IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 5, 2022, accepted April 15, 2022, date of publication April 25, 2022, date of current version May 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3168549

Smart Packet Transmission Scheduling in Cognitive IoT Systems: DDQN Based Approach

ADEEB SALH[1,2,3], LUKMAN AUDAH[1,2], MOHAMMED A. ALHARTOMI[3], (Member, IEEE), KWANG SOON KIM[4], (Senior Member, IEEE), SAEED HAMOOD ALSAMHI[5,6], FARIS A. ALMALKI[7], QAZWAN ABDULLAH[1,2,8], (Member, IEEE), ABDU SAIF[9], (Member, IEEE), AND HANEEN ALGETHAMI[10], (Senior Member, IEEE)

[1]Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, Johor 86400, Malaysia
[2]Malaysia and Community College, Yarim, Yemen
[3]Department of Electrical Engineering, University of Tabuk, Tabuk 47512, Saudi Arabia
[4]School of Electrical and Electronics Engineering, Yonsei University, Seodaemun-gu, Seoul 03277, South Korea
[5]SRI, Athlone Institute of Technology, Technical University of the Shannon: Midlands Midwest, Athlone, Westmeath, N37 F6D7 Ireland
[6]Faculty of Engineering, Ibb Univerisity, Ibb, Yemen
[7]Department of Computer Engineering, College of Computers and Information Technology, Taif University, Taif 21944, Saudi Arabia
[8]Faculty of Electrical Engineering, Universiti Teknikal Malaysia Melaka, Durian Tunggal 76100, Malaysia
[9]Department of Electrical Engineering, Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia
[10]Department of Computer Science, College of Computers and Information Technology, Taif University, Taif 21974, Saudi Arabia

Corresponding authors: Lukman Audah (hanif@uthm.edu.my) and Qazwan Abdullah (gazwan20062015@gmail.com)

This work was supported in part by Universiti Tun Hussein Onn Malaysia under Grant E15216; in part by the University of Tabuk, Saudi Arabia, under Project S-0237-1438; and in part by the Deanship of Scientific Research at Taif University, Saudi Arabia, through Taif University Researchers Supporting Project TURSP-2020/265. The work of Saeed Hamood Alsamhi was supported in part by the European Union's Horizon 2020 Research and Innovation Program under the Marie Skłodowska-Curie under Grant 847577, and in part by the Research Grant from Science Foundation Ireland (SFI) through thr Ireland's European Structural and Investment Funds Programs and the European Regional Development Fund (2014–2020) under Grant 16/RC/3918.

ABSTRACT The convergence of Artificial Intelligence (AI) can overcome the complexity of network defects and support a sustainable and green system. AI has been used in the Cognitive Internet of Things (CIoT), improving a large volume of data, minimizing energy consumption, managing traffic, and storing data. However, improving smart packet transmission scheduling (TS) in CIoT is dependent on choosing an optimum channel with a minimum estimated Packet Error Rate (PER), packet delays caused by channel errors, and the subsequent retransmissions. Therefore, we propose a Generative Adversarial Network and Deep Distribution Q Network (GAN-DDQN) to enhance smart packet TS by reducing the distance between the estimated and target action-value particles. Furthermore, GAN-DDQN training based on reward clipping is used to evaluate the value of each action for certain states to avoid large variations in the target action value. The simulation results show that the proposed GAN-DDQN increases throughput and transmission packet while reducing power consumption and Transmission Delay (TD) when compared to fuzzy Radial Basis Function (fuzzy-RBF) and Distributional Q-Network (DQN). Furthermore, GAN-DDQN provides a high rate of 38 Mbps, compared to actor-critic fuzzy-RBF's rate of 30 Mbps and the DQN algorithm's rate of 19 Mbps.

INDEX TERMS Artificial intelligence, Cognitive Internet of Things, transmission delay, packet error rate.

I. INTRODUCTION

Recently, the Internet of Things (IoT) has emerged as a promising vision. Beyond fifth-Generation (B5G) can be intelligently interconnected to the growing usage of application services such as mobile phones, video streaming, and video conferencing in business and daily life. Application services enable people to access streaming applications anywhere and anytime while also providing big data in real-time. Improving a wireless multimedia application (i.e., vehicles, monitors, YouTube, Skype, and web browsing) is dependent on the packet transmission schedule in Ultra-Reliable Low Latency Communications (URLLC) [1]–[3]. In addition, URLLC is closely related to mission-critical IoT applications due to stringent constraints on the combined latency and reliability [2].

The associate editor coordinating the review of this manuscript and approving it for publication was Tariq Umer.

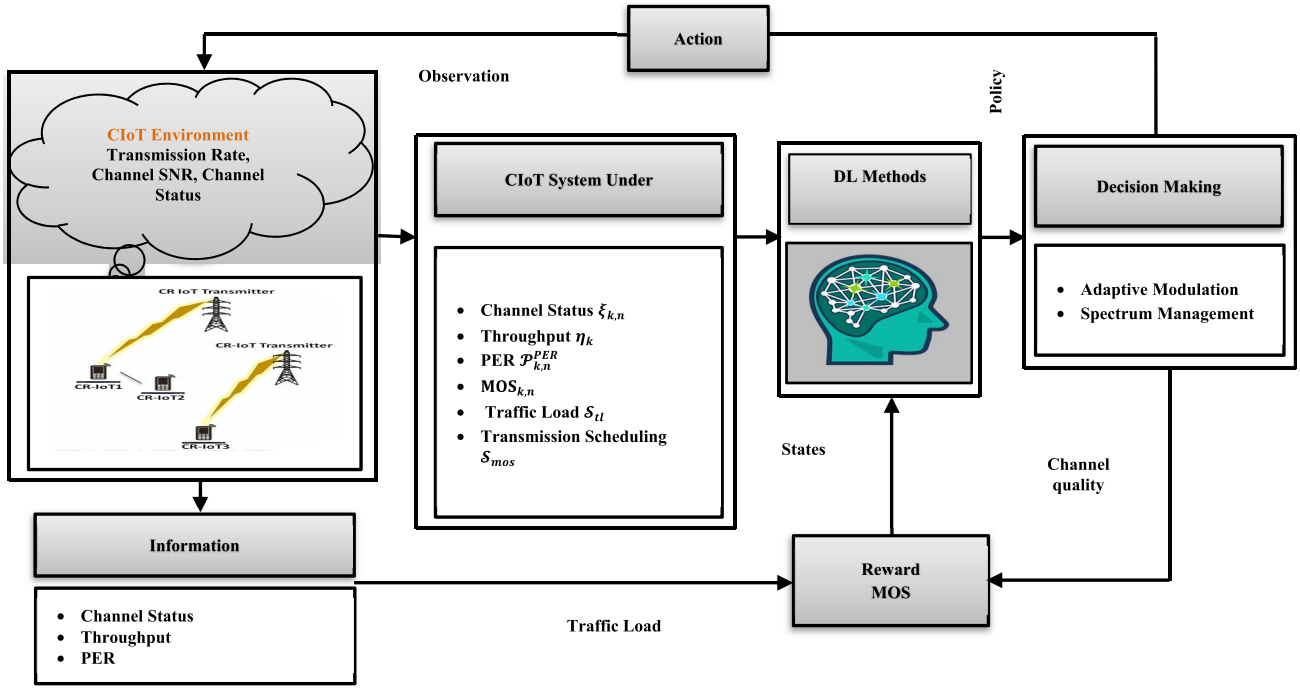

FIGURE 1. Designing of the decision-making of DNN in CIoT system.

In particular, controlling loss and enhancing packet Transmission Scheduling (TS) in the Cognitive IoT (CIoT) system are two significant challenges for URLLC systems. Many studies have proposed spectrum schemes [4], [5] to guarantee the quality of service requirements and provide a high transmission data rate. Artificial Intelligence (AI) has been used in CIoT to keep up with data volume while minimizing energy consumption, traffic management, and data storage. A new Q-learning-based TS is proposed to solve packet transmission efficiency in CIoT systems [6]. The packet transmission efficiency using a Deep- Learning (DL) agent can substantially enhance its prediction if it becomes more intelligent for the CIoT [6].

To provide better-performing TS in the Cognitive Internet of Vehicles (CIoV) system, Deep-Reinforcement Learning (DRL) has difficulty achieving a large label of a real dataset in real-time [7]. The Generative Adversarial Networks (GANs) are an emerging technique that generates new virtual data similar to the existing data needed to achieve packet TS, thus allowing the DRL agent to gain knowledge. The Deep Distribution Q Network (DDQN)-based GAN proposed to plan an intelligent agent [7]. The model-free actor-critic for DRL is proposed to solve the problem of TS by applying the learning problem for intelligent resource allocation in CIoT systems and improving transmission packet rate and power consumption [8]. In addition, the GAN-based DDQN algorithm improves training stability in CIoT by efficient transfer to estimate the value of every action for certain states and the expectation of the action-value distribution. Our entire procedure of using DRL for CIoT systems is dependent on designing an intelligent agent Fig. 1.

A. MOTIVATION AND CONTRIBUTIONS

The current low packet transmission efficiency of IoT faces a problem of the crowded spectrum because of the rapidly increasing popularity of various wireless applications. A major challenge in CIoT is packet transmission efficiency. The unexpected growth in arrival rate to all Users (UEs) necessitates unnecessary overhead and long retransmit in the case of extreme events. Because URLLC applications are sensitive to reliability and latency, short periods of unreliability or latency can significantly impact UEs. Managing spectrum decisions must enhance the dynamic channel nature of Cognitive Radio Networks (CRNs), which provide a large volume of data based on the estimated Packet Error Rate (PER), channel status, throughput, and packet retransmission delay.

Furthermore, to overcome the issue of underestimation of action-value due to the effect of random noise in CIoT. Therefore, we propose a GAN-based DDQN empowered by the Software-Defined Network (SDN) controller in a highly complex IoT environment for intelligent TS. The main contributions of this work are as follows:

- We explore how to improve CIoT throughput by maximizing the quality of the transmitted packet rate and reducing Transmission Delay (TD) based on channel transmission, Signal-to-Noise Ratio (SNR), and PER for choosing a good channel and reducing the spectrum handoff in the multimedia applications.
- We propose a Radial Basis Function (RBF) learning algorithm for reducing transmission power, which is dependent on the current state of the decision policy to obtain intelligent TS for every UE in CIoT systems.

Furthermore, we use convergent actor-critic to provide a reasonable decision for real-time learning at the output layer of the fuzzy-RBF learning algorithm. The integration of the actor-critic fuzzy-RBF learning algorithm has the capability of solving the TS problem for a large number of transmission packets under updating temporal-difference error.

- We propose GAN- DDQN to enhance the action improvement value for each action, resolve the DRL long training process issue, and real-time processing of the collected real data. The DDQN learning suffers from the challenge that only a small part of the generator output is included in calculating the loss function during the training. The proposed GAN-DDQN can remove this loss when the agent is deployed, reduce delay, improve throughput, and perform TS. To stabilize GAN-DDQN training, we propose a new reward-clipping process that can prevent large variations in the target action value.

B. RELATED WORKS

Achieving an intelligent spectrum handoff decision in CRN depend on the proposed transfer actor-critic learning selection that uses a comprehensive reward function that considers a good knowledge of channel quality, packet error rate, packet dropping rate, and throughput [9]. Energy management, resource allocation, and TS are the challenges in the CIoT system, which require the design of a learning agent to develop decision-making ability [9]. To solve the distributed resource allocation problem for IoT using cognitive hierarchy devices, human-type devices exist in CIoT systems and machine-type devices [10]. Moreover, reducing power consumption and low-energy UEs such as narrowband-IoT depend on scheduling data transmissions to fulfill the URLLC requirements for cellular networks [11]. Therefore, improved CIoT systems for URLLC requirements must provide data transmission, reducing average packet latency, reliability, energy efficiency, and internet connectivity [12]–[15]. Furthermore, improving the packets scheduling strategy for CIoT based on the proposed discrete permutation particle swarm optimization for scheduling packets for every time interval in [16] depends on minimizing the packets' queuing delay and the number of dropped packets at each packet time interval for CIoT. To support large-size traffic and guaranteed traffic packets in Real-time for future smart systems.

The authors in [17], [18] formulate the problem by choosing the optimal CRN channel selection and applying deep-RL for refiner GANs to provide sufficiently accurate traffic packs in real-time. Based on previous studies [7]–[18], CIoT systems are still not smart enough to search for the optimal policy. To make the system performance more intelligent to search for the optimal policy, we propose an RBF to extract the intelligence and improve the performance of TS based on developing 'intelligent' fuzzy controllers. Improving the success of transmitted packets is dependent on minimizing the contradiction between the evaluated and target action-value distributions, which achieves the optimal resource allocation policy for GAN-powered DRL [19]. Good transmission packet scheduling guarantees high End-to-End (E2E) reliability based on the proposed experienced DRL for action space reducer that reduces the size of the action space of GAN [20]. Previous studies [13]–[20] often override the real-time request for real-time traffic and the influence of delay for retransmissions. Whereas real-time communications over IEEE 802.11 are vital to meet the high efficient TS in the CIoT system.

II. SYSTEM MODEL

In this section, we consider that the wireless IoT devices' downlink is randomly distributed in a circular cell, and every IoT device selectively adapts to the lower modulation levels: Binary Phase-Shift Keying (BPSK), 4- Quadrature Amplitude Modulation (QAM), 8-QAM, and 16-QAM. The GAN scheduling is applied in the SDN-based radio access network for dynamic CIoT and a noisy wireless bandwidth with downlink transmissions in the radio access network.

A. CHANNEL STATE

We considered a CRN with *nth* independent channels; every channel is allocated to UEs. The transmit packet arrival rate in CIoT is modeled as a Poisson distribution process in IoT. The SNR is independent and identically distributed between different transmissions during the transmission packet. The probability distribution function density of the received SNR statistically can be written as:

$$\mathcal{P}\left(\xi_{k,n}\right) = \frac{1}{\bar{\xi}} \exp\left(-\frac{\xi}{\bar{\xi}}\right), \tag{1}$$

where ξ represents the instantaneous SNR of the *n th* channel at the receiver and $\bar{\xi}$ represents the average received SNR. Let $\xi_{k,n}$ be the received SNR at the *kth* transmission after the packets are combined at time slot $\mathfrak{Q}$, then ensuing SNR at the *kth* transmission is an adopted system. The received SNR defines the perfect channel state information at the receiver. The status of *n th* channel involvements blocks Rayleigh fading with time slot $\mathfrak{Q}$, $\xi_{k,n}$ presents a binary variable $\epsilon \{0, 1\}$ [15], [21]. If $f_k(\xi_n) = 1$, the channel is busy by one transmission packet; else, the channel is idle.

B. POWER CONSUMPTION MODEL

Power consumption depends on the small-scale channel gains. The several packets must wait for retransmission the next time. Every device has two power consumption statuses in each time slot $\mathfrak{Q} \epsilon \{0, 1\}$. The transmit power $\mathcal{P}_j$ indicates the transmitted data packets to every device. To reduce the status power consumption under the queuing list to access this channel and wait to be arranged to other channels to obtain the low-power level for every device on the *nth* channel is modeled as:

$$\mathcal{P}_j = \begin{cases} \mathcal{P}_C + \mathcal{P}_n^{tx} & \text{if } \mathfrak{Q} = 1 \text{ send packets} \\ \mathcal{P}_C & \text{if } \mathfrak{Q} = 0 \text{ sleep mode active}, \end{cases} \tag{2}$$

where $\mathcal{P}_C$ is the circuit power and $\mathcal{P}_n^{tx}$ is the transmission power consumption on the *nth* channel in CIoT systems.

C. TRANSMIT PACKET RATE MODEL

In this subsection, we explain the SNR boundary value of the Rayleigh fading channel based on the packet loss rate [22]. In this paper, we choose the $2^{\mathbf{\Phi}}$- QAM. When the received SNR exceeds the minimum SNR. The minimum SNR required to achieve the target Bit Error Rate (BER) is derived as $\xi_{k,n} = \frac{1}{b_n}\ln(a_n/BER_{k,n}), n = 1, 2, ..N_{t-H}$, as shown in [23]. In this equation, a_n and b_n represent the modulation and coding scheme levels, respectively. The N_{t-H} represents the maximum number of transmissions in Hybrid Automatic Repeat Request (HARQ). To avoid deep channel fades, no payload bits will be sent, and the received SNR ξ must be lower than ξ_1, for all modulation and coding scheme levels of SNR $[\xi_n, \xi_{n+1}]$ [21]. The BER is estimated based on the calculated SNR and modulation level $BER_{k,n} = 1-(1-2(1-\mathbf{\Phi}^{\frac{1}{2}})\mathcal{Q}(\sqrt{3\xi_{k,n}/(\mathbf{\Phi}-1)}))$, where $\mathbf{\Phi}$ represents noise power and $\mathcal{Q}$ is the Q-function used to find the tail distribution function of the normal criterion distribution. The average PER in the HARQ mode for all SNR values equal to $\xi_{k,n1}, \xi_{k,n2} \cdots, \xi_{k,nN_{t-H}}$, including the number of packet transmissions, which can be expressed as:

$$\begin{aligned}\mathcal{P}_{HARQ}^{PER} = &\sum_{n1}^{N}\sum_{n2}^{N}\cdots\sum_{nN_{t-H}}^{N}\int_{\xi_{1,n1}}^{\xi_{1,n1}+1}\int_{\xi_{2,n2}}^{\xi_{2,n2}+1}\cdots\\ &\cdots\int_{\xi_{k,nN_{t-H}}}^{\xi_{k,nN_{t-H}}+1}\mathcal{P}_{1,2,...k,n1,n2,....,nN_{t-H}}^{PER}\\ &\times\left(\xi_{1,n1},\xi_{2,n2}\cdots,\xi_{k,nN_{t-H}}\right)\mathcal{P}\left(\xi_{1,n1}\right)\ldots\\ &\times\mathcal{P}\left(\xi_{k,nN_{t-H}}\right)d(\xi_{k,nN_{t-H}})\ldots..\\ &\times\left(d\xi_2,n2\right)\left(d\xi_1,n1\right),\end{aligned} \tag{3}$$

where $\mathcal{P}\left(\xi_{1,n1}\right)\ldots\mathcal{P}\left(\xi_{k,nN_{t-H}}\right)d\left(\xi_{k,nN_{t-H}}\right)$ represents the probability of an error occurring in the channel state after the *kth* transmission reaches the maximum number N_{t-H}. The PER can be related to the BER value on the *nth* channel through $\mathcal{P}_{k,n}^{PER} = 1-(1-BER_{k,n})^{L_n^{packet}}$, where L_n^{packet} is the packet size transmitted successfully on the *n th* channel. When a packet is retransmitted, the receiver tries to recover errors by combining them efficiently. To confirm packets established from previous transmissions, $\mathcal{P}_{k,n}^{PER}$ with a combined retransmitted packet can be calculated to perform the BER for the retransmitted packet as:

$$BER_{k,n} = \left[\frac{1-(1-\mathcal{P}_n^{PER})^{1/L_n^{packet}}}{a_{n_{d_n}}2^{d_n}}\right]^{2/d_n}, \tag{4}$$

where $a_{n_{d_n}}$ represents the total number of errors proceeding with the permitted distance of the complication code d_n at the *kth* transmission attempt. The estimated BER for the retransmitted packet is calculated by PER, whereas the BER of the retransmitted packet is not independent of the previously transmitted packet, as shown in (4). The successful transmit packet rate of the *kth* packet transmission on the *n th* channel can be expressed as:

$$\begin{aligned}\Pi_{k,n} &= 1-\mathcal{P}_{k,n}^{PER} = \left(1-BER_{k,n}\right)^{L_n^{packet}}\\ &= 1-(1-2\left(1-\mathbf{\Phi}^{\frac{1}{2}}\right)\mathcal{Q}\left(\sqrt{\frac{3\xi_{k,n}}{\mathbf{\Phi}-1}}\right))^{L_n^{packet}}.\end{aligned} \tag{5}$$

D. TRANSMISSION DELAY MODEL

CRN may suffer from TD problems; to meet the high bandwidth of real-time transmission, there is a need to reduce the average TD in wireless CRN, which consists of two kinds of delay [24]: handoff and retransmission [25]. Retransmission is used to improve reliability and meet performance targets with low power consumption. Moreover, the system mainly focuses on packet delays caused by channel errors and subsequent retransmissions. The delay can be estimated by assuming that one packet must be transmitted $\mathcal{N}_{t-H}$ times at the Medium Access Control (MAC) layer. The packet retransmission delay should be calculated as $\mathcal{T}_{tx}(\mathcal{N}_{t-H}) = (\mathcal{T}_{mac}+\mathcal{T}_{data})(\mathcal{N}_{t-H}+1)$, where $\mathcal{T}_{mac}$ and $\mathcal{T}_{data}$ represent the treating time of handshake in MAC transmission delay and the time required to transmit the data packet [26]. The average delay can be determined based on *nth* channel allocation and the effect of retransmissions, as shown as:

$$\begin{aligned}\tau_{ret} &= \sum_{i=1}^{\mathcal{N}_{t-H}}\left(\mathcal{P}_{k,n}^{PER}\right)^{i-1}\left(1-\mathcal{P}_{k,n}^{PER}\right)\mathcal{T}_{tx}(i-1),\\ &= \sum_{i=1}^{\mathcal{N}_{t-H}}\left(\mathcal{P}_{k,n}^{PER}\right)^{i-1}\left(1-\mathcal{P}_{k,n}^{PER}\right)\\ &\quad\times(\mathcal{T}_{mac}+\mathcal{T}_{data})(i-1).\end{aligned} \tag{6}$$

The number of the retransmission times and the PER are used to determine the maximum retransmission delay τ_{ret}^{max}, as shown in (6). The packet TD is minimized by computing the maximum handoff time of one packet by analyzing the processing time for both loosen and sending packets in the *nth* channel allocation. According to TD's real-time traffic analysis [24], the handoff TDs can be written as (7), shown at the bottom of the next page, where τ_w and τ_p represent the processing time of the handover process. The average TD of a single packet can be computed by adding the average delay for retransmitting and the handoff TD as:

$$\tau_{delay} = \tau_{ret} + \tau_{hand}. \tag{8}$$

High data throughput is estimated based on the average TD required to provide a packet and the impact of the maximum number of real-time retransmissions.

E. THROUGHPUT

Each packet has the same coding rate indicated by γ. The total transmission throughput η (in a bit) and throughput in bits per symbol of the *kth* packet transmitted using the $2^{2\mathbf{\Phi}_k}$-QAM level is $\gamma\times\mathbf{\Phi}_k\times\mathcal{Z}$, where $\mathcal{Z}$ represents the number of symbols per *kth* packet, and $\mathbf{\Phi}_k$ represent the modulation scheme used.

Therefore, the successfully transmitted total throughput for K packets can be written as:

$$\eta_k = \sum_{k=1}^{K} \gamma \times \mathbf{\Phi}_k \times \mathcal{Z}. \tag{9}$$

Improving throughput depends on choosing a high modulation level to transmit more bits per symbol and an adaptive modulation scheme.

F. QOS FOR IMPROVE SMART PACKET TRANSMISSION SCHEDULING

Maximize Mean Opinion Score (MOS), and the handoff scheme can maximize the quality of the transmitted data while minimizing TD by improving the Quality of Experience (QoE) when considering the PER, packet length, channel transmission, and SNR. The MOS is a metric used to access the multimedia UEs perception of the highest quality [27]. The performance of the TS depends on the maximum expected MOS for spectrum handoff, which is achieved by choosing an available channel with the minimum estimated PER and identifying the transmit packet rate that corresponds to a QoE-driven spectrum handoff, which is expressed as:

$$MOS_{k,n} = \frac{\beta_1 - \beta_2 \Pi_{avg} + \beta_3 \eta_{nor}}{1 + \beta_4 \mathcal{P}_{nor} + \beta_5 \tau_{delay}}, \tag{10}$$

where the variables Π_{avg}, η_{nor}, $\mathcal{P}_{nor}$, and τ_{delay} represent the transmit packet rate, the normalized total throughput, normalized low-power consumption level for each device, and average TD, respectively. The variables β_1, β_2, β_3, β_4, β_5 can be obtained by a linear deterioration analysis [28]. From (10), the performance of TS depends on when the CIoT networks periodically generate data packets and transmit the average packets rate in the CIoT networks $\Pi_{avg} = \sum_k^K \sum_n^N \alpha_{k,n} \Pi_{k,n} / (KN)$, where $\alpha_{k,n} \in \{0, 1\}$ represent whether the kth packet is transmitted on the nth channel, in which case $K = 1$; otherwise, $K = 0$. Moreover, by improving the $MOS_{k,n}$, the maximum normalized system throughput can be written as $\eta_{nor} = \eta_k / \eta_{idl}$, where η_{idl} represents the ideal throughput. To reduce the level of power consumption, it can be expressed as $\mathcal{P}_{nor} = \sum_k^K \sum_n^N \alpha_{k,n} \mathcal{P}_{k,n} / \mathcal{P}_{Max} (KN)$, where $\mathcal{P}_{Max}$ represents the maximum consumed power threshold.

The average delay for retransmitting mechanism is affected when the time delay continues to grow, the QoE decay of on-demand throughput is more, and this can be expressed as $\tau_{nor} = \tau_{delay} / \tau_{Tot}$, where τ_{Tot} represents the total TD threshold.

III. PROBLEM FORMULATION

The goal is to maximize MOS by ensuring the performance of TS based on evolutionary conditions. The DNN is used to obtain an optimal policy, which can be achieved by applying DRL. This agent reacts to its environment as a Markov Decision Process (MDP) $(\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{S}$ stands for the state space, $\mathcal{A}$ contains each a potential actions space set, $\mathcal{R}$ is the immediate reward function $\mathcal{S} \times \mathcal{A} \to \mathcal{R}$, and $\mathcal{P}$ is a transition probability function $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0, 1]$. In addition, π is denoted as the decision policy that performs a state to the action $\pi \colon \mathcal{S} \to \mathcal{A}$. The DRL agent exposes $s_t = s \in \mathcal{S}$, where t is an episode, and the agent selects an action $a_t = a \in \mathcal{A}(s_t)$. According to policy π, the agent interacts with an environment through actions. Then, the environment changes into a new state $s_{t+1} = s^{\wedge} \in \mathcal{S}$ with transition probability $\mathcal{P}_{ss^{\wedge}}(a)$ and offers the agent a feedback reward, indicated as $r_t(s, a)$. r is the reward of action on a state and is defined as the predicted MOS of multimedia transmission. The objective of the DRL agent is to maximize the discounted cumulative reward, which can be written as:

$$\mathcal{V}^{\pi}(s) = \mathbb{E}_{\pi} \left\{ \sum_{t=0}^{\infty} \Psi^t r_t (s_t, \pi(s_t)) \mid s_0 = s \right\}, \tag{11}$$

where $\Psi \in (0, 1)$ represents a discount factor and $\mathbb{E}_{\pi}$ is the expected return. From (11), we can determine the state function that follows a policy denoted by $\mathcal{V}^{\pi}(s)$, which can be rewritten as:

$$\mathcal{V}^{\pi}(s) = \mathcal{R}(s, \pi(s)) + \Psi \sum_{s^{\wedge} \in \mathcal{S}} \mathcal{P}_{ss^{\wedge}\pi}(s) \mathcal{V}^{\pi}(s^{\wedge}). \tag{12}$$

The value of the reward is denoted as $\mathcal{R}(s, \pi(s)) = \mathbb{E}\{r(s, \pi(s))\}$. Because evaluating the policy π for reward function $\mathcal{R}(s, \pi(s))$ and transition probability $\mathcal{P}_{ss^{\wedge}}$ in (12) is difficult. We used the Bellman equation to get the optimal policy. $\mathcal{R}(s, \pi(s))$ represents the reward of action on a state and is also described as the predicted MOS of multimedia transmission. In Q-learning, the policy is established by performing the state-action pairs and can be written as:

$$\mathcal{Q}^{\pi}(s, a) = \mathcal{R}(s, \pi(s)) + \Psi \sum_{s^{\wedge} \in \mathcal{S}} \mathcal{P}_{ss^{\wedge}}(a) \mathcal{V}^{\pi}(s^{\wedge}). \tag{13}$$

The predictable discounted cumulative reward begins with s taking action a under the policy π. Consequently, the optimal policy π^* of the value function, indicated by $\mathcal{V}^*$ as shown in (12), can be mathematically written as:

$$\begin{aligned} \mathcal{V}^*(s) &= \mathcal{V}^{\pi^*}(s) = \max_{\pi} \mathcal{V}^{\pi}(s) \\ &= \max_{a \in \mathcal{A}(s)} \Big(\mathcal{R}(s, \pi(s)) \\ &\quad + \Psi \sum_{s^{\wedge} \in \mathcal{S}} \mathcal{P}_{ss^{\wedge}\pi}(a) \mathcal{V}^*(s^{\wedge}) \Big). \end{aligned} \tag{14}$$

Let Q-learning of the value function $\mathcal{Q}^*(s, a) = \mathcal{Q}^{\pi^*}(s, a) = \max_{\pi} \mathcal{Q}^{\pi}(s, a)$ be the optimal action under the

$$\begin{aligned} \tau_{hand} = {} & \left(\mathcal{P}_{k,n}^{PER}\right)^{2\mathcal{N}_{t-H}} \left[\left(\mathcal{P}_{k,n}^{PER}\right)^{4\mathcal{N}_{t-H}} \times \tau_{ret}^{max} + \left\{ 1 - \left(\mathcal{P}_{k,n}^{PER}\right)^{4\mathcal{N}_{t-H}} \right\} \times \left(\tau_{ret}^{max} + \tau_w\right) \right] \\ & + \left\{ 1 - \left(\mathcal{P}_{k,n}^{PER}\right)^{2\mathcal{N}_{t-H}} \right\} \left[\left(\mathcal{P}_{k,n}^{PER}\right)^{4\mathcal{N}_{t-H}} \times \left(\tau_{ret}^{max} + \tau_p\right) + \left(1 - \left(\mathcal{P}_{k,n}^{PER}\right)^{4\mathcal{N}_{t-H}} \right) \times \left(\tau_{ret}^{max} + \tau_w + \tau_p\right) \right] \end{aligned} \tag{7}$$

optimal policy π^*. The reward of action on a state $\mathcal{r}$ is defined as the predicted MOS of multimedia transmission.

$$\begin{aligned}\mathcal{V}^*(\mathcal{s}) &= \max_{\pi} \mathcal{V}^{\pi}(\mathcal{s}) \\ &= \max_{a\in\mathcal{A}(\mathcal{s})} \left(\mathrm{MOS}(\mathcal{s},\pi(\mathcal{s})) \right. \\ &\quad \left. + \Psi \sum\nolimits_{\mathcal{s}^{\wedge}\in\mathcal{S}} \mathcal{P}_{\mathcal{s}\mathcal{s}^{\wedge}\pi}(a)\,\mathcal{V}^*\left(\mathcal{s}^{\wedge}\right)\right).\end{aligned}$$

The optimal policy yielding the highest value of the optimal value function for all sets of actions and states as $\mathcal{V}^*(\mathcal{s}) = \max_{a\in\mathcal{A}(\mathcal{s})}(\mathcal{Q}^*(\mathcal{s},a))$. Moreover, it can be written in terms of the optimal policy as $\pi^*(\mathcal{s}) = \arg\max_{a\in\mathcal{A}(\mathcal{s})}(\mathcal{Q}^*(\mathcal{s},a))$. The optimal Bellman equation can be written as:

$$\begin{aligned}\mathcal{Q}^*(\mathcal{s},a) &= \mathcal{R}(\mathcal{s},a) \\ &\quad + \Psi \sum\nolimits_{\mathcal{s}^{\wedge}\in\mathcal{S}} \mathcal{P}_{\mathcal{s}\mathcal{s}^{\wedge}}(a) \\ &\quad \times \max_{a^{\wedge}\in\mathcal{A}(\mathcal{s}^{\wedge})}(\mathcal{Q}^*\left(\mathcal{s}^{\wedge},a^{\wedge}\right)),\end{aligned} \tag{15}$$

The DRL concept of a learning experience is combined with the reward principle to solve this problem. The DRL concept can be discussed below to maximize the total discount reward function.

- **Agent:** The vision of having an intelligent network running can be achieved by considering the quality of learning based on using information from previous successful experiences to create intelligence in the SDN control panel.
- **System State:** The current situation of the agent is defined as $\mathcal{s} = \left\{\mathcal{S}_{Ch}, \mathcal{S}_{pl}, \mathcal{S}_{cq}, \mathcal{S}_{tl}, \mathcal{S}_{mos}\right\}$, where $\mathcal{S}_{Ch}$ represents the status of the channel (idle or busy), $\mathcal{S}_{pl}$ indicates the priority level that assigns to each of the channels, $\mathcal{S}_{cq}$ shows the quality of the channel (SNR), $\mathcal{S}_{tl}$ represents the traffic load of the chosen channel and $\mathcal{S}_{MOS}$ represents the performance of the TS in the CIoT system in terms of minimizing TD.
- **Action Space:** It is necessary to adjust all policy and to determine its improvement $a = \left\{\mathcal{A}_{po}, \mathcal{A}_{sm}, \mathcal{A}_{am}, \mathcal{A}_{bp}\right\}$, where $\mathcal{A}_{po}$ represents the power consumption control (active or sleep), $\mathcal{A}_{sm}$ denotes the spectrum management access, which should avoid unnecessary waiting time or handoff, A_{am} shows the transmission modulation selection, and $\mathcal{A}_{bp}$ represents the bandwidth allocation in every packet.
- **Reward:** It is designed based on traffic scheduling policies that take URLLC service requests into account. The reward function is used to improve training with probability ratio clipping of MOS, η, Π_{avg}, τ_{delay}, and $\mathcal{P}_{avg}$. Therefore, we offer a new mechanism for URLLC scheduling, called the actor-critic, based on a fuzzy-RBF algorithm, which can schedule and avoid large computations in the learning process.

IV. FUZZY-RBF ALGORITHM BASED ACTOR-CRITIC LEARNING FOR URLLC SCHEDULING

The goal of DRL is to address the problem of intelligent TS and reduce power transmission levels based on the current state of the decision policy. To solve the TS problem under massive transmission packets, we propose a fuzzy-RBF learning algorithm to converge both the action of the actor and the state-action of the critic. Fuzzy-RBF can adjust its stochastic learning policies in CIoT systems under a great dimensional system state. To increase the sum discounted reward and enhance a transmission schedule, depending on calculated Bellman optimality [28] as $\mathbb{J}(\pi) = \int_{\mathcal{s}} \Psi^t P(\mathcal{s}|a,\pi) \int_{\mathcal{A}} \pi_{\phi} \mathcal{Q}^{\pi_{\phi}}(\mathcal{s},a)\, d\mathcal{s}\, da$, where π_{ϕ} relative to the regular, predictable reward per time step under the policy. The fuzzy-RBF consists of three types of layers. In this environment, the state space represents the input of the actor and critic. The output of fuzzy-RBF depends on the estimation of the actor and critic function. The connection weight vector in both the actor and critic learning frameworks is based on estimating software expansion potential, which requires the determination of the hidden layer of the fuzzy-RBF. The UE-specified system state is denoted as $\mathcal{s}_t = \left\{\mathcal{S}_{1,t}, .., \mathcal{S}_{N,t}\right\}^T \in r^{\mathrm{M}}$ at the time step t for the input layer. Every neuron in the input layer represents the input state variable $\mathcal{S}_{\mathrm{M},t}$. After that, each node of the hidden layer signifies the front part of a fuzzy rule, and the output of hidden layers using the Gaussian kernel function is given as:

$$\mathcal{O}_{ji}(\mathcal{s}_t) = \begin{cases} e^{\left[-\frac{\left(\mathcal{s}_j - \mathfrak{x}_{ji}(\mathcal{s}_j)\right)^T\left(\mathcal{s}_j - \mathfrak{x}_{ji}(\mathcal{s}_t)\right)}{2\sigma_{ji}^2}\right]}, & \text{if } \left|\mathcal{s}_j - \mathfrak{x}_{ji}(\mathcal{s}_t)\right| < \sigma_{ji}, \\ 0, & \text{otherwise}, \end{cases} \tag{16}$$

where $\mathcal{s}_j$ represents the pattern of Gaussian kernel function in the $j\,th$ hidden layer node, $\mathfrak{x}_{ji}(\mathcal{s}_t)$ represents the weight vector of the Gaussian kernel function, and σ is the variance controlling the sensitivity of the Gaussian to off-center input. The associations of the hidden layer with output are then learned by squared error minimization as:

$$\begin{aligned}\varphi_i(\mathcal{s}_t) &= \prod\nolimits_{j=1}^{\mathrm{M}} \mathcal{O}_{ji}(\mathcal{s}_t) \\ &= \exp\left(-\sum\nolimits_{j=1}^{\mathrm{M}} \frac{\left(\mathcal{s}_j - \mathfrak{x}_{ji}\left(\mathcal{s}_j\right)\right)^2}{2\sigma_{ji}^2}\right), \quad i = 1, 2, .., \mathrm{M}.\end{aligned}$$

On the other hand, the $i\,th$ hidden layer nodes, the normalized fitness of the fuzzy of every rule [29], provides the following necessary condition as $\mathcal{f}_i(\mathcal{s}_t) = \varphi_i(\mathcal{s}_t) \Big/ \sum_{l=1}^{\mathrm{M}} \varphi_l(\mathcal{s}_t)$. This $i\,th$ node in the hidden layer is capable to achieve it quickly and simultaneously without iterative learning. The fuzzy-RBF learning algorithm for actor and critic is composed in the output layer, representing the actor outputs for the action function as $\mathcal{A}_l(\mathcal{s}_t) = \sum_{i=1}^{\mathrm{M}} \hat{\mathcal{C}}_{ji} \mathcal{f}_i(\mathcal{s}_t)$ and value function

$\mathrm{X}(\mathcal{s}_t) = \sum_{i=1}^{\mathrm{M}} \mathrm{x}_i \mathcal{f}_i(\mathcal{s}_t)$, where the $\hat{\mathcal{C}}_{ji}$ is the weight vector between *i th* hidden layer nodes and *j th* is the output node of the actor-network, and x_i represents the weight vector between *ith* hidden layer nodes and *j th* output node of the critic network. Due to the exploration utilization of "Gaussian interference", the output action $\mathcal{A}_i(\mathcal{s}_t)$ cannot be used directly. So, to achieve the actual action function $\widetilde{\mathcal{A}}_i(\mathcal{s}_t)$, it is necessary to detect and remove the learning progress of inactive hidden layer based on calculating the error between the estimated value and real values in terms of the temporal-difference method as:

$$\mathcal{B}_t = \mathcal{R}(\mathcal{s}_t) + \Psi \mathrm{X}_{\mathrm{x}}\left(\mathcal{s}_t^{\wedge}\right) - \mathrm{X}_{\mathrm{x}}(\mathcal{s}_t). \tag{17}$$

The update temporal-difference error depends on the corresponding weight vector x_i between *i th* hidden layer nodes and *j th* output nodes. To handle a delayed reward, the eligibility trace mechanism in DRL, which connects a weight vector x_i depends on developing the learning process and propagate the temporal-difference error [30]. The fuzzy-RBF learning updates the weight vector x_i and the eligibility trace x_t at the time step$t$; can be written as:

$$\mathcal{n}_t = \sum_{j}^{t-1} (\Psi\lambda)^{t-j} \mu_{\mathrm{x}_i} \mathrm{X}_j(\mathcal{s}_t), \tag{18a}$$

$$\mathrm{x}_i(t+1) = \mathrm{x}_i(t) + \alpha_c \mathcal{B}_t \mathcal{n}_t, \tag{18b}$$

where α_c represent the learning rate for the weights vector of the critic network and $\lambda \in [0, 1]$ represents a decay parameter for the eligibility trace mechanism. While the actor part is at the end of time step t, the policy can be improved by using the update temporal-difference error as:

$$\mathcal{A}_l\left(\mathcal{s}_t^{\wedge}\right) = \mathcal{A}_l(\mathcal{s}_t) + \alpha_a \mathcal{B}_t, \tag{19}$$

where α_a represents a positive parameter of the actor-network. Therefore, improving the policy based on the updating temporal-difference error without requiring the system's prior knowledge provides a better approximation for the actor's action and the current action in the critic part, as shown in (18) and (19).

Sudden increases in arrival rate for all UEs can result in unnecessary costs and long retransmit in extreme cases. At the same time, the DRL needs a transient time to learn the status training data. Though, in the URLLC scenarios for retransmitting data, this transient time will be critical to the execution of the system. From (section IV) The deep-RL algorithm must be able to address more action space in real-time. To solve this problem, a deep distributional RL based on GAN- scheduling was proposed to reduce the size of the action space without limiting it, as shown in (section V).

V. DEEP DISTRIBUTIONAL RL BASED ON GAN-SCHEDULING FRAMEWORK

The proposed GAN -scheduling creates a virtual environment for training DRL agents and operates in highly reliable systems. The agent attempts to obtain the optimal TS based on the distributional perspective on DRL [31], [32] is the random return $\mathcal{Y}$ whose expectation is the value $\mathcal{Q}^{\pi}$. The random

Algorithm I: Training Algorithm of the Proposed Fuzzy-RBF Based Actor-critic Learning.

1- Set learning rate for the weights vector α_c, variance controlling the sensitivity σ_{ji}^2, and λ decay parameter for eligibility trace mechanism in DRL.
2- Determine: initial state $\mathcal{s}_0$, a fuzzy weight vector $\hat{\mathcal{C}}_{ji}$ between *i th* hidden layer nodes and *j th* output node of the actor-network, and x_i weight vector between *i th* hidden layer nodes and *j th* output node of the critic network.
3- **for** all time step $t = 0, 1, 2, \ldots..$ do
4- Perform the actor element receives the measured system state $\mathcal{s}_t$, and uses them to generate new rules if the condition $\mathcal{O}_{ji}(\mathcal{s}_t)$, $i = 1, 2, .., \mathrm{M}$,
5- Achieve the reward of action on a state of the predicted MOS of multimedia transmission $\mathcal{R}(\mathcal{s}_t)$.
6- Calculate the action function: $\mathcal{A}_l(\mathcal{s}_t) = \sum_{i=1}^{\mathrm{M}} \hat{\mathcal{C}}_{ji} \mathcal{f}_i(\mathcal{s}_t)$;
7- Calculate the Gaussian kernel function for the hidden output layer as shown in (16);
8- Calculate the temporal-difference error as shown (17);
9- Update the fuzzy weight $\hat{\mathcal{C}}_{ji}$, the eligibility trace $\mathcal{n}_t$, and update the weight vector x_i as shown in (18a) and (18b);
10- Update decay parameter for eligibility trace $\lambda \in [0, 1]$;
11- **If** $\left|\mathcal{s}_j - \mathrm{x}_{ji}(\mathcal{s}_t)\right| < \sigma_{ji}$
12- The total number of iterations is satisfied, Stop.
13- Calculate $\mathcal{r}$, update temporal-difference error as shown (19) based on the end the time step t;
14- **Else** go to step 4.
15- **end if**
16- **end for**

return achieved by adhering to a current policy π by performing an action a from the state $\mathcal{s}$ indicated by the random variable $\mathcal{Y}_q^{\pi}(\mathcal{s}, a)$ due to the unexpected predictability in the environment; thus, resulting in $\mathcal{Q}^{\pi}((\mathcal{s}, a) = \mathbb{E}\left[\mathcal{Y}_q^{\pi}(\mathcal{s}, a)\right]$ and analogous distributional Bellman equation, that is,

$$\mathcal{Y}_q^{\pi}(\mathcal{s}, a) \triangleq \mathcal{R}(\mathcal{s}, a) + \Psi \mathcal{Y}_q^{\pi}\left(\mathcal{s}', a'\right), \tag{20}$$

where $\mathcal{s}'$ and a' are random nature of the next state-action pair after developing a policy, and $A :\stackrel{\triangle}{=} B$ indicates random variable A has a similar probability law as B. Consequently, the behavior of the policy evaluation for the distributional Bellman operator $\mathcal{T}$ can be defined by

$$\begin{aligned}\mathcal{T}\, \mathcal{Y}_q^{\pi}(\mathcal{s}, a) :\stackrel{\triangle}{=} \mathcal{R}(\mathcal{s}, a) + \Psi \mathcal{Y}_q^{\pi} \\ \times \left(\mathcal{s}', \arg\max_{a' \in \mathcal{A}} \mathbb{E}\left[\mathcal{Y}_q^{\pi}\left(\mathcal{s}', a'\right)\right]\right) \\ \mathcal{s}' \sim P(.\,|\mathcal{s}, a), a' \sim \pi\left(.\,\middle|\,\mathcal{s}'\right).\end{aligned} \tag{21}$$

Our objective is to decrease a statistical distance based on the traditional DRL:

$$\sup_{\mathcal{s},a} \text{dis}\, \mathcal{T}\mathcal{Y}_q^{\pi}(\mathcal{s},a), \mathcal{Y}_q^{\pi}(\mathcal{s},a), \tag{22}$$

where dis(A, B) represents the distance between random variables A and B, which can be restrained by several metrics, such as p-Wasserstein [33] and Kullback-Leibler divergence [31]. The p-Wasserstein metric extends the cumulative distribution functions. For $\mathcal{F}$, $\mathcal{D}$ two cumulative distribution functions over the reals, it is defined as $\mathfrak{d}_P(\mathcal{F},\mathcal{D}) = \inf_{\mathbb{U},\mathbb{V}} \|\mathbb{U}-\mathbb{V}\|_P$, where the infimum is possessed overall pairs of random variables $(\mathbb{U},\mathbb{V})$ with respective cumulative distribution $\mathcal{F}$ and $\mathcal{D}$. By applying the inverse cumulative distribution function, the achieved transform of a random variable $\mathcal{W}$ uniformly distributed on [0, 1] as $\mathfrak{d}_P(\mathcal{F},\mathcal{D}) = \left\|\mathcal{F}^{-1}(\mathcal{W}) - \mathcal{D}^{-1}(\mathcal{W})\right\|_P$. For $P < \infty$, this is more explicitly expressed as the C51 algorithm [31] $\mathcal{Y}_q^{\pi}(\mathcal{s},a)$ using a discrete distribution and attained state-of-the-art performance on Atari 2600 games. The p-Wasserstein between them is given by

$$\mathfrak{d}_P(\mathcal{F},\mathcal{D}) = \left(\int_0^1 \mathcal{F}^{-1}(\mathrm{w}) - \mathcal{D}^{-1}(\mathrm{w}) d\mathrm{w}\right)^{1/P}. \tag{23}$$

Assumed two random variables $\mathbb{U},\mathbb{V}$ with cumulative distribution functions $\mathcal{F}_{\mathbb{U}}$, $\mathcal{F}_{\mathbb{V}}$, can create $\mathfrak{d}_P(\mathbb{U},\mathbb{V}) := \mathfrak{d}_P(\mathcal{F}_{\mathbb{U}}, \mathcal{F}_{\mathbb{V}})$. The optimal possible action value depends on the distributional Bellman optimality operator, a hard contraction in the p-Wasserstein distance and decreasing (22) with p-Wasserstein distance (error). To enhance the action improvement value for each action and decrease random noise's effect in CIoT. We propose GANs to evaluate real data and synthetic data by controlling the generation of real data in real-time.

A. GENERATIVE ADVERSARIAL NETWORK (GAN)

Generative adversarial networks offer a virtual environment for training and experimenting with DRL agents. The GANs train two models: a generative model $\mathcal{G}$ and a discriminative model $\mathbb{D}$. The Wasserstein GAN guarantees the suitability of the discriminator as a 1-Lipschitz function, which is proposed [34], [35] to adopt the gradient retribution and perform as follows:

$$\min_{\mathcal{G}} \max_{\mathbb{D}\in\mathbb{D}} \mathbb{E}_{\text{ж}} \sim \mathrm{P}_{data}[\mathbb{D}(\text{ж})] - \mathbb{E}_{\mathbb{Z}\sim \mathrm{P}_{\mathbb{Z}}}(\mathbb{Z})\left[\mathbb{D}\left(\mathcal{G}(\mathbb{Z})\right)\right] + \mathrm{P}(\gamma), \tag{24}$$

where $\mathbb{D}$ represents the set of 1-Lipschitz functions, ж represent a real data sample, $\mathbb{Z}$ is a random distribution sample, and the probability of packets that have the same coding rate is indicated by $\mathrm{P}(\gamma) = \gamma/2\left(\left\|\nabla_{\text{ж}'}\mathbb{D}(\text{ж}')\right\|_2 - 1\right)^2$, $\text{ж}' = \varepsilon\text{ж} + (1-\varepsilon)\mathcal{G}(\mathbb{Z})$, $\varepsilon \sim \cup(0,1)$, where γ is the gradient penalty coefficient. To handle the output using multiple neural layers in (18), (19) must depend on the flow of a DNN to approximate the state-value distribution.

B. GAN- DDQN BASED ON REWARD CLIPPING TECHNIQUE FOR DISCRIMINATOR NETWORK

According to the problems with large action space, we use the GAN- DDQN algorithm to estimate the value of every action for specific states. The discriminator network $\mathbb{D}$ uses a 1-Wasserstein criterion to decrease the error (distance) between target action-value particles and the estimated action-value particles, as shown in (22) and (23). The current state $\mathcal{S}_t = \mathcal{s}$ and sample $\mathbb{t}$ from the uniform distribution $\cup(0,1)$ are fed to the network $\mathcal{G}$ by the agent at iteration t [36]. To perform the predicted action-value particles (samples), the agent computes $\mathcal{G}(\mathcal{s},a) = (1/\mathcal{N})\sum \mathcal{G}^a(\mathcal{s},\mathbb{t}), \forall a \in \mathcal{A}$, and select the action $a^{\wedge} = \arg\max_a \mathcal{G}(\mathcal{s},a), \forall a \in \mathcal{A}$. Consequently, the agent receives a reward $\mathcal{R}$, and the environment travels to the next state $\mathcal{S}_{t+1} = \mathcal{s}'$. The tuple transition $(\mathcal{s}', a^{\wedge}, \boldsymbol{r}', \gamma)$ is collected into a replay buffer $\mathfrak{B}$, as shown in Fig. 2. The networks $\mathcal{G}$ and $\mathbb{D}$ are updated using every transition tuples in $\mathfrak{B}$ for every $\mathcal{N}$ iterations [37]. From the transition i, the target action-value particles is denoted as $\mathcal{Y}_{\mathrm{i}} = \boldsymbol{r}_{\mathbf{i}}' + \Psi\mathcal{G}'^{a_{\mathrm{i}}^{\wedge}}(\mathcal{s}_i', \mathbb{t}_i)$, where $a_i^{\wedge}$ represent the action of the highest expectation action-value particle, where $a_i^{\wedge} = arg\,\underset{a}{max}(1/\mathcal{N})\sum \mathcal{G}'^{a_i^{\wedge}}(\mathcal{s}_i', \mathbb{t}_i)$. The loss functions are utilized by the agent to train $\mathcal{G}$ and $\mathbb{D}$ networks, respectively:

$$L_{\mathbb{D}} = \underset{\substack{\mathbb{t}\sim\cup(0,1)\\(\mathcal{s},a)\sim\mathfrak{B}}}{\mathbb{E}}\left[\mathbb{D}\mathcal{G}^a(\mathcal{s},\mathbb{t})\right] - \underset{(\mathcal{s},a^{\wedge},\mathcal{R},\mathcal{s}')\sim\mathfrak{B}}{\mathbb{E}} \times \left[\mathbb{D}\mathcal{G}^a(\mathcal{s},\mathbb{t}) + \mathrm{P}(\gamma)\right] \tag{25}$$

$$L_{\mathbb{D}} = -\underset{\substack{\mathbb{t}\sim\cup(0,1)\\(\mathcal{s},a)\sim\mathfrak{B}}}{\mathbb{E}}\left[\mathbb{D}\mathcal{G}^a(\mathcal{s},\mathbb{t})\right] \tag{26}$$

where $\mathrm{P}(\gamma)$ is declared in (24), and $\mathcal{G}^a(\mathcal{s},\mathbb{t})$ is the output of the network parameterized by $\mathbb{t}$ when the input $\mathcal{s}$ is provided. The loss function, as shown in (25), will be high when the discriminator $\mathbb{D}$ can discriminate between the real data distributed according to replay buffer $\mathfrak{B}$. We propose a new reward-clipping mechanism to prevent great variation in the target action value, as shown in (27). The clipping strategy can be formulated as follow:

$$\boldsymbol{r}^{\mathbf{Clip}}(\mathbb{D}) = \mathrm{Clip}(\boldsymbol{r}(\mathbb{D}), 1-\epsilon, 1+\epsilon) \tag{27}$$

where $1-\epsilon$ and $1+\epsilon$ are the thresholds that are manually set. This new reward-clipping is used to measure the difference between the precision of the network $\mathbb{D}$ distinguishing and the optimal action-value particles generated by network $\mathcal{G}$. We assume the ϵ thresholds that partition the transmission schedule increase the utility and set the constant $1+\epsilon$ that are taken as the rewards in RL, whose values are much lower than the utility in the reward. Then, the utility as the reward in RL is followed by clipping to these $1+\epsilon$ constants. If the reward clipping (RC) parameter is large $\boldsymbol{r}$, then it can take a long time for any weights, thus making the process of setting parameters more sophisticated. However, if the RC is small $\boldsymbol{r}$, this can easily lead to disappearing gradients when the number of ϵ thresholds is small.

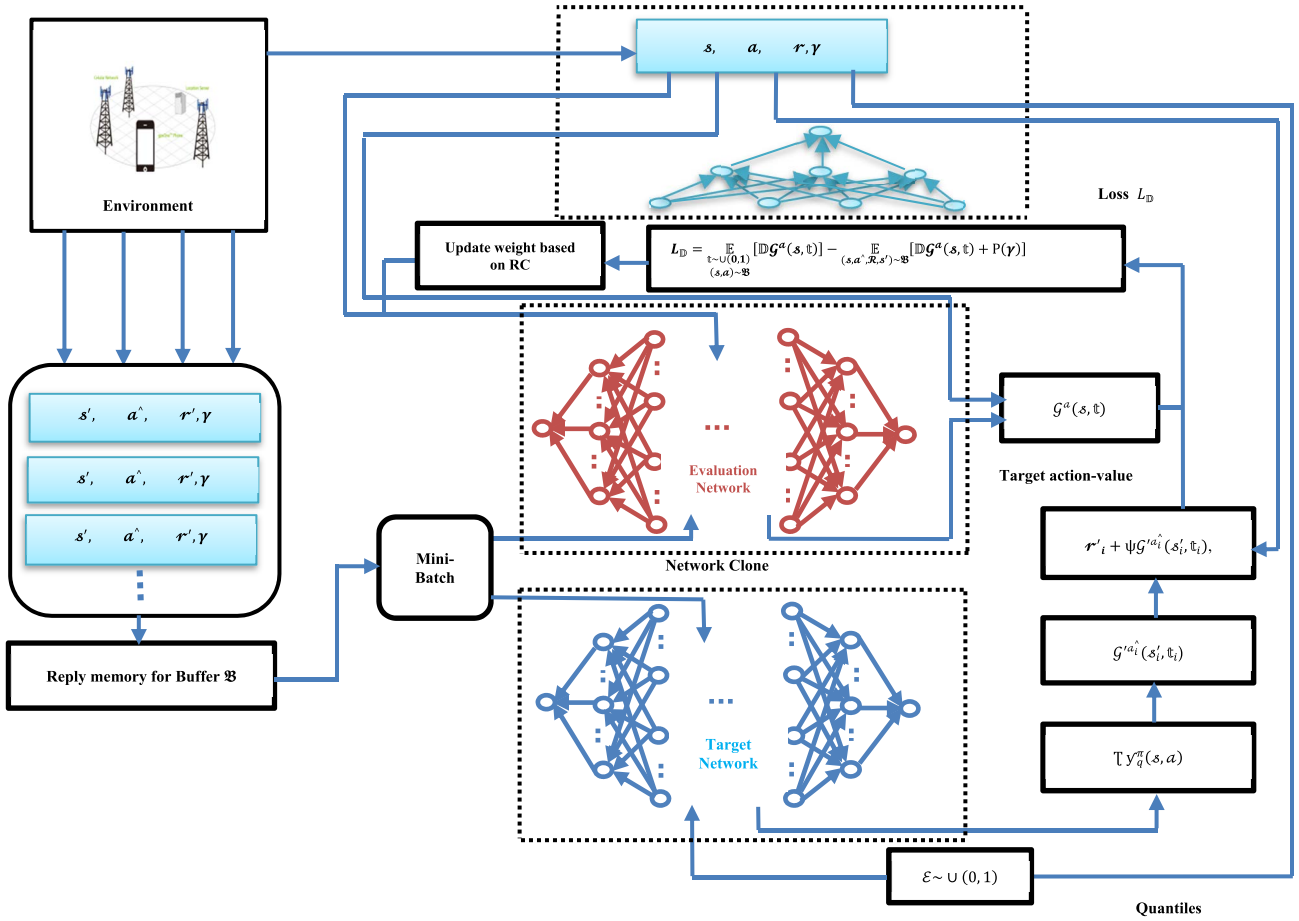

FIGURE 2. The Flow chart of GAN- DDQN.

VI. SIMULATION RESULTS

This section evaluated the performance of our proposed GAN-DDQN algorithm in the CIoT system. The target schemes are compared to GAN-DDQN [37], a standard actor-critic RL algorithm based on the policy gradient for TS[15], and also depend on the analyzed deep Q-learning RL algorithm for TS [19]. URLLC was evaluated under the GAN scheduling by achieving a large real dataset in real-time, and the URLLC packet is small in size. The main simulation parameters are listed in Table 1.

TABLE 1. Simulation parameters.

Parameter	Value
γ	0.8
$\mathcal{P}_c$	5 mW
$\mathcal{P}_n^{tx}$	50 mW
τ_{ret}^{max}	100 ms
$\mathbb{B}_t$	0.1
Time slot (x)	10 ms
URLLC packet size	64
Total bandwidth	40 MHz
Cell radius	500 m
Modulation levels for each IoT	BPSK, 4-QAM, 8QAM, and 16QAM
Number of available $n\ th$ channel	8

A. TRANSMISSION SCHEDULING AND TRANSMISSION DELAY

This section examined the learning process in terms of the GAN-DDQN scheduling learning procedure compared to actor-critic fuzzy-RBF and DQN concerning the power consumption level, throughput, and transmit packet rate value when the rate of normalized packet arrival is 0.5. The performance gap between other algorithms and the GAN-DDQN scheduling learning shows that the GAN-DDQN scheduling becomes more pronounced and effective learning due to the increase in the number of iterative steps. Figure 3 presents normalized throughput, packet transmission, and powerful learning processes. Although the three DRL algorithms achieve similar performance for the normalized throughput, as shown in Fig. 3(a), during the training process, the GAN-DDQN scheduling slightly achieves better performance throughput than the other actor-critic fuzzy-RBF and DQN. Furthermore, from Fig. 3(b), the GAN-DDQN scheduling slightly improves transmit packet rate in fewer

Algorithm II: Enhance Intelligent TS Based on Proposed A GAN- DDQN Algorithm.

1) **Initialize** a generator $\mathcal{G}$ and a discriminator $\mathbb{D}$ with random weights $q_{\mathcal{G}}$ and $q_{\mathbb{D}}$, discount factor Ψ, the number of predicted action-values $\mathcal{N}$, and γ gradient penalty coefficient.
2) **Initialize** the agent iteration $t = 0$, target action-value $\mathcal{G}^{'}$ with weight $q_{\mathcal{G}}$.
3) **Initialize** replay buffer $\mathfrak{B}$ to update every transition tuples for $\mathcal{G}$ and $\mathbb{D}$.
4) **for** transition $i \geq 0$ **do**
5) Select current state $\mathcal{S}_t = \mathcal{s}$ with smallest $\mathcal{G}^{'a_i^{\wedge}}(\mathcal{s}_i^{'}, \mathbb{t}_i)$ that partition the transmission schedule and increases the utility and set the constant $1 + \epsilon$,
6) Get the next state-action by search the predicted action-value $\mathcal{G}^{a}(\mathcal{s}, \mathbb{t})$ from the current state $\mathcal{S}_t = \mathcal{s}$, and agent samples $\mathbb{t}$ from the uniform distribution $\cup(0, 1)$.
7) Perform the predicted action-value for the agent computes $\mathcal{G}(\mathcal{s}, a) = (1, \mathcal{N}) \sum \mathcal{G}^{a}(\mathcal{s}, \mathbb{t}), \forall a \in \mathcal{A}$; and select the action $a^{\wedge} = \arg\max_{a} \mathcal{G}(\mathcal{s}, a), \forall a \in \mathcal{A}$.
8) The agent receiving a reward $\mathcal{R}$, and the environment travel to the next state $\mathcal{S}_{t+1} = \mathcal{s}^{'}$,
9) The tuple $(\mathcal{s}^{'}, a^{\wedge}, \boldsymbol{r}^{'}, \gamma)$ is collected into $\mathfrak{B}$,
10) The agent updates the $q_{\mathcal{G}}$ and $q_{\mathbb{D}}$ of $\mathcal{G}$ and $\mathbb{D}$ using transition tuples in $\mathfrak{B}$ for $\mathcal{N}$ iterations,
11) **for** various transitions $m \geq 0$ in GAN training, **do**
12) Fulfillment the target action; the agent first chooses $\boldsymbol{m}$
13) transitions from $\mathfrak{B}$ as a mini-batch $(\mathcal{s}^{'}, a^{\wedge}, \boldsymbol{r}^{'}, \gamma)$.
14) The target action-value denoted as $\mathcal{Y}_i = \boldsymbol{r}_{\mathbf{i}}^{'} + \Psi \mathcal{G}^{'a_i^{\wedge}}(\mathcal{s}_i^{'}, \mathbb{t}_i)$, the agent expectation action-value $a_i^{\wedge} = \arg\max_{a}(1/\mathcal{N}) \sum \mathcal{G}^{'a_i^{\wedge}}(\mathcal{s}_i^{'}, \mathbb{t}_i)$,
15) Estimate the action value of every action, and the probability of packets that have the same coding rate $\mathcal{K}^{'} = \varepsilon\mathcal{K} + (1 - \varepsilon)\mathcal{G}(\mathbb{Z})$, by set a mini-batch $\varepsilon \sim \cup(0, 1)$.
16) Use the gradient descent to update the weight $q_{\mathbb{D}}$ to $(1/\boldsymbol{m}) \sum_i^{\boldsymbol{m}} L_i$, where $L_i = \mathbb{D}\left(\mathcal{G}^{a_i}(\mathcal{s}_i, \mathbb{t}_i)\right) - \mathbb{D}(\mathcal{Y}_i + \gamma\left(\left\|\nabla_{\mathcal{K}^{'}}\mathbb{D}(\mathcal{K}^{'})\right\|_2 - 1\right)^2$,
17) Use the gradient descent to update the weight $q_{\mathcal{G}}$ to $(-1/\boldsymbol{m}) \sum_i^{\boldsymbol{m}} \mathbb{D}\left(\mathcal{G}^{a_i}(\mathcal{s}_i, \mathbb{t}_i)\right)$,
18) **end for**
19) Set all the transitions in $\mathfrak{B}$ for training and resetting $q^{'}_{\mathcal{G}} = q_{\mathcal{G}}$, by the agent for replicating network $\mathcal{G}$ to $\mathcal{G}^{'}$,
20) Set a new reward-clipping mechanism as shown in (27),
21) Calculates the difference between the precision of the $\mathbb{D}$ distinguishing and the optimal action-value generated by $\mathcal{G}$ by $\boldsymbol{r}^{\boldsymbol{Clip}}(\mathbb{D}) = Clip(\boldsymbol{r}(\mathbb{D}), 1 - \epsilon, 1 + \epsilon)$.
22) Update the iteration index $t \leftarrow t + 1$.
23) Predefined ending condition $(1/\boldsymbol{m}) \sum_i^{\boldsymbol{m}} L_i$, set number of iterations index t is satisfied.
24) **end for**

iterations than actor-critic fuzzy-RBF and DQN. In contrast, the GAN-DDQN scheduling allows it to generate better

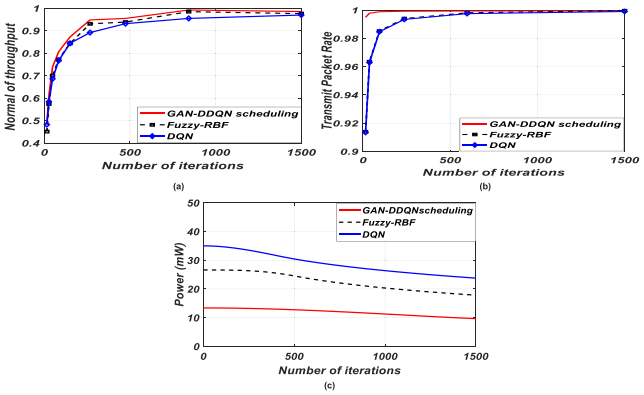

FIGURE 3. Comparison of scheduling scheme a) Normal of throughput, b) Transmit packet rate, c) Power, based on the number of iterations in DRL process for the three algorithms.

candidates with high fitness, while the actor-critic fuzzy-RBF and DQN provide the same performance packet rate during the training process. From Fig. 3(c), increasing the arrival rate to all UEs will necessitate more re-transmissions and increase the number of handoff processes. The lowest transmit power is depicted by applying GAN-DDQN to learn the action value distribution, reducing the difference between the predictable action-value and target action-value distribution. Moreover, the GAN- DDQN scheduling provides the best case of random noise compared to actor-critic fuzzy-RBF and DQN, as shown in Fig. 3(c).

Figure 4 presents the average TD performance by considering transmission power. When the value of power is small, the performance of the average TD for the *nth* channel in CIoT is almost the same good as the optimum. However, when the power level is small, the actual average transmission power of the scheduling packet rate remains constant, the constraint on average transmission power weakens further, and most packets must be transmitted with a very short delay, as shown in (10). When the average packet delay increases, more packets are in the queue waiting for transmission or more packet retransmissions. However, in Fig. 4, increasing the average transmission power does not assist in sending more

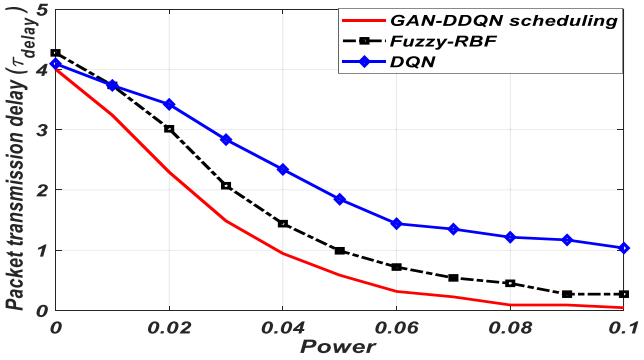

FIGURE 4. The average packet transmission delay against power for different retransmission times.

of the packet transmissions. In addition, the average packet TD curves are nearly flat when the level of transmission power is increased. This is because the packet arrival process determines the average packet TD, not by transmission power level.

B. THROUGHPUT

In this section, we examine the performance of the system's throughput that can be achieved under channel status during the training process. From Fig. 5, the success of the packet arrival depends on minimizing the amount of time it takes for a packet and the difference in packet delay. The long waiting time for the packets reduces throughput, making the packets wait longer to transmit. The normalized throughput decreases when the packet rate increases. The normal throughput is close to 1 with successful training for GAN-DDQN scheduling and achieves a high packet arrival rate, which does not necessitate additional training for various TDs. The packet arrival is randomly selected with a learning rate ranging from 0 to 0.14, and an optimization algorithm is used to obtain successful training for actor-critic fuzzy-RBF and DQN. GAN-DDQN increases intelligence and minimizes error (distance) between target action particles and predicted action-value particles to achieve minimal long-term costs based on signal processing planning and the use of the RC for GAN-DDQN data $\Psi\mathcal{G}^{'\hat{a}_i'}(\mathcal{s}_i^{'},\mathbb{t}_i)$, to distinguish real samples from $\mathcal{G}^{a}(\mathcal{s},\mathbb{t})$ samples produced by reducing the $L_{\mathbb{D}}$ loss to transmit big data. The transition probability is nearly proportionally to the packet arrival rate, as shown in Fig. 6. Figure 6 also shows that as the packet arrival rate increases, the transmission of transition probability of the three algorithms improves. When the packet arrival rate increases, the transition dropping probability increases markedly and influences high traffic load and continues to increase. On the other hand, the high transition probability occurs when more packets are received to provide the optimal TS. When the system's radio resource is fixed under heavy traffic, and more packets arrive, the optimal broadcast planning decision increases linearly as the traffic packet arrival rate increases. Moreover, the big average data improves with the arrival rate when the packet arrival rate exceeds the transmission transition probability. The transition probability depends on the successful transmission

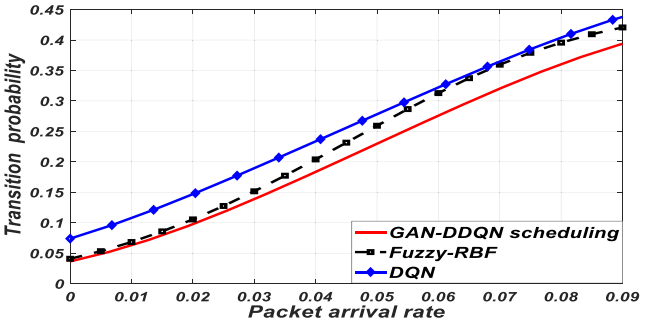

FIGURE 6. Transition probability versus packet arrival rate.

probability of data packets in SNR as shown in (1), and also the average arrival rate, as shown in (11) to (13). Finally, the proposed GAN-DDQN scheduling algorithm has a better transition probability performance than actor-critic fuzzy-RBF and DQN algorithms to improve transmission packet scheduling in the CIoT system.

C. SMALL PACKETS FOR URLLC TO GUARANTEE QOE

In this part, we considered the performance of the proposed algorithms within the scenario that the packet size of the URLLC service is small, as shown in Table 1. We considered two cases: the more bandwidth allocation resolution is either 1 MHz or 200 KHz to guarantee meeting the MOS of URLLC service.

From Fig. 7, the impact of average data rate for different URLLC traffic depends on the GAN-DDQN scheduling, actor-critic architecture for fuzzy-RBF, and DQN algorithm to distribute the URLLC traffic. The GAN-DDQN scheduling algorithm learns the URLLC traffic, improves the channel variations that come with difficulties, and adjusts the URLLC weight dynamically (27), leading to more reliable transmissions based on the proposed new method of new reward-clipping to prevent significant variation in the target action value. The GAN-DDQN scheduling provides an average bit rate of 38 Mbps when the URLLC arrival load is 5 and decreases to 8 Mbps when increasing the average URLLC load to 100 packets/time slot. However, the average big rate obtained by the actor-critic fuzzy-RBF and DQN algorithms varies from 30 Mbps to 19 Mbps when increasing the average

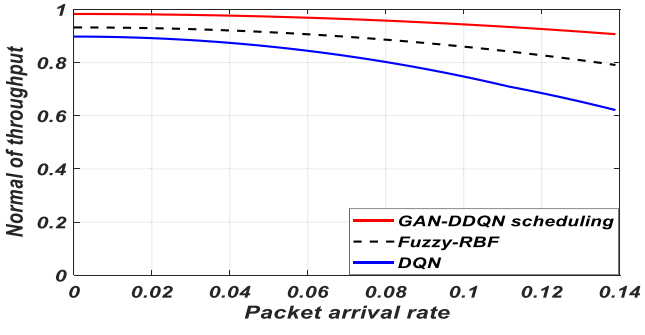

FIGURE 5. Normalized throughput versus packet arrival rate.

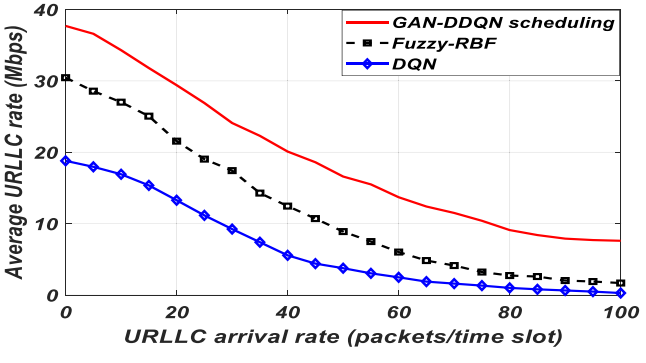

FIGURE 7. Average URLLC rate for different arrival packet rate.

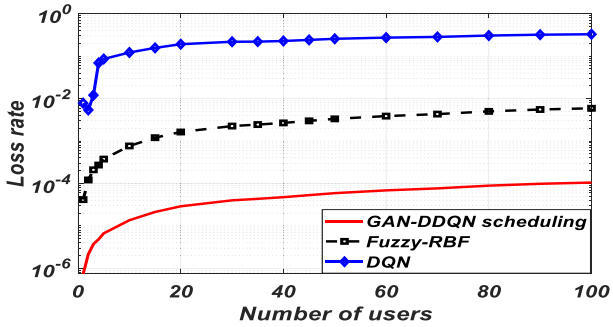

FIGURE 8. Loss rate versus number of UEs.

URLLC load from 5 to 100 packets/time slot. Figure 8 shows the packet losses rate by considering the varying numbers of UEs for three algorithms; the loss rate increases with more UEs. The algorithm of GAN-DDQN scheduling provides a smaller loss rate than actor-critic fuzzy-RBF, and DQN when the number of UEs is less than 100. Furthermore, the control performance loss and enhanced TS depend on enabling discrimination between the real-data distributed according to replay buffer $\mathfrak{B}$, as shown in (25) for stringent URLLC requirements. From Fig. 8, the GAN-DDQN scheduling provides good performance with less loss of rate than actor-critic fuzzy-RBF and DQN algorithms based on applying the RC by increasing the intelligence and decreasing the error between the target action-value and the estimated action-value, as shown in (22) and (23). While the actor-critic fuzzy-RBF provides a lower performance loss rate than GAN-DDQN with an increased number of UEs depends on minimization error for the hidden layer, as shown in (16), and also depends on calculating the error between the estimated value and real values by update temporal-difference error, as shown in (19).

VII. CONCLUSION

In this paper, designing a learning agent with intelligent decision-making ability is challenging in the CIoT system. The smart scheduling in DRL for the RC guarantees a good transmission packet with high reliability. Our proposal investigated the combination of GAN-DDQN used to solve the intelligent TS in CIoT systems. In addition, the proposed RC adopted in GAN-DDQN scheduling improves the training stability with probability ratio clipping of reward, power consumption, transmission packet rate, and throughput. The simulation results show that improving the training stability and increasing the intelligence for the GAN-DDQN scheduling algorithm based on the action-value for the discriminator network for RC decreases the error between the target action-value particles and the estimated action-value particles. Also, the simulation results show that the GAN-DDQN scheduling algorithm has a more significant performance than other DRL algorithms. Our future work will investigate the distributed implementation of our proposed GAN-DDQN based on removing the temporary training time in DRL in the case of unforeseen maximum events that cause failure in URLLC systems.

REFERENCES

[1] A. Bozkurt, "Optimal delay analysis for real-time traffics over IEEE 802.11 wireless LANs," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 1–13, Dec. 2016.
[2] T. N. Weerasinghe, I. A. M. Balapuwaduge, and F. Y. Li, "Preamble reservation based access for grouped mMTC devices with URLLC requirements," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–6.
[3] A. Salh, L. Audah, N. S. M. Shah, A. Alhammadi, Q. Abdullah, Y. H. Kim, S. A. Al-Gailani, S. A. Hamzah, B. A. F. Esmail, and A. A. Almohammedi, "A survey on deep learning for ultra-reliable and low-latency communications challenges on 6G wireless systems," *IEEE Access*, vol. 9, pp. 55098–55131, 2021.
[4] H. Yang, A. Alphones, W. Zhong, C. Chen, and X. Xie, "Learning-based energy-efficient resource management by heterogeneous RF/VLC for ultra-reliable low-latency industrial IoT networks," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5565–5576, Aug. 2020.
[5] X. Chen, H. Zhang, C. Wu, S. Mao, Y. Ji, and M. Bennis, "Optimized computation offloading performance in virtual edge computing systems via deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4005–4018, Jun. 2019.
[6] J. Zhu, Y. Song, D. Jiang, and H. Song, "A new deep-Q-learning-based transmission scheduling mechanism for the cognitive Internet of Things," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2375–2385, Aug. 2018.
[7] H. Yang and X. Xie, "An actor-critic deep reinforcement learning approach for transmission scheduling in cognitive Internet of Things systems," *IEEE Syst. J.*, vol. 14, no. 1, pp. 51–60, Mar. 2020.
[8] I. Ahmad, S. Shahabuddin, T. Sauter, E. Harjula, T. Kumar, M. Meisel, M. Juntti, and M. Ylianttila, "The challenges of artificial intelligence in wireless networks for the Internet of Things: Exploring opportunities for growth," *IEEE Ind. Electron. Mag.*, vol. 15, no. 1, pp. 16–29, Mar. 2021.
[9] J.-H. Park, M. M. Salim, J. H. Jo, J. C. S. Sicato, S. Rathore, and J. H. Park, "CIoT-Net: A scalable cognitive IoT based smart city network architecture," *Hum.-Centric Comput. Inf. Sci.*, vol. 9, no. 1, pp. 1–9, Dec. 2019.
[10] N. Abuzainab, W. Saad, C. S. Hong, and H. V. Poor, "Cognitive hierarchy theory for distributed resource allocation in the Internet of Things," *IEEE Trans. Wireless Commun.*, vol. 16, no. 12, pp. 7687–7702, Dec. 2017.
[11] M. R. Amini and M. W. Baidas, "Performance analysis of URLL energy-harvesting cognitive-radio IoT networks with short packet and diversity transmissions," *IEEE Access*, vol. 9, pp. 79293–79306, 2021.
[12] F. Naeem, S. Seifollahi, Z. Zhou, and M. Tariq, "A generative adversarial network enabled deep distributional reinforcement learning for transmission scheduling in internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4550–4559, Jul. 2021.
[13] R. Li, Z. Zhao, X. Chen, J. Palicot, and H. Zhang, "TACT: A transfer actor-critic learning framework for energy saving in cellular radio access networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 4, pp. 2000–2011, Apr. 2014.
[14] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4585–4600, Jul. 2021.
[15] A. Kaur and K. Kumar, "Intelligent spectrum management based on reinforcement learning schemes in cooperative cognitive radio networks," *Phys. Commun.*, vol. 43, pp. 101226–101239, Dec. 2020.
[16] D. Tarek, A. Benslimane, M. Darwish, and A. M. Kotb, "A new strategy for packets scheduling in cognitive radio Internet of Things," *Comput. Netw.*, vol. 178, pp. 107292–107304, Sep. 2020.
[17] A. Salh, L. Audah, K. S. Kim, S. H. Alsamhi, M. A. Alhartomi, Q. Abdullah, F. A. Almalki, and H. Algethami, "Refiner GAN algorithmically enabled deep-RL for guaranteed traffic packets in real-time URLLC B5G communication systems," *IEEE Access*, early access, Apr. 25, 2022, doi: 10.1109/ACCESS.2022.3170447.
[18] J. Huang, H. Wang, Y. Qian, and C. Wang, "Priority-based traffic scheduling and utility optimization for cognitive radio communication infrastructure-based smart grid," *IEEE Trans. Smart Grid*, vol. 4, no. 1, pp. 78–86, Mar. 2013.
[19] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[20] A. T. Z. Kasgari, W. Saad, M. Mozaffari, and H. V. Poor, "Experienced deep reinforcement learning with generative adversarial networks (GANs) for model-free ultra reliable low latency communication," *IEEE Trans. Commun.*, vol. 69, no. 2, pp. 884–899, Feb. 2021.
[21] Y. Yang, H. Ma, and S. Aïssa, "Cross-layer combining of adaptive modulation and truncated ARQ under cognitive radio resource requirements," *IEEE Trans. Veh. Technol.*, vol. 61, no. 9, pp. 4020–4030, Nov. 2012.
[22] L. Zhao, H. Wang, and X. Zhong, "Interference graph based channel assignment algorithm for D2D cellular networks," *IEEE Access*, vol. 6, pp. 3270–3279, 2018.
[23] D. Wu and S. Ci, "Cross-layer combination of hybrid ARQ and adaptive modulation and coding for QoS provisioning in wireless data networks," in *Proc. 3rd Int. Conf. Quality Service Heterogeneous Wired/Wireless Netw. (QShine)*, Waterloo, ON, Canada, 2006, pp. 47–56.
[24] L. Zhu, F. R. Yu, B. Ning, and T. Tang, "Handoff performance improvements in MIMO-enabled communication-based train control systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 2, pp. 582–593, Jun. 2012.
[25] D. Krishnaswamy, "Game theoretic formulations for network-assisted resource management in wireless networks," in *Proc. IEEE 56th Veh. Technol. Conf.*, Sep. 2002, pp. 1312–1316.
[26] V. Francois-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," pp. 1–14, Nov. 2018, *arXiv:1811.12560*.
[27] A. Khan, L. Sun, E. Jammeh, and E. Ifeachor, "Quality of experience-driven adaptation scheme for video applications over wireless networks," *IET Commun.*, vol. 4, no. 11, pp. 1337–1346, 2010.
[28] Y. Wei, F. R. Yu, M. Song, and Z. Han, "User scheduling and resource allocation in HetNets with hybrid energy supply: An actor-critic reinforcement learning approach," *IEEE Trans. Wireless Commun.*, vol. 17, no. 1, pp. 680–692, Jan. 2018.
[29] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. Boston, MA, USA: Springer, 1981, pp. 1–239.
[30] R. S. Sutton and S. P. Singh, "Reinforcement learning with replacing eligibility tracess," *Mach. Learn.*, vol. 22, pp. 123–158, Jan. 1996.
[31] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, Sydney, NSW, Australia, 2017, pp. 693–711.
[32] G. Barth-Maron, M. W. Hoffman, D. Budden, W. Dabney, D. Horgan, T. B. Dhruva, A. Muldal, N. Heess, and T. Lillicrap, "Distributed distributional deterministic policy gradients," pp. 1–16, Apr. 2018, *arXiv:1804.08617*.
[33] W. Dabney, M. Rowland, M. G. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, New Orleans, LA, USA, Oct. 2018, pp. 2892–2901.
[34] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*.
[35] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs—Enhanced reader," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
[36] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," in *Proc. 35th Int. Conf. Mach. Learn. (ICML)*, Stockholm, Sweden, Jun. 2018, pp. 1774–1787.
[37] X. Jia, M. Zhou, X. Dang, L. Yang, and H. Zhu, "Diversity and delay performance of max link selection relay cooperation systems over non-identical Nakagami-m fading channels," *EURASIP J. Wireless Commun. Netw.*, vol. 2016, no. 1, pp. 29–41, Dec. 2016.

ADEEB SALH received the bachelor's degree in electrical and electronic engineering from Ibb University, Ibb, Yemen, in 2007, and the master's and Ph.D. degrees in electrical and electronic engineering from University Tun Hussein Onn Malaysia, in 2015 and 2020, respectively. He is currently a Postdoctoral Researcher with the Faculty of Electrical and Electronic Engineering, Universiti Tun Hussein Onn Malaysia. From 2007 to 2012, he worked as a Lecturer Assistant with the Yareem Community College. His research interests include 5G, 6G wireless communication, massive MIMO, artificial intelligence (AI), and the Internet of Things (IoT).

LUKMAN AUDAH received the Bachelor of Engineering degree in telecommunications from Universiti Teknologi Malaysia, in 2005, and the M.Sc. degree in communication networks and software and the Ph.D. degree in electronic engineering from the University of Surrey, U.K. He is currently a Senior Lecturer with the Communication Engineering Department, Universiti Tun Hussein Onn Malaysia. His research interests include wireless and mobile communications, internet traffic engineering, network system management, data security, and satellite communications.

MOHAMMED A. ALHARTOMI (Member, IEEE) received the Ph.D. degree in electronic and electrical engineering from Leeds University, U.K., in 2016. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Tabuk. His research interests include wireless and mobile communications, signal processing, optical wireless systems design, and visible light communications.

KWANG SOON KIM (Senior Member, IEEE) received the B.S. *(summa cum laude)*, M.S.E., and Ph.D. degrees in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in February 1994, February 1996, and February 1999, respectively. From March 1999 to March 2000, he was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of California at San Diego, La Jolla, CA, USA. From April 2000 to February 2004, he was a Senior Member of Research Staff with the Mobile Telecommunication Research Laboratory, Electronics and Telecommunication Research Institute, Daejeon. Since March 2004, he has been with the Department of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, where he is currently a Professor. His research interests include signal processing, communication theory, information theory, stochastic geometry applied to wireless heterogeneous cellular networks, wireless local area networks, wireless D2D networks, wireless *ad hoc* networks, and new radio access technologies for 5G. He was a recipient of the Postdoctoral Fellowship from Korea Science and Engineering Foundation (KOSEF), in 1999. He received the Outstanding Researcher Award from the Electronics and Telecommunication Research Institute (ETRI), in 2002, the Jack Neubauer Memorial Award (Best System Paper Award, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY) from IEEE Vehicular Technology Society, in 2008, and the LG R&D Award: Industry-Academic Cooperation Prize, LG Electronics, in 2013. From 2006 to 2012, he served as an Editor for *The Journal of the Korean Institute of Communications and Information Sciences* (KICS). From 2013 to 2016, he served as the Editor-in-Chief for *The Journal of KICS*. Since 2008, he has been serving as an Editor of the *Journal of Communications and Networks* (JCN). From 2009 to 2014, he served as an Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.

SAEED HAMOOD ALSAMHI received the B.Eng. degree from the Communication Division, Department of Electronic Engineering, Ibb University, Yemen, in 2009, and the M.Tech. degree in communication systems and the Ph.D. degree from the Department of Electronics Engineering, Indian Institute of Technology (Banaras Hindu University), IIT (BHU), Varanasi, India, in 2012 and 2015, respectively. In 2009, he worked as a Lecturer Assistant with the Engineering Faculty, Ibb University. After that, he held a postdoctoral position with the School of Aerospace Engineering, Tsinghua University, Beijing, China, in optimal and smart wireless network research and its applications to enhance robotics technologies. Since 2019, he has been an Assistant Professor. He has published more than 80 articles in high reputation journals in IEEE, Elsevier, Springer, Wiley, and MDPI publishers. His research interests include B5G, green communication, green Internet of Things, QoE, QoS, multi-robot collaboration, blockchain technology, and space technologies (high altitude platform, drone, and tethered balloon technologies). Currently, he is a MSCA SMART 4.0 Fellow with the Athlone Institute of Technology, Athlone, Ireland.

FARIS A. ALMALKI received the B.Sc. degree in computer engineering from Taif University, the M.Sc. degree in broadband and mobile communication networks from Kent University, and the Ph.D. degree in wireless communication networks from Brunel University London. He is currently an Associate Professor in wireless communications and drones with the Computer Engineering Department, Taif University, a Research Fellow with the Department of Electronic and Computer Engineering, Brunel University London. He is a member of the IEEE Communication Society. He is a reviewer in many respected journals and publishers, including Springer, IEEE, Elsevier, and Oxford Press. His research interests include unmanned aerial vehicles (UAVs) and satellites and their application in *ad hoc* wireless networks. Besides, topics related to artificial intelligence, the Internet of Healthcare Things, machine learning, encrypted wireless communications, and emerging trends and applications.

QAZWAN ABDULLAH (Member, IEEE) was born in Taiz, Yemen. He received the bachelor's degree in electrical and electronic engineering and the master's degree in electrical and electronic engineering (major in science) from Universiti Tun Hussein Onn Malaysia (UTHM), in 2013 and 2015, respectively. He has more than 40 scientific publications. Currently, he is a Research Assistant with research interests that include control theory, adaptive fuzzy logic controller, mobile communication (5G/6G), fuzzy logic control and its applications, motor drive, electric vehicle, and antenna filter design.

ABDU SAIF (Member, IEEE) received the B.E. degree in electronics communication from Ibb University, Yemen, in 2005, and the M.Sc. degree in project management from the University of Malaya, Malaysia, in 2017. He is currently pursuing the Ph.D. degree in electrical engineering (major in wireless communication) with the Faculty of Engineering, University of Malaya, Kuala Lumpur, Malaysia. He has more than nine years of industrial experience in telecommunications companies. His research interests include wireless networks, 3D coverage by UAV, the Internet of Things, emergency management systems, and public safety communication for 5G.

HANEEN ALGETHAMI (Senior Member, IEEE) received the B.Sc. degree from Taif University, Saudi Arabia, in 2006, and the M.Sc. degree in advanced computing science and the Ph.D. degree in computer science from the University of Nottingham, U.K., in 2012 and 2017, respectively. Since 2018, she has been an Assistant Professor with the Computer Science Department, Taif University. Her research interests include real-world applications of combinatorial problems while using search algorithms and optimization techniques.