Massive Random Access Control for VLC: A Reinforcement Learning Driven Approach

Sicong Liu, *Senior Member, IEEE,* Xiao Tang, and Linqi Song, *Senior Member, IEEE*

***Abstract*—Visible light communication (VLC) has been widely applied to provide dense network access. Usually random access requests are sparse, and the users can be efficiently detected using compressed sensing (CS) based methods. However, in case of bursting network traffic, massive access requests can significantly degrade the performance of user detection. To this end, we propose an intelligent massive random access control scheme, i.e., Sparse Adaptive Random Access (SARA), based on reinforcement learning (RL). Through iterative interactions with the complex and time-varying environment, the proposed scheme of SARA can smartly provide appropriate flow control levels for users with different priorities. Thus, it can not only respond to high-priority users in a timely manner, but also avoid the low detection accuracy caused by massive access requests. The simulation results demonstrate that the proposed scheme outperforms the benchmark schemes in case of high concurrent traffic.**

***keyword*—Visible light communication, random access, compressed sensing, reinforcement learning.**

I. INTRODUCTION

Visible light communication (VLC) technology is often used to provide communication services for dense indoor devices due to its advantages such as free spectrum license, no electromagnetic interference, and large capacity [1]. However, to date, most VLC-related communication schemes are heterogeneous schemes combining radio frequency uplink and VLC downlink, which cannot fully utilize the anti-jamming properties of VLC in the face of massive random access. Therefore, the different characteristics of the channel and the transformation of the communication mode brought about by VLC make its uplink access a problem worth studying [2].

In the uplink of the VLC, the micro base station (mBS) is responsible for detecting and coordinating the active users, so as to meet the service requests of multiple devices. The concept of multi-packet reception is a signal processing method at the physical layer that attempts to decode multiple packets from colliding signals, and much of the literature on random access revolves around this idea. A random access scheme similar to ALOHA is proposed, which obtains the optimal access strategy by establishing a system access conflict graph and evaluating the status of multiple groups of devices [3]. Aiming at maximizing the access success rate of devices with low latency requirements, Sim divides devices into multiple categories, and proposes a priority-based access class barring (PACB) algorithm [4]. In the initial stage of the algorithm, the number of each type of active devices is estimated by the observed random access results, and the flow control factor is decided according to the estimation results latter.

With the increase of smart devices, how to efficiently use the computing resources to coordinate the access requests is the core driving force for the design of massive random access [5]. Fortunately, users' access requests are sporadic most of the time, relying on this sparse property, compressed sensing methods can achieve efficient active user detection at low cost [6]. Based on this sparsity assumption, approximate message passing is used to detect sparse active users, and the characteristics of cooperative multiple-input multiple-output (MIMO) are used to improve the reliability of identifying users at the edge of the cell [7]. Ke *et al.* formulated pseudo-random pilots for uplink access, combining alternate user detection and channel estimation into a multi-measurement based sparse recovery problem [8].

However, CS performance is strongly influenced by the sparsity of the signal of interest, which makes it difficult to accurately detect users during high concurrent traffic periods, resulting in degradation of access efficiency. In view of this situation, it is necessary for us to develop an access and flow control strategy to give priority to the access requests of emergency services, and temporarily restrict those services with higher delay tolerance, so that the sparsity of user access requests can be sustained to guarantee reliable user detection. Nevertheless, the model of massive random access is complex and the mutual influence between different environmental factors is implicit, making it difficult for conventional schemes to adapt to the dynamic and intricate environments. If reinforcement learning (RL) is introduced in this problem, an RL agent can utilize the utility or value function obtained from interacting with the environment to update its strategy, enabling it to adaptively and rapidly make favorable decisions that adapt to the time-varying environments [12].

To this end, in this letter we redesign a dynamic access

This work is supported in part by National Natural Science Foundation of China under grant 62471414 and 62371411, in part by Guangdong Basic and Applied Basic Research Foundation under grant 2024A1515030150, in part by the Natural Science Foundation of Fujian Province of China under grant 2023J01001, and in part by Science and Technology Planning Project of Fujian Province under grant 2022I0001 (*Corresponding Author: Sicong Liu*).

Sicong Liu and Xiao Tang are with the School of Informatics and the National-Local Joint Engineering Research Center of Navigation and Location Services, Xiamen University, Xiamen 361000, China, and also with Shenzhen Research Institute of Xiamen University, Shenzhen 518057, China (E-mail: liusc@xmu.edu.cn).

Linqi Song is with Department of Computer Science, City University of Hong Kong, and also with City University of Hong Kong Shenzhen Research Institute (Email: linqi.song@cityu.edu.hk).

scheme named Sparse Adaptive Random Access (SARA) based on RL, which can customize differentiated access strategies for different user priorities based on time-varying channels and user status in massive random access.

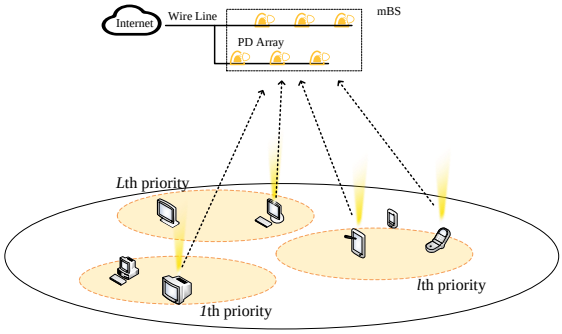

Fig. 1. The massive random access model based on VLC: the mBS receives pilot signals sent by users with different priorities through *M* PDs.

Considering various factors such as access request priority and access efficiency, the scheme intelligently achieves access requests sparsification through the deep deterministic policy gradient (DDPG) algorithm [9]. It dynamically adjusts network parameters through the system utility of environmental feedback, and adaptively tracks the optimal access control strategy.

II. System Model

The massive random access model based on VLC is shown in **Fig. 1**, where an VLC mBS is set up on the ceiling. The number of users within the management range of the VLC mBS is *N*, and each user is assigned a unique pilot sequence $\boldsymbol{\lambda}_n = [\lambda_{n,1}, \lambda_{n,2}, ..., \lambda_{n,P}]^T$ for easy detection by mBS. The mBS deploys *M* photo-diode (PD) arrays with an area A_{R}, which is used to receive signals sent by multiple users. The duration of an access cycle is τ, and the simulation time can be divided into multiple time slots of length τ indexed by *t*. A vector $\boldsymbol{\alpha}^t = [\alpha_1^t, \alpha_2^t, ..., \alpha_N^t]^T$ is designed to represent the users' request status, for example, where $\alpha_n^t = 1$ indicates that the *n*th user has requested access to the mBS, and it is equal to 0 otherwise.

The interaction process between a single PD and multiple users in the whole random access process will be described below. First, the active user sends its own pilot sequence by a small light-emitting diode (LED) array, and the sequence is modulated into an optical signal in the manner of intensity-modulation direct-detection (IM/DD). The pilot sequences of multiple users are superimposed on the *m*th PD through the visible light channel to form an observation vector $\mathbf{y}_m^t \in \mathbb{C}^{P\times 1}$, which is expressed as:

$$\mathbf{y}_m^t = \sum_{n=1}^{N} \alpha_n^t \boldsymbol{\lambda}_n h_{n,m}^t + \mathbf{w}_m^t = \Lambda \mathbf{H}_m^t \boldsymbol{\alpha}^t + \mathbf{w}_m^t = \mathbf{A}_m^t \boldsymbol{\alpha}^t + \mathbf{w}_m^t \cdot \quad (1)$$

where $\Lambda \in \mathbb{C}^{P\times N}$ is the pilot matrix, and $\mathbf{w}_m^t \in \mathbb{C}^{P\times 1}$ is the additive optical noise by the *m*th PD with i.i.d entries $\sim \mathcal{CN}(0, \sigma^2)$. $\mathbf{H}_m^t = \mathrm{diag}(h_{1,m}^t, h_{2,m}^t, ..., h_{N,m}^t)$ is the diagonal matrix with the main diagonal elements are $\{h_{n,m}^t \,|\, 1 \le n \le N\}$, where $h_{n,m}^t$ represents the VLC channel impulse response from the *n*th user to the *m*th PD, following the Lambertian reflection model [10]. The matrix $\mathbf{A}_m^t \in \mathbb{C}^{P\times N}$ is an underdetermined matrix composed of Λ and $\mathbf{H}_m^t$, i.e. perception matrix. Combining perception matrix $\mathbf{A}_m^t$ and observation vector $\mathbf{y}_m^t$, the mBS can obtain the estimated request status vector $\hat{\boldsymbol{\alpha}}_m^t$ on the *m*th PD:

$$\hat{\boldsymbol{\alpha}}_m^t = [\hat{\alpha}_{m,1}^t, \hat{\alpha}_{m,2}^t, ..., \hat{\alpha}_{m,N}^t]^T = f(\mathbf{A}_m^t, \mathbf{y}_m^t) \quad (2)$$

where *f* represents the CS algorithm. For the detection results $\{\hat{\boldsymbol{\alpha}}_m^t \,|\, 1 \le m \le M\}$ of multiple PDs, the final estimated result $\hat{\boldsymbol{\alpha}}^t = [\hat{\alpha}_1^t, \hat{\alpha}_2^t, ..., \hat{\alpha}_N^t]^T$ is generated by the following rules:

$$\hat{\alpha}_n^t = \left\lfloor \frac{2}{M} \sum_{m=1}^{M} \hat{\alpha}_{m,n}^t \right\rfloor, n = 1, 2, ..., N. \quad (3)$$

III. Reinforcement Learning Driven Sparse Adaptive Random Access Control

The user detection accuracy of the CS algorithm is related to two factors: the sparsity of the original signal and the restricted isometry property (RIP) of the perception matrix. This means that under different traffic densities and channel states, there should be an optimal sparsity level for access requests, which can ensure user detection accuracy while accessing as many users as possible [11].

To this end, we design a dynamic hierarchical access scheme to coordinate access requests. The users are divided into *L* priorities, the number of users in each priority is $\{N_l^t \,|\, 1 \le l \le L\}$, and the priority level is represented by $\{r_l \,|\, 1 \le l \le L\}$. At the beginning of each access cycle, the mBS will broadcast a flow control (FC) vector $\mathbf{p}^t = [p_1^t, p_2^t, ..., p_L^t]^T, p_l^t \in (0,1)$ to all users for *access check*. Specifically, if the *n*th user is active and belongs to the *l*th priority, it will generate a random value $q_n \in (0,1)$ before requesting access to mBS, when the access check is passed ($q_n \le p_l^t$), the user will select the current time slot to request access, i.e. $\alpha_n^t = 1$, otherwise wait to repeat the process in the next time slot.

By analyzing the environment to adjust the FC vector in real time, the system can meet the sparsity requirements of the CS algorithm while ensuring the availability of services to high-priority users. However, the ever-changing environment and huge amount of data in massive random access prompt us to adopt some dynamic big data analysis scheme, and the RL-based deep neural networks is obviously a good candidate.

Thus, the proposed SARA scheme realizes the dynamic requests sparsification for massive random access in the time-varying environment. We design a reasonable system utility value to motivate the training of the model and give definitions of various elements in RL, and the proposed

scheme can theoretically converge to the optimal solution through trial-and-error learning from multiple "state-action-feedback" behavior chains.

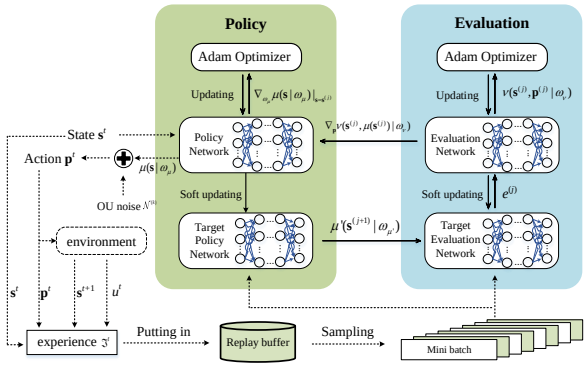

Fig. 2. Flow control framework with environment state as input and FC vector as output. The DDPG framework is composed of policy module, evaluation module and replay buffer, which is used to analyze massive dynamic data.

The RL framework mainly consists of three elements: state $\mathbf{s}^t$, action $\mathbf{p}^t$ and the feedback, i.e. system utility u^t, and the specific definitions are as follows.

1)Issued action: The action is defined as the FC vector $\mathbf{p}^t=[p_1,p_2,...,p_L]^T$ sent by mBS, which is used to control the user's access behavior. The set of active users passing the access check is $\mathcal{A}^t=\{n\,|\,\alpha_n^t=1,1\le n\le N\}$, and the number of elements N_{e}^t in it can be approximately expressed as:

$$N_{\mathrm{e}}^t\approx \mathrm{card}(\mathcal{A}^t)\approx\sum_{l=1}^{L}N_l^t p_l^t \tag{4}$$

where card() represents the number of elements in the set.

After the user cluster requests access to the mBS, the user detection result of the mBS is $\hat{\boldsymbol{\alpha}}^t$. mBS sends confirmations to users in $\mathcal{A}_{\mathrm{e}}^t=\{\hat{\alpha}_n^t=1|1\le n\le N\}$, and the number of successfully established connections is:

$$N_{\mathrm{c}}^t=\mathrm{card}(\mathcal{A}^t\bigcap\mathcal{A}_{\mathrm{e}}^t). \tag{5}$$

Then, the user detection accuracy can be calculated by $c^t=N_{\mathrm{c}}^t/N_{\mathrm{e}}^t$.

2)State formulation: The current FC vector, user detection accuracy and channel impulse response are packaged as the environment state at the next time slot, which is expressed as:

$$\mathbf{s}^{t+1}=[\mathbf{p}^t,c^t,\{\mathbf{H}_m^t\,|1\le m\le M\}]. \tag{6}$$

3)The system utility u^t is expressed as:

$$u^t=c^t\sum_{l=1}^{L}p_l^tN_l^tr_l-\rho_1\left[\frac{1}{L}\sum_{l=1}^{L}(p_l^t-\overline{p^t})^2\right]^{1/2}-\rho_2\left[1-(c^t)^2\right] \tag{7}$$

The utility u^t is an incentive in the optimization process, and its setting takes into account three factors: First, the first item means that we want to make more successful accesses of high-priority users, which can be obtained by combining the detection accuracy, the number of users per priority, the FC vector and the current priority level; The second item is the standard deviation of the internal elements in FC vector, which is weighted by a non-negative value ρ_1, and it can prevent the system from completely ignoring low-priority user access; In addition, we also need to penalize the case of low user detection accuracy, represented by the third term $\rho_2(1-(c^t)^2)$, where ρ_2 represents the weight of this term in the system utility.

The training model is devised in the framework of DDPG, includes current policy network $\mu(\mathbf{s}\,|\,\omega_\mu)$, current evaluation network $\nu(\mathbf{s},\mathbf{p}\,|\,\omega_\nu)$ and their respective target networks $\mu'(\mathbf{s}\,|\,\omega_{\mu'})$ and $\nu'(\mathbf{s},\mathbf{p}\,|\,\omega_{\nu'})$, in which the current network is used for real-time update, and the target network is used to stabilize the training process. Specifically, as shown in **Fig. 2**, its operation process is summarized in **Algorithm 1**:

Algorithm 1. The Proposed SARA Algorithm

Initialize:

$\mu(\mathbf{s}\,|\,\omega_\mu)$, $\mu'(\mathbf{s}\,|\,\omega_{\mu'})$, $\nu(\mathbf{s},\mathbf{p}\,|\,\omega_\nu)$ and $\nu'(\mathbf{s},\mathbf{p}\,|\,\omega_{\nu'})$

Rest replay buffer $\mathcal{R}$.

For episode=1,2,3...max_episode **do**

Randomly generate an initial state $\mathbf{s}^1$.

For t=1,2,3...max_step **do**

Get current state $\mathbf{s}^t$, and feed it into policy network

Policy network outputs action $\mathbf{p}^t=\mu(\mathbf{s}\,|\,\omega_\mu)+\mathcal{N}^{(k)}$, which acts on environment, obtaining feedback information

Calculate current system utility u^t by (7)

Construct next state $\mathbf{s}^{t+1}=[\mathbf{p}^t,c^t,\{\mathbf{H}_m^t\,|1\le m\le M\}]$

Pack current experience $\mathfrak{I}^t=\{\mathbf{s}^t,\mathbf{p}^t,u^t,\mathbf{s}^{t+1}\}$ and store it into replay buffer $\mathcal{R}$

Sample mini-batch $\{\mathfrak{I}^{(j)}\,|1\le j\le J\}$

Update current network weights ω_ν and ω_μ by

$$e^{(j)}=u^{(j)}+\gamma\nu'(\mathbf{s}^{(j+1)},\mu'(\mathbf{s}^{(j+1)}\,|\,\omega_{\mu'})\,|\,\omega_{\nu'}),$$

$$\omega_\nu\leftarrow\arg\min_{\omega_\nu}\frac{1}{J}\sum_{J}(e^{(j)}-\nu(\mathbf{s}^{(j)},\mathbf{p}^{(j)}\,|\,\omega_\nu))^2,$$

$$\omega_\mu\leftarrow\arg\max_{\omega_\mu}\frac{1}{J}\sum_{J}\nabla_{\mathbf{p}}\nu(\mathbf{s},\mathbf{p}\,|\,\omega_\nu)|_{\mathbf{s}=\mathbf{s}^{(j)},\mathbf{p}=\mu(\mathbf{s}^{(j)})}*\nabla_{\omega_\mu}\mu(\mathbf{s}\,|\,\omega_\mu)|_{\mathbf{s}=\mathbf{s}^{(j)}}.$$

If (t *mod* T == *0*) then

Soft update target network weights $\omega_{\nu'}$ and $\omega_{\mu'}$ as

$$\begin{cases}\omega_{\nu'}=\varsigma\omega_\nu+(1-\varsigma)\omega_{\nu'}\\ \omega_{\mu'}=\varsigma\omega_\mu+(1-\varsigma)\omega_{\mu'}\end{cases}$$

End

End

End

IV. Simulation Results

In the VLC random access paradigm considered in this paper, the number of PDs in the mBS is *M*=16. The total number of users managed by a single mBS is *N*=256, and these users are equally divided into *L*=2 priorities (r_1=1, r_2=4). In the training stage, the weights ρ_1 and ρ_2 in system utility

This article has been accepted for publication in IEEE Photonics Technology Letters. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/LPT.2025.3533587

are set to 20 and 40, respectively. The size of the replay buffer

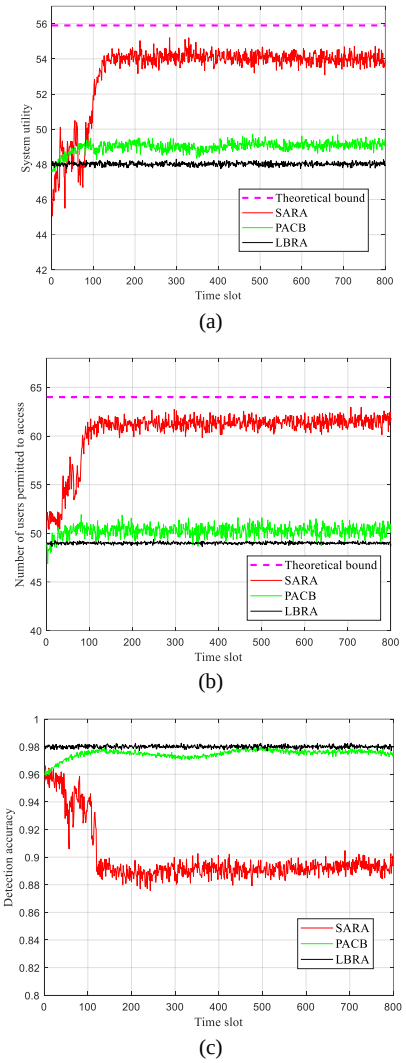

Fig. 3. Performance comparison between SARA and the benchmark schemes in (a) system utility, (b)access number of users allowed to request for access and (c) the detection accuracy of user.

R is set to 10000, and the mini-batch size J=64. The learning rate of policy network and evaluation network are 10^{-3} and 2×10^{-3}. The discount rate $\gamma = 0.9$, the soft update interval of the target network is T=10, and the soft update factor is $\varsigma = 0.02$.

We considered the following three schemes in our simulations: 1) Lasso-based RA (LBRA) [6]: There is no user access management, and the problem of user access detection is solved by multi-group Lasso; 2) Priority-based access class barring (PACB) [4]: A flow control algorithm with two stages; 3) The proposed SARA: A dynamic sparse adaptive approach to trade-off priority-based random access and network traffic.

Fig. 3 shows the optimization of the three algorithms of LBRA, PACB and SARA when 25% of the users are active. As can be seen from Fig. 3(a), since LBRA has no access management strategy, the system utility will drop significantly. SARA produces the highest utility u=53.8, which is 9.5% higher than that of PACB. Fig. 3(b) shows that in terms of the number of users allowed to request for access, PACB converges to 50.4, while SARA is 21.8% higher than PACB because it will try to allow more high-priority users to request access during training. As shown in Fig. 3(c), the access success rate of PACB is 97.5%, while our proposed scheme is only 89.1%. In LBRA, strictly controlled user access avoids multi-user collision and leads to a higher user detection accuracy.

V. Conclusion

In this letter, we proposed a sparse adaptive random access algorithm for massive random access in VLC. To cope with the conflicts when active users request access simultaneously, we start to limit the concurrent access of users with multiple priorities. We propose a SARA scheme based on deep RL and CS. During the optimization process, SARA trains the network according to the environmental feedback and performs real-time random access management based on flow control. SARA implements differentiated access management for different priorities. Simulation results have verified its superiority compared with the benchmark schemes.

References

[1] H. Yang, W. -D. Zhong, C. Chen, A. Alphones and P. Du, "QoS-Driven Optimized Design-Based Integrated Visible Light Communication and Positioning for Indoor IoT Networks," in IEEE Internet of Things Journal, vol. 7, no. 1, pp. 269-283, Jan. 2020.

[2] Z. Wei et al., "Optical Uplink, D2D and IoT Links Based on VCSEL Array: Analysis and Demonstration," in Journal of Lightwave Technology, vol. 40, no. 15, pp. 5083-5096, 1 Aug.1, 2022.

[3] L. Zhao, X. Chi and S. Yang, "Optimal ALOHA-Like Random Access With Heterogeneous QoS Guarantees for Multi-Packet Reception Aided Visible Light Communications," in IEEE Transactions on Wireless Communications, vol. 15, no. 11, pp. 7872-7884, Nov. 2016.

[4] Y. Sim and D. -H. Cho, "Performance Analysis of Priority-Based Access Class Barring Scheme for Massive MTC Random Access," in IEEE Systems Journal, vol. 14, no. 4, pp. 5245-5252, Dec. 2020.

[5] L. Qian, X. Chi and L. Zhao, "Hybrid Access Algorithm for eMBB Terminals with Heterogeneous QoS in MPR Aided VLC System," 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 2018, pp. 1-6.

[6] J. Ahn, B. Shim, and K. B. Lee, "EP-Based Joint Active User Detection and Channel Estimation for Massive Machine-Type Communications," IEEE Transactions on Communications, vol. 67, no. 7, pp. 5178–5189, Jul. 2019.

[7] Z. Chen, F. Sohrabi and W. Yu, "Multi-Cell Sparse Activity Detection for Massive Random Access: Massive MIMO Versus Cooperative MIMO," in IEEE Transactions on Wireless Communications, vol. 18, no. 8, pp. 4060-4074, Aug. 2019.

[8] M. Ke, Z. Gao, Y. Wu, X. Gao and R. Schober, "Compressive Sensing-Based Adaptive Active User Detection and Channel Estimation: Massive Access Meets Massive MIMO," in IEEE Transactions on Signal Processing, vol. 68, pp. 764-779, 2020.

[9] Y. Wei, F. R. Yu, M. Song and Z. Han, "Joint Optimization of Caching, Computing, and Radio Resources for Fog-Enabled IoT Using Natural Actor–Critic Deep Reinforcement Learning," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 2061-2073, April 2019.

[10] L. E. M. Matheus, A. B. Vieira, L. F. M. Vieira, M. A. M. Vieira and O. Gnawali, "Visible Light Communication: Concepts, Applications and Challenges," *IEEE Commun. Surv. Tutor.*, vol. 21, no. 4, pp. 3204-3237, 4nd Quart. 2019.

[11] S. Qaisar, R. M. Bilal, W. Iqbal, M. Naureen and S. Lee, "Compressive sensing: From theory to applications, a survey," in Journal of Communications and Networks, vol. 15, no. 5, pp. 443-456, Oct. 2013.

[12] N. Jiang, Y. Deng, A. Nallanathan, and J. A. Chambers, "Reinforcement Learning for Real-Time Optimization in NB-IoT Networks," in IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1424–1440, Jun. 2019.