell.* vol. 53, no. 11, pp. 13677-13722, 2023.
- [8] A. Y. Majid, S. Saaybi, V. Francois-Lavet, R. V. Prasad, C. J. I. T. o. N. N. Verhoeven, and L. Systems, "Deep reinforcement learning versus evolution strategies: a comparative survey," *IEEE T. Neur. Net. Lear.* 2023.
- [9] T. Sandholm, B. Huberman, B. Hamzeh, and S. J. a. p. a. Clearwater, "Learning to wait: Wi-fi contention control using load-based predictions," *arXiv preprint arXiv:1912.06747*, 2019.
- [10] C. Chen, J. Li, V. Balasubramaniam, Y. Wu, Y. Zhang, and S. J. I. I. o. T. J. Wan, "Contention resolution in Wi-Fi 6-enabled Internet of Things based on deep learning," *IEEE Internet. Things.* vol. 8, no. 7, pp. 5309-5320, 2020.
- [11] W. Wydmański and S. Szott, "Contention window optimization in IEEE 802.11 ax networks with deep reinforcement learning," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, 2021, pp. 1-6: IEEE.
- [12] C.-H. Ke, L. J. K. T. o. I. Astuti, and I. Systems, "Applying deep reinforcement learning to improve throughput and reduce collision rate in IEEE 802.11 networks," *KSH. T. Internet. Inf.* vol. 16, no. 1, pp. 334-349, 2022.
- [13] M. Malli, Q. Ni, T. Turletti, and C. Barakat, "Adaptive fair channel allocation for QoS enhancement in IEEE 802.11 wireless LANs," in *2004 IEEE International Conference on Communications (IEEE Cat. No. 04CH37577)*, 2004, vol. 6, pp. 3470-3475: IEEE.