SIMULTANEOUS BEAM SELECTION AND USERS SCHEDULING EVALUATION IN A VIRTUAL WORLD WITH REINFORCEMENT LEARNING

Ilan Correa[1], Ailton Oliveira[1], Bojian Du[2], Cleverson Nahum[1], Daisuke Kobuchi[2], Felipe Bastos[1], Hirofumi Ohzeki[2], João Borges[1], Mohit Mehta[3], Pedro Batista[4], Ryoma Kondo[2], Sundesh Gupta[3], Vimal Bhatia[3], and Aldebaro Klautau[1]
[1]Universidade Federal do Pará - LASSE — www.lasse.ufpa.br, Av. Perimetral S/N, Belém, Pará , Brazil, [2]Team MLAB-RL, Morikawa Narusue Laboratory, The University of Tokyo, Japan, [3]Team IITI-RL, Indian Institute of Technology Indore, India, [4]Ericsson Research, 164 80 Stockholm, Sweden

NOTE: Corresponding author: Ilan Correa, ilan@ufpa.br

Abstract – *The fifth generation of mobile networks evolved to serve applications with distinct requirements, which results in a high management complexity due to simultaneous real-time tasks. In the physical layer, code words that allow proper data exchange between the Base Station (BS) and the served users must be chosen. While, in higher layers, the BS must choose users to be served in a given transmission opportunity. There are approaches based on Machine Learning (ML) to solve these combined tasks. However, due to the high amount of possible inputs, a challenge is the availability of data to train the models. In some cases, there may not even exist a predefined optimal answer to use as a "label" for supervised approaches. In this paper, we evaluate solutions for the combined problems of beam selection and user scheduling with Reinforcement Learning (RL), which does not need labels, as a solution for problems without a predefined answer. The algorithms were proposed for Problem Statement 6 of the challenge organized by the International Telecommunication Union (ITU) in 2021, which ranked as the finalists. We compare the approaches in relation to the cumulative reward received by the agents and show a performance comparison of different RL approaches by comparing them with baselines developed for the challenge. The paper also shows how the action taken by the trained agents affect network operation by comparing the number of packets transmitted, which is highly related to the proper selection of users and code words.*

Keywords – Beam selection, reinforcement learning, user scheduling, virtual world

1. INTRODUCTION

The fifth-generation (5G) and beyond of the mobile wireless communications envisages, among other features, higher data rates with the use of greater bandwidths. Due to the scarcity of available spectrum at the currently mostly used sub-6 GHz frequencies, wider bandwidths are being reserved for mobile communications at millimeter Wave (mmWave) bands, such as 28 GHz and 60 GHz [1]. A drawback of the mmWave bands is the higher attenuation in comparison to sub-6 GHz frequencies. Thus, Multiple-Input Multiple-Output (MIMO) techniques are among the core technologies of 5G development at mmWave bands, since they provide better directionality of the electromagnetic wave, allowing to circumvent the high path attenuation [2]. MIMO can also allow increasing system capacity over the same available time-frequency resources, increasing significantly the spectral efficiency [3].

On top of the previously described MIMO-based Physical Layer (PHY), the Base Station (BS) must perform efficiently the so-called Radio Resource Allocation (RRA) [4] or users scheduling to serve the users. These devices can be classified into one or more use cases of the 5G networks, namely: enhanced Mobile Broadband (eMBB), Ultra-Reliable Low-Latency Communications (URLLC), and massive Machine Type Communications (mMTC).

In other words, these networks must serve devices with very distinct requirements, such as the Internet of Things (IoT), terrestrial vehicles, Unnamed Aerial Vehicles (UAVs), pedestrians, and infrastructure. Furthermore, users' mobility and interactions with the environment make the task even harder to solve.

These devices may have data available from several sensors, which could eventually be available to optimize the MIMO and the user scheduling operations [5, 6]. As an example, there is a trend toward autonomous vehicles, which can take advantage of the increasing connectivity options, resulting in applications such as Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-everything (V2X) communications. These vehicles can deploy devices, such as cameras, Light detection and ranging (Lidar), Global Navigation Satellite System (GNSS), etc. The sensors are related, for example, to detection of pedestrians and other vehicles, interpretation of signaling on the streets, automatic and semi-automatic driving, and so on.

Given this possible high amount of data and the increasing difficulty of the several tasks performed in the network, ML techniques have been adopted in several works [7, 8], especially Deep Neural Networks (DNNs). DNNs are optimized for an application, in general, with supervised learning approaches, which may require a prohibitive

amount of data, depending on the model being trained. However, for problems with high-dimensional input data, such as for joint users scheduling and beam selection, there may not exist a predefined optimal answer to use as a "label" for the supervised approach. Therefore, RL-based techniques [9, 10] can be adopted for these cases, which still require many examples, but relaxes the need for labels.

This trend confirms the need for data to properly evaluate scenarios of interest and train the ML models. At the PHY level, similar to the situation faced in 5G, the 6G measurement campaigns will require expensive equipment to support ultra-massive MIMO and terahertz frequency bands. In this case, simulation methodologies for generating communication channels can provide data in controlled conditions to fill the gap until measurements are available. It is also desirable to mimic users' behavior, such as data consumption and movement, which would allow evaluations in more realistic scenarios. To address this challenge, there has been research exploring the use of virtual worlds to generate datasets by creating environments for communications in general [11], and AI/ML applied to 5G/6G [5]. Virtual worlds leverage the fact that 5G and beyond systems will benefit from rich contextual information to improve performance and reduce the loss of radio resources to support its services [12, 13, 14].

In this context, we present in this paper reinforcement learning approaches to the simultaneous problems of user scheduling and beam selection. The goal is to serve efficiently a UAV, vehicles, and pedestrians, composing a scenario with aerial and terrestrial User Equipment (UE). It is assumed an RL agent is executed at the BS, which should periodically take actions based on the information captured from the environment. The information includes channel estimates, buffer status, and positioning data, such as orientation and coordinates from a GNSS. The RL agent receives a reward based on the service provided to the users. The presented algorithms were proposed for Problem Statement 6 (PS-006)[1] of the challenge organized by the ITU in 2021. The teams that proposed the algorithms in this paper ranked as the finalists in this contest.

The contributions of this paper are the design and evaluation of reward functions, two extrinsic and one intrinsic. The extrinsic reward is given to the agents by the environment, while the intrinsic is given to the agents by the agent itself. Another contribution is the evaluation of several state-of-the-art RL algorithms to the relevant problem of joint user scheduling and beam selection in the context of 5G mobile networks. The proposed rewards and the RL algorithms are presented and compared considering the cumulative reward received by the proposed RL-based algorithms, and in relation to how they affect the data rate achieved in the downlink.

[1]ITU-ML5G-PS-006: ML5G-PHY-Reinforcement learning: scheduling and resource allocation

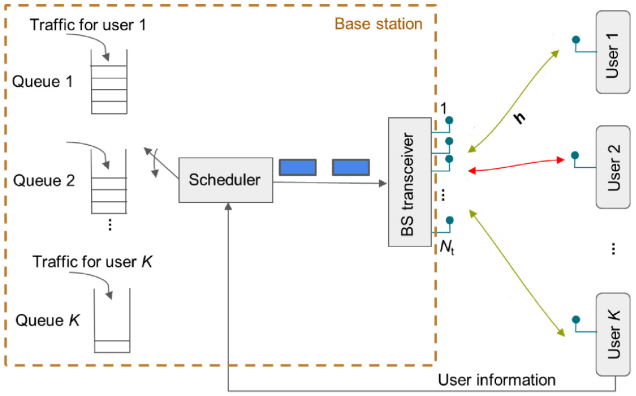


Fig. 1 – Representation of a BS buffering incoming packets, which should be sent to users using an UPA array and beamforming.

This paper is organized as follows. Section 2 presents the modeling of the users scheduling and beam selection targeted in this paper. Section 3 discusses the dataset used in this paper, which was generated with the Communication networks and Artificial intelligence immersed in Virtual or Augmented Reality (CAVIAR) methodology [15] and post-processed to generate additional data to represent the environment. Section 4 presents the RL-based approaches to solve the problem and the proposed reward functions. Section 5 presents experimental evaluations considering the reward received by the RL agents from the environment. Lastly, Section 6 presents the final remarks of this paper.

2. SYSTEM MODEL

Fig. 1 shows the model of the system evaluated in this paper. We assume the *downlink* of a mmWave mobile communication system with MIMO transmission and reception capabilities and K users. We adopt a Uniform Planar Array (UPA) with N_t antenna elements at the BS with $\lambda/2$ spacing. The BS uses beamforming to transmit signals to a user, such that the received signal at the k-th user is

$$y_k = \mathbf{w}_p^H \mathbf{H} \mathbf{f}_q \mathbf{s}, \qquad (1)$$

where $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ is the channel matrix, $\mathbf{f}_q$ is a $N_t \times 1$ precoding vector used at the BS to perform the beamforming, $\mathbf{w}_p$ is a $N_r \times 1$ (combining) vector used at the UE to combine the received signal, with $(\cdot)^H$ denoting the conjugate transpose, and $\mathbf{s}$ is the transmitted symbol.

The vectors $\mathbf{f}_q$ and $\mathbf{w}_p$ are chosen from the codebooks $\mathcal{C}_t = \{\mathbf{w}_1, \ldots, \mathbf{w}_{|\mathcal{C}_t|}\}$ and $\mathcal{C}_r = \{\mathbf{f}_1, \ldots, \mathbf{f}_{|\mathcal{C}_r|}\}$, respectively, where $|\mathcal{C}_t|$ and $|\mathcal{C}_r|$ are the cardinalities of the codebooks. The indexes q and p can be represented by a unique index $i \in \{0, 1, \cdots, M-1\}$, where $M \leq N_t N_r$. Thus, the *optimal* beam index $\hat{i}$ can be referred to as

$$\hat{i} = \arg \max_{i \in \{1, \cdots, M\}} |y_i|. \qquad (2)$$

Fig. 1 illustrates how the BS handles incoming packets in the higher layers. The BS serves the K users of the system by keeping a set of K buffers, i.e., one for each user. These buffers store incoming packets until they can be sent to the destination through the air interface. The BS can only serve one user at a given instant, thus it should point a given beam to the user that should be served in an instant of time. This scenario illustrates the problems of **scheduling** and **beam selection** in which the BS must schedule a user and choose the beam index that points to it. The incoming packets for each user are buffered until they can be sent through the air interface if there is buffer space available. The excess packets are tail-dropped. The packets are also dropped when they occupy the buffer for more than a given amount of time.

In this context, the goal is to implement an agent that is executed at the BS and that periodically chooses a user and the beam that points to that user. The users can be a UAV, vehicles, and pedestrians, composing a scenario with aerial and terrestrial UEs. The agent can use information captured from the environment, such as channel estimates, buffer status, and positions from a GNSS [16, 17, 18]. For that, it is assumed the BS maintains a database with information about the UEs, which can be used to help on the task.

All the information stored in the database, such as UEs' locations, could be acquired and reported to the BS through, for example, an anchor cell operating at Long-Term Evolution (LTE) frequencies [19]. This anchor cell operates at sub-6 GHz or another band less impacted by the propagation attenuation, which can provide a more reliable connection to report the GNSS data and other control information. In turn, the mmWave link is used to achieve very high data rates when a reliable signal is detected. We assume in this paper the additional data is readily available at the BS through such an additional connection, but we do not model it.

3. THE CAVIAR METHODOLOGY

To simulate the environment presented in the previous section, we used the CAVIAR methodology [5], which is composed of the processes shown in Fig. 2. CAVIAR is based on Unreal, Airsim, and a set of scripts in Python that process the outputs of the former software. In this way, the methodology allows generating UEs' movement in a tri-dimensional (3D) virtual world with Unreal and Airsim. Then, the electromagnetic propagation and the network traffic can be generated based on UEs' position to create additional information about the environment. Lastly, ML models can be trained with the generated data and an already trained ML model can be used to interact with the 3D environment. CAVIAR's processes related to the positions, electromagnetic propagation, and network traffic are described in this section. Table 1 and Table 2 summarize all the spatial information, the network traffic, and electromagnetic propagation that can be generated with the CAVIAR methodology. The code that implements the processing of Fig. 2 is available on Github[2].

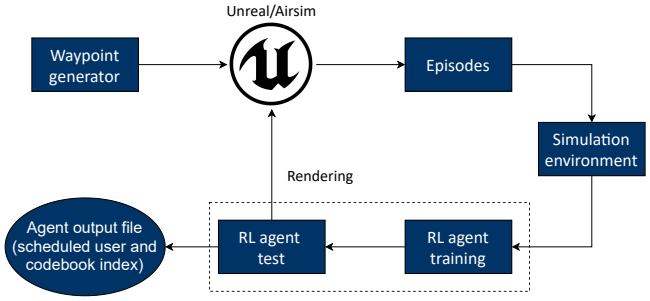


Fig. 2 – CAVIAR data generation

3.1 Trajectories generation in the 3D virtual world

In the CAVIAR methodology, trajectories of all moving objects in a simulation are saved as Comma-Separated Values (CSV) in files. To generate the trajectories, a 3D scenario is executed in Unreal and a waypoint file, which is a text file with reference points, must be executed by the Airsim software. Each execution is called an *Episode*, which lasts about three minutes with a sampling interval of ten milliseconds. In each sampling interval, a line is saved in the CSV file, and it is referred to in this paper as a step, to be consistent with the concept of discrete time-step adopted in the RL jargon. Each column of the file contains information related to the positions and orientations of pedestrians and cars. In addition, the file contains information of acceleration, linear, and angular velocities for the UAVs. CAVIAR currently supports 37 entities (34 pedestrians, 2 cars, and 1 UAV) in the environment.

3.2 MIMO propagation

The generation of the MIMO propagation considers an analog architecture of a UPA with N_t antenna elements at the BS, and UEs with UPA with N_r antennas. Therefore, the MIMO channel between the BS and a given UE is represented by a $N_r \times N_t$ matrix $\mathbf{H}$. We assume downlink transmission, carrier frequency $f_c = 60$ GHz, and a bandwidth of 100 MHz. Then, $\mathbf{H}$ is generated as a narrowband [20] Line-of-Sight (LoS) channel model. For simplicity, the UEs have a single antenna ($N_r = 1$) while the BS has an 8×8 UPA ($N_t = 64$). The *geometric channel model* [20] is adopted with $L = 2$ multipath components:

$$\mathbf{H} = \sqrt{N_t N_r} \sum_{\ell=1}^{L} \alpha_\ell \mathbf{a}_r(\phi_\ell^A, \theta_\ell^A) \mathbf{a}_t^*(\phi_\ell^D, \theta_\ell^D). \qquad (3)$$

The parameters in *(3)* are obtained as follows. The phase of the complex-gain α_ℓ is obtained from a uniform distribution with support $[0, 2\pi]$. To generate the magnitude

[2]https://github.com/lasseufpa/ITU-Challenge-ML5G-PHY-RL

Table 1 – Content of an episode file generated with CAVIAR

Content	Description
timestamp	When the dataset was gathered
obj	Name of the mobile entity
pos_x	Position in meters (north positive)
pos_y	Position in meters (east positive)
pos_z	Position in meters (down positive)
orien_x	Quaternion in degrees
orien_y	Rotation in degrees
orien_z	Rotation in degrees
orien_w	Rotation in degrees
linear_vel_x	Linear velocity in m/s
linear_vel_y	Linear velocity in m/s
linear_vel_z	Linear velocity in m/s
angular_vel_x	Angular velocity in m/s
angular_vel_y	Angular velocity in m/s
angular_vel_z	Angular velocity in m/s
linear_acc_x	Linear acceleration in m/s^2
linear_acc_y	Linear acceleration in m/s^2
linear_acc_z	Linear acceleration in m/s^2
angular_acc_x	Angular acceleration in m/s^2
angular_acc_y	Angular acceleration in m/s^2
angular_acc_z	Angular acceleration in m/s^2

$|\alpha_\ell|$, first the distance d between the BS and the UE is used to calculate the received power via the Friis equation. The path loss is obtained from this equation and determines $|\alpha_\ell|$, which decreases with d. The elevation ϕ_ℓ and azimuth θ_ℓ angles for departure and arrival are obtained from the orientation provided by the LoS path. The nominal LoS angles are slightly changed by adding to them Gaussian random variables with zero-mean and variance of 1 degree. These angles are used to compose the steering vectors $\mathbf{a}_t$ and $\mathbf{a}_r$.

With *(3)*, all the combinations of $\mathbf{w}_p$ and $\mathbf{f}_q$ could be evaluated according to *(1)*, and the indexes p and q that result in the received signal with the highest magnitude can be selected as the best indexes. Therefore, p and q, or equivalently the combined index i, can be used in supervised learning approaches as a label to train a model. In contrast, an RL approach may not need labels, as the RL agent could choose an index i based on its learning process, which results in a reward from the environment.

In both cases of RL and supervised learning, the chosen index i results in an equivalent channel, referred to as beam_index and channel_mag in Table 2, respectively, for a given user k. With the equivalent channel, or with the magnitude of the received signal $|y_k|$, the spectral efficiency is

$$S_{k,t,i} = \log_2(1 + \text{SNR}), \qquad (4)$$

for a given Signal-to-Noise Ratio (SNR) value. The maximum amount of bits that can be transmitted is $T_{k,t,i} = S_{k,t,i}\text{BW}$, where BW indicates the system bandwidth. $T_{k,t,i}$ is presented in Table 2 as bit_rate_gbps. Note that, the index choice and channel also depends on ue_index, which is discussed in Section 3.3.

Table 2 – Traffic and propagation data generated for a given step

Content	Description
chosen_ue	Name of the chosen UE
ue_index	Identification of chosen UE
beam_index	Selected beam
pkts_dropped	Total of dropped packets
pkts_transmitted	Total of transmitted packets
pkts_buffered	Total of buffered packets
bit_rate_gbps	Data rate in Gbps achieved
channel_mag	Combined channel magnitude
reward	Reward obtained during

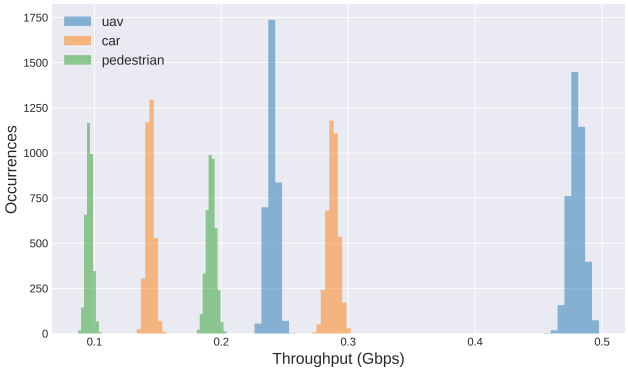


Fig. 3 – Histogram of packets traffic received by the BS for each user

3.3 Data traffic generation

To generate data traffic, we assume the BS has an individual buffer to store packets that should be forwarded for each UE (downlink). The packets have a fixed size of 8188 bytes. The users' data traffic is defined as Poisson processes with a time-varying mean $\lambda_k[t]$ for user k. Each user has a specific traffic magnitude defined as a fraction of the total throughput according to Table 3, distinguishing different applications. Fig. 3 shows the histogram of traffic throughput for each user in Gbps.

We specified two different network load scenarios for each type of UE, namely, light and heavy network traffic, to simulate the traffic variations along with the scenes. The total throughput of the heavy scenario is greater than the lighter one, and the simulation alternates between them after 1000 steps. This alternating behavior is shown in Fig. 3 as two clusters for each type of UE, which represent the light and heavy situations. Regardless of the traffic intensity, the incoming traffic for each user is buffered when there is buffer space available, otherwise the excess of packets are *tail-dropped*. The packets are also dropped when they occupy the buffer for more than 10 time steps.

The BS must choose a user for which a packet is sent in the current step, which is shown in Table 2 as ue_index. Moreover, the BS must also choose a beam that points to that user, as discussed in Section 3.2. For the chosen user and beam index, the following procedure is performed to emulate a transmission in the channel. The combined chan-

Table 3 – Network load information for light and heavy scenarios

Network load	Total throughput	UAV (%)	Pedestrian (%)	Car (%)
Light	0.48 Gbps	50%	20%	30%
Heavy	0.96 Gbps	50%	20%	30%

nel's bit rate $T_{k,t,i}$ is calculated, if the bit rate is greater than the number of bits of the packet that should be transmitted to the chosen UE, the packet is assumed as transmitted and removed from the buffer. On the other hand, if the packet's size is greater than the channel capacity, the packet is assumed as dropped during the transmission and it is kept in the buffer. In case of packets dropped, no packet is transmitted in that step.

4. RL-BASED USERS SCHEDULING AND BEAM SELECTION

As discussed in previous sections, this paper presents RL-based approaches to solve efficiently the combined problems of users scheduling and beam selection. The scenario consists of a BS that runs an RL agent, which receives a reward based on the service provided to the users. The solutions presented in this section were proposed for the ITU Challenge PS-006 of 2021. The organizers of the PS-006 provided a dataset created with the CAVIAR methodology, which represents an environment composed of a UAV, vehicles, and pedestrians, with rich interactions with the environment. The generation is detailed in Section 3 and results in distinct types of data that can be used by the agent to generate its choice. The action space is composed of two integers: a numeric identifier for the user allocated at the specific step, that can range between $[0, K-1]$; and the codebook index of the beam to be used to serve the user, which is an integer in the range $[0, M-1]$.

The RL agent can use several features as inputs such as data from GNSS, UE's orientation in the three rotation coordinates (front and side roll angles - pitch and roll), and its rotation over its axis (yaw). It is also possible to use information related to the transmission of packets, such as dropped, transmitted, and buffered packets. The last two other available input features for the agent are the bit rate and the channel magnitude at each step of the simulation, which are calculated as discussed in Section 3.2 considering a maximum SNR of 26 dB, which can be decreased according to the equivalent channel's magnitude.

A total of 700 episodes were created, from which 500 were used for training the proposed RL agents, and 200 for testing. To keep the challenge simpler, it was assumed that the RL agent, only serves three users, a car, a pedestrian, and the UAV. Thus, we used a subset of all the available UEs, given that a complete episode file contains information related to all moving objects in a scene (36 in total, consisting of all pedestrians, cars, and the UAV).

4.1 Baseline agents and reward

In the PS-006, three baseline approaches were provided, which are identified with the prefix "B-", namely, **B-Dummy**, **B-BeamOracle** and **B-A2C**. B-Dummy and B-BeamOracle are not based on RL. The B-Dummy agent assumes random action choices for both the scheduled user and for the beam index. Because of its random choices, B-Dummy is provided as a baseline for worst performance. The B-BeamOracle agent follows a sequential user scheduling pattern (e.g., 0-1-2-0-1-2 ...), but it always chooses the optimal beam index $\hat{i}$ for the selected user. Thus, it is expected B-BeamOracle presents the best performance. The third agent is based on the RL approach, which is based on the Advantage Actor Critic (A2C) [21] and is referred to in this paper as B-A2C. To implement the B-A2C agent, we used Stable Baselines[3] version 2.10.0, which was trained with default parameters.

A return function G, which is described in the next paragraphs, was used to evaluate the solutions. The participants were allowed to adopt other returns functions in the training, but the evaluations to generate the final ranking of the challenge for all the agents, including the provided baselines, were made according to G for the test episodes. The return G_e for episode e is

$$G_e = \sum_{t=1}^{N_s^e} r_e[t], \tag{5}$$

where N_s^e is the number of scenes in episode e. The corresponding reward $r_e[t]$ at discrete-time t is a weighted sum of transmitted and discarded packets given by

$$r_e[t] = \frac{P_{\text{tx}}[t] - 2P_d[t]}{P_b[t]}, \tag{6}$$

where $P_{\text{tx}}[t]$, $P_d[t]$, and $P_b[t]$ correspond, respectively, to the total amount (summation for all users) of transmitted, dropped and buffered packets at time t. The reward $r_e[t]$ is restricted to the range $-2 \leq r_e[t] \leq 1$. At each time t, a single user can be served, but $P_b[t]$ accounts for the number of packets in all three buffers. Hence, $r_e[t] = 1$ only if all buffered packages of the scheduled user are transmitted, while the buffers of the other two users were empty. Finally, the accumulated return is

$$G = \sum_{e=1}^{N_e} G_e, \tag{7}$$

where N_e is the number of episodes in the test set, which is disjoint from the training set.

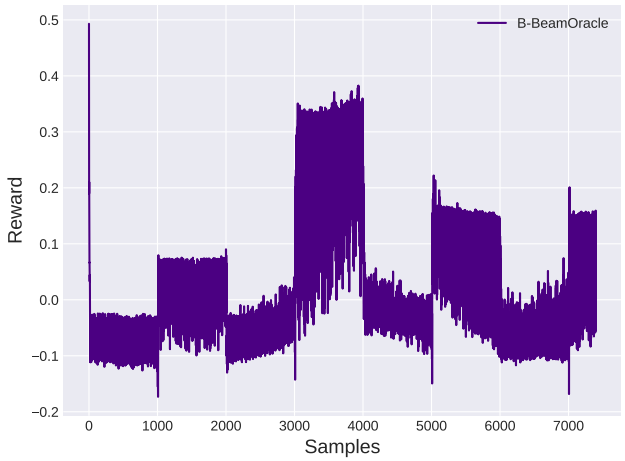


Fig. 4 – Reward obtained by the B-BeamOracle agent for a given episode. The traffic load switches every 1000 time steps between "heavy" and "light".

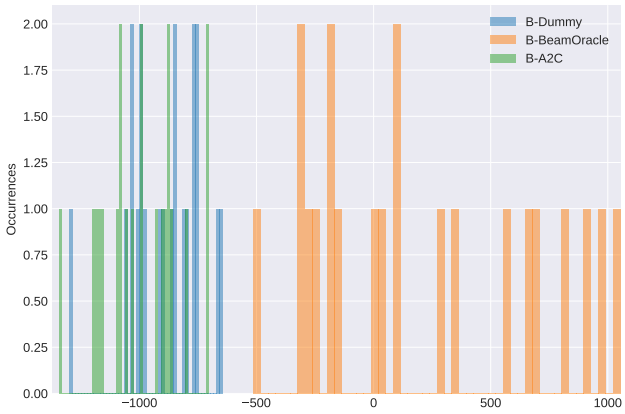


Fig. 5 – Histogram of the total sum of rewards achieved in episodes 449 - 469

Fig. 4 shows G for the switching behavior at every 1000 samples, between the "heavy" and the "light" data traffic that is described in Section 3.3. In Fig. 4, the B-BeamOracle was used, and it can be realized that in both situations of light and heavy traffic, the B-BeamOracle receives positive rewards in most of the steps. It can be concluded that B-BeamOracle may be sufficient to attend to the demand even without proper scheduling. However, B-BeamOracle is not realistic, as, during the operation of the mobile network, the BS does not know the best index $\hat{i}$. In practice, it would require a full beam-sweeping, which can be time-consuming and, in some cases, unfeasible due to the high delay it would incur.

To compare the B-Dummy, B-BeamOracle and the B-A2C agents, Fig. 5 shows the histogram of their rewards for along 20 test episodes. As expected, the B-BeamOracle presents the best performance and most of the rewards it received are positive. While B-Dummy and B-A2C's performances are similar. One reason for the bad performance of B-A2C is the choice of its input parameters. It may indicate the features provided to trained B-A2C cannot help the agent to learn patterns that lead to good user and beam index choices. Better modeling of the agent can substantially improve its performance. However, B-A2C was provided as an example to the participants, which could use it as a starting point to modify its parameters and substitute by other agents.

4.2 Team MLAB-RL

To improve the performance of the reinforcement learning, the agent, state, and reward were customized in this solution. The baseline B-A2C agent was substituted by the Proximal Policy Optimization (PPO) [22], which combines ideas from A2C and Trust Region Policy Optimization (TRPO) [23]. Specifically, the PPO2 algorithm from the Stable-Baselines package was utilized, as it allows execution in a Graphics Processing Unit (GPU). The batch size was set to 32 as it was verified empirically that its performance is slightly better than other cases.

The state is changed into a vector, containing the information of pos_x, pos_y, pos_z, bit_rate_gpbs, ue_index, and the recent history of UE selection. A 3-dimensional vector is used to reflect the selected UE. In detail, the information of the selected UE is obtained via chosen_ue, and is modified to [1, 0, 0], [0, 1, 0], and [0, 0, 1] for the UAV, car, and pedestrian, respectively. Another 3-dimensional vector is added taking the recent selections of UE into consideration. The history of the last 10 selections of UE is considered because it is highly related to the amount of dropped packets as the size of buffer storage is 10. This vector is calculated by adding up the selections of each UE type for each dimension. For example, when the last 10 history selections are [uav, uav, car, ped, uav, uav, car, car, ped, uav], the state of the recent history is expressed as [uav, car, ped] = [5, 3, 2]. Lastly, all 10 dimensions are scaled, where the pos_x, pos_y, pos_z are scaled to the range [-1, 1], and the others are scaled to the range [0, 1].

The reward was customized for the training phase r_{train} as follows:

$$r_{\text{train}} = r_{\text{bonus}} + r_{\text{weighted}}. \quad (8)$$

A bonus r_{bonus} is added because the original reward provided in the challenge, referred to in this section as r_{ori}, tends to keep a low value, and it deteriorates the performance of the learning. The bonus is given when r_{ori} is larger than -0.1 during the training phase. The bonus r_{bonus} is defined as follows:

$$r_{\text{bonus}} = 10 \times r_{\text{ori}} + 1, \quad (9)$$

where 1 is added to ensure the r_{bonus} is non-negative.

In addition, a weighted reward $r_{weighted}$ is used to assess the dispersion of the selections. It is encouraged that the BS avoids choosing the same UE. Because when the selections of UE are dispersed in the recent history, the amount

[3] https://github.com/hill-a/stable-baselines

of dropped packets can be reduced. Note that it may depend on the arrival rate of the packets for each user, and for users with traffic with different distributions another solution, such as weighted consideration of the history, could be more adequate. The weighted w is defined as follows:

$$w = \frac{10}{n_{\text{chosen\_ue}}}, \qquad (10)$$

where $n_{\text{chosen\_ue}}$ is the number of 10-recent-history selections of the current selected UE. For example, when the last 10 history selections are [uav, uav, car, ped, uav, uav, car, car, ped, uav], the w of selecting "uav" is $10/5 = 2$.

The weighted reward r_{weighted} is defined as follows:

$$r_{\text{weighted}} = w \times (r_{\text{ori}} + 2). \qquad (11)$$

As the range of r_{ori} is [-2, 1], it is added with 2 to ensure the r_{weighted} is non-negative.

4.3 Team IITI-RL

This solution explored various representations of input states and existing RL models, such as Deep Q-Learning (DQN) [24], policy gradient network [25] and A2C. These models were evaluated with inputs as position (x, y, z), orientation (x, y, z, w), packets dropped, packets transmitted, packets buffered, packets, bit rate and channel magnitude. These models use an extrinsic reward, which is the reward given by the environment. The main problem with this is that this function is hardcoded, which is not scalable for our problem.

Therefore, we proposed as a solution for the challenge the idea of curiosity-driven learning [26], which is based on embedding a reward function that is intrinsic to the agent, i.e., generated by the agent itself. In curiosity-driven learning, a policy π is trained to optimize the sum of the extrinsic reward (r^e in *(6)*) provided by the environment and the curiosity-based intrinsic reward signal (r^i) generated by the network. The intrinsic reward signal r^i is defined as follows:

$$r^i = \frac{\eta}{2}||\hat{\phi}(s_{t+1}) - \phi(s_{t+1})||_2^2, \qquad (12)$$

where $\eta > 0$ is a scaling factor, s_t represents the state at timestamp t, $\hat{\phi}(s_{t+1})$ is predicted the feature vector of the next state and $\phi(s_{t+1})$ is the real feature vector of the next state. We used a learning rate of 10^{-4} and batch size of 32. The results in Section 5 demonstrate the benefit of using the intrinsic reward in the given scheduling and resource allocation problem.

5. PERFORMANCE EVALUATION

In this section, we present performance evaluations of the techniques described in Section 4. The main metric adopted is the cumulative reward received. We also show how the actions taken by the agents affect the network as the total number of transmitted packets. Increasing the number of transmitted packets can be considered the main goal of the learning process because a higher number of packets transmitted can lead to better usage of the radio and network resources available. Moreover, increasing the number of transmitted packets could also reflect better service in general to the users.

Fig. 6 and Fig. 7 show the reward received by the agents in the evaluation of each team. As one of the rewards is presented on a different scale, we present distinct figures to preserve the data provided by the participants and to allow better individual analysis. Fig. 6 shows the PPO agent of the MLAB Team presents an increase in the cumulative reward compared to the dummy agent. The best and worst average rewards are -0.0330 (in ep. 115) and -0.1602 (in ep. 30), respectively. Also, it was noted by observing the output files that the BS tends to select the same beam nearly all the time. Fig. 6 also shows the baseline agents as dashed lines, as this figure uses the same reward.

Fig. 7 shows the reward received by the four RL algorithms evaluated by the IITI-RL team. Models with extrinsic rewards had a poorer performance as compared to curiosity-driven learning. DQN has the least return G since it is the least robust model and has problems supporting continuous values and a high number of input state variables in the observation space. The A2C and the policy gradient network achieved almost the same reward, which is a slight improvement over DQN. The curiosity-driven solution performs the best out of the four models, and it shows the effectiveness of intrinsic rewards among the evaluated algorithms.

An evaluation considering *(6)* as the metric is shown in Fig. 8. According to Fig. 8, the solution presented by the MLAB-RL team increased the performance in relation to the B-A2C. While the solution presented by the IITI-RL had a slightly decreased performance. Fig. 9 shows the histogram of the number of transmitted packets, which is another perspective of the results presented in Fig. 8. It can be assumed that the goal of the BS is to increase the global data rate with the actions of the agent. It is equivalent to moving the histogram of occurrences in Fig. 9 to the right, and it would reflect in more packets transmitted, and consequently an increase in the global data rate. The data in Fig. 9 is consistent with what is shown in Fig. 8; in other words, the actions taken by the agent provided by the MLAB-RL team performed better in increasing the number of transmitted packets in relation to the B-A2C agent.

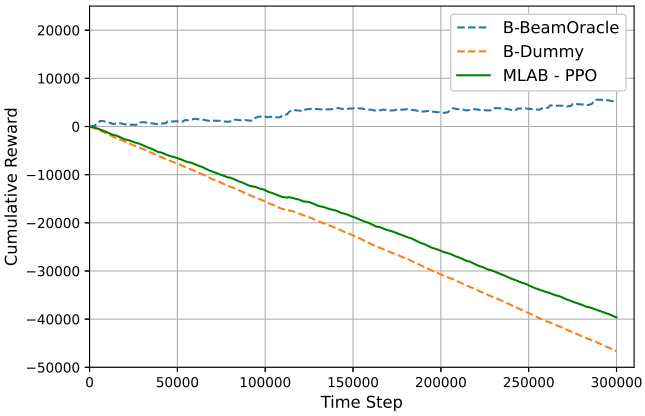


Fig. 6 – Cumulative reward for the PPO agent of the MLAB Team and the baseline agents

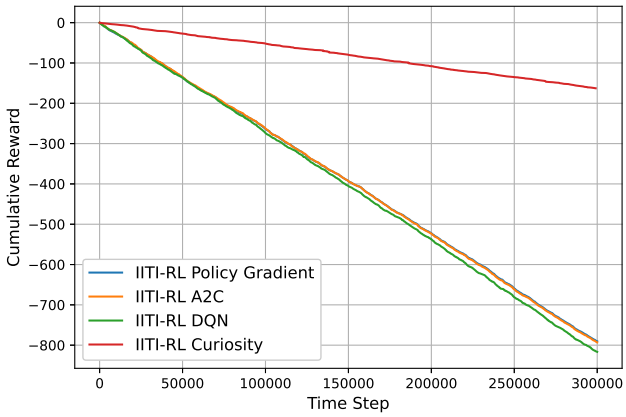


Fig. 7 – Customized cumulative reward for the agents evaluated by the IITI-RL Team

6. CONCLUSION

This paper presented solutions based on reinforcement learning approaches for the combined problems of beam selection and user scheduling. We consider the RL agent is implemented at the BS, and that it must choose the user to be served and the code words for the transmission by the BS and reception by the UE (downlink). The literature shows that the problem of joint scheduling and beam selection is challenging, and it demands sophisticated modeling in order to obtain outstanding results. As a standalone problem, the literature about beam selection relies on the use of strategies such as top-k classification, where the objective is choosing a subset of beams that has a high probability of containing the optimal choice instead of choosing the best beam to serve a given user, which relaxes the difficulty of the problem. Therefore, the combination of the beam selection with scheduling task is even harder, but an efficient scheduling, in relation, for example, to the service level agreement, with a fast beam selection, by skipping the time-consuming beam sweeping, can make the network operation more efficient.

We used a dataset generated with a scenario created with the CAVIAR methodology, which mimics a complex scenario within a 3D virtual environment. The CAVIAR methodology allows generating several types of input

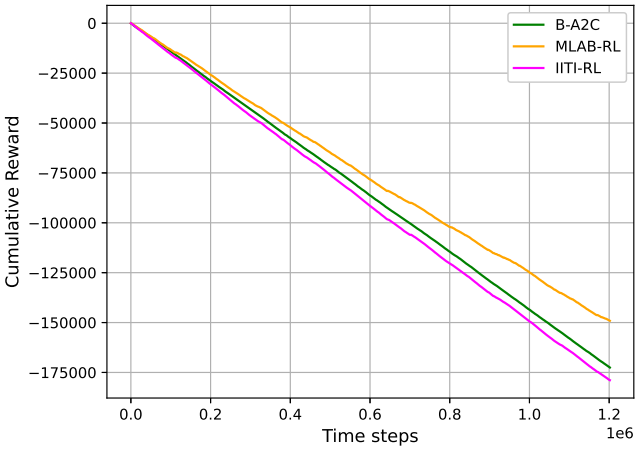


Fig. 8 – Cumulative reward for the PPO and Curiosity Driven agents of the MLAB-RL and IITI-RL teams, respectively, with action evaluated according to baseline metric provided

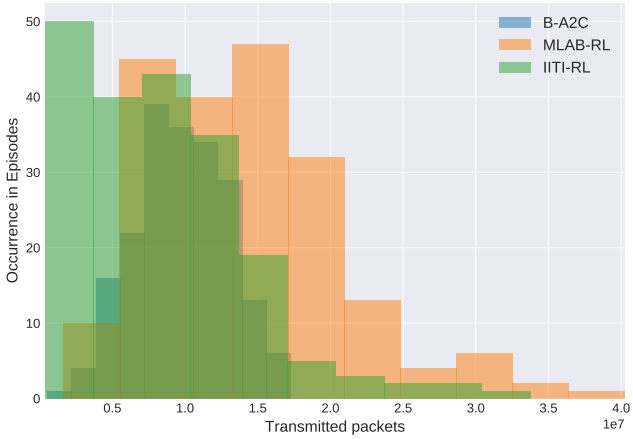


Fig. 9 – Histogram of the number of transmitted packets per episode with the actions taken by the agents

data, such as positions, orientations, channel realizations, network traffic, etc. All the data could be used in the training and evaluation processes of RL agents. There are a total of 37 mobile UEs in the 3D environment (34 pedestrians, 2 cars and 1 UAV). Due to the difficulty of proposing solutions to handle all the UEs in the scenario, in this paper, only a UAV, a pedestrian and a vehicle were served by the BS. The other UEs were kept in the environment to maintain its richness in terms of mobility and interactions, as they could serve as moving scatterers or blockers to the electromagnetic signal propagation.

We presented evaluations of several RL approaches to handle the beam selection and user scheduling task at the BS to serve three UEs. The RL approaches were evaluated according to the cumulative reward received. We also presented an evaluation about how the actions taken by the trained agents influence the overall downlink data rate, as increasing it can be correlated to better service to the users in general. In the evaluations we show that both the cumulative reward and the data rate can be increased in relation to the performance of the baseline approach.

Despite the improvement to the RL agents presented in relation to the baseline, there is still room for improvement for RL approaches in the context of 5G and future networks. RL still has downsides that need to be addressed in order for it to achieve the same success as supervised learning approaches on a wide variety of problems, such as the impact that the framing of the problem and the reward engineering have on the agent's capacity to converge.

The results helped to identify some open research topics in this area. For example, the three UEs served by the BS in this paper represent a relatively simple situation. Thus, further evaluation on how to handle all the UEs, or a greater number of UEs, as well as considering both uplink and downlink. Also, in practice, the BS does not have a fixed number of users, thus future research in this area should consider how to deal with a variable number of users. The variable number of users is related to the handover operations, which is based on coordination between distinct BSs to switch moving UEs between their cells. All these further evaluations result in a more difficult scenario, thus proper future research could focus on creating approaches more appropriate to mobile networks, which can include (not limited to) parameterization and selection of the RL algorithms, input data scaling and organization and reward design.

ACKNOWLEDGEMENTS

The authors would like to thank Vishnu Ram OV, Thomas Basikolo and all the team for the dedication in organizing the ITU AI/ML Challenge, and for the invitation to propose the problem statement detailed in this paper. The authors also thank all the judges and participants that proposed solutions. The work of Ilan Correa, Ailton Oliveira, Cleverson Nahum, Felipe Bastos, João Borges, Pedro Batista and Aldebaro Klautau was supported by the São Paulo Research Foundation (FAPESP) with grant #20/05127-2, Innovation Center, Ericsson Telecomunicações S.A. and CNPq, Brazil.

REFERENCES

[1] W. Roh, J. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar. "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: theoretical feasibility and prototype results". In: *IEEE Communications Magazine* 52.2 (Feb. 2014), pp. 106–113. ISSN: 1558-1896. DOI: 10.1109/MCOM.2014.6736750.

[2] E. Björnson, L. Van der Perre, S. Buzzi, and E. G. Larsson. "Massive MIMO in Sub-6 GHz and mmWave: Physical, Practical, and Use-Case Differences". In: *IEEE Wireless Commun.* 26.2 (2019), pp. 100–108.

[3] E. G. Larsson, O. Edfors, F. Tufvesson, and T. L. Marzetta. "Massive MIMO for next generation wireless systems". In: *IEEE Commun. Mag.* 52.2 (2014), pp. 186–195.

[4] Abdulmalik Alwarafy, Mohamed Abdallah, Bekir Sait Ciftler, Ala Al-Fuqaha, and Mounir Hamdi. "Deep Reinforcement Learning for Radio Resource Allocation and Management in Next Generation Heterogeneous Wireless Networks: A Survey". In: (2021). arXiv: 2106.00574 [eess.SP].

[5] Aldebaro Klautau, Ailton de Oliveira, Isabela Pamplona Trindade, and Wesin Alves. "Generating MIMO Channels For 6G Virtual Worlds Using Ray-tracing Simulations". In: *arXiv preprint arXiv:2106.05377* (2021).

[6] Shuaifeng Jiang and Ahmed Alkhateeb. *Computer Vision Aided Beam Tracking in A Real-World Millimeter Wave Deployment*. 2021. arXiv: 2111.14803 [eess.SP].

[7] Guillem Reus-Muns, Batool Salehi, Debashri Roy, Tong Jian, Zifeng Wang, Jennifer Dy, Stratis Ioannidis, and Kaushik Chowdhury. "Deep Learning on Visual and Location Data for V2I mmWave Beamforming". In: *17th International Conference on Mobility, Sensing and Networking (MSN)*. 2021, pp. 559–566. DOI: 10.1109/MSN53354.2021.00087.

[8] Jonggyu Jang, Jung Hwa Park, and Hyun Jong Yang. "Supervised-Learning-Based Resource Allocation in Wireless Networks". In: *2020 International Conference on Information and Communication Technology Convergence (ICTC)*. 2020, pp. 1022–1024. DOI: 10.1109/ICTC49870.2020.9289481.

[9] Chunmei Xu, Shengheng Liu, Cheng Zhang, Yongming Huang, and Luxi Yang. "Joint User Scheduling and Beam Selection in mmWave Networks Based on Multi-Agent Reinforcement Learning". In: *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*. 2020, pp. 1–5. DOI: 10.1109/SAM48682.2020.9104386.

[10] Faris B. Mismar, Brian L. Evans, and Ahmed Alkhateeb. "Deep Reinforcement Learning for 5G Networks: Joint Beamforming, Power Control, and Interference Coordination". In: *IEEE Transactions on Communications* 68.3 (Mar. 2020), pp. 1581–1592. ISSN: 1558-0857. DOI: 10.1109/tcomm.2019.2961332. URL: http://dx.doi.org/10.1109/TCOMM.2019.2961332.

[11] Esteban Egea-Lopez, Fernando Losilla, Juan Pascual-Garcia, and Jose Maria Molina-Garcia-Pardo. "Vehicular networks simulation with realistic physics". In: *IEEE Access* 7 (2019), pp. 44021–44036.

[12] Aldebaro Klautau, Pedro Batista, Nuria González-Prelcic, Yuyang Wang, and Robert W Heath. "5G MIMO data for machine learning: Application to beam-selection using deep learning". In: *2018 Information Theory and Applications Workshop (ITA)*. IEEE. 2018, pp. 1–9.

[13] Aldebaro Klautau, Nuria González-Prelcic, and Robert W Heath. "LIDAR data for deep learning-based mmWave beam-selection". In: *IEEE Wireless Communications Letters* 8.3 (2019), pp. 909–912.

[14] Wei Jiang, Bin Han, Mohammad Asif Habibi, and Hans Dieter Schotten. "The road towards 6G: A comprehensive survey". In: *IEEE Open Journal of the Communications Society* 2 (2021), pp. 334–366.

[15] Ailton Oliveira, Felipe Bastos, Isabela Trindade, Walter Frazão, Arthur Nascimento, Diego Gomes, Francisco Müller, and Aldebaro Klautau. "Simulation of Machine Learning-Based 6G Systems in Virtual Worlds". In: *Submitted to ITU Journal on Future and Evolving Technologies* (2021).

[16] A. Klautau, P. Batista, N. González-Prelcic, Y. Wang, and R. W. Heath. "5G MIMO Data for Machine Learning: Application to Beam-Selection Using Deep Learning". In: *2018 Information Theory and Applications Workshop (ITA)*. 2018, pp. 1–9.

[17] Y. Wang, A. Klautau, M. Ribero, A. C. K. Soong, and R. W. Heath. "MmWave Vehicular Beam Selection With Situational Awareness Using Machine Learning". In: *IEEE Access* 7 (2019), pp. 87479–87493.

[18] Y. Heng and J. G. Andrews. "Machine Learning-Assisted Beam Alignment for mmWave Systems". In: *2019 IEEE Global Communications Conference (GLOBECOM)*. 2019, pp. 1–6.

[19] Q. C. Li, H. Niu, G. Wu, and R. Q. Hu. "Anchor-booster based heterogeneous networks with mmWave capable booster cells". In: *2013 IEEE Globecom Workshops (GC Wkshps)*. 2013, pp. 93–98.

[20] R. W. Heath, N. González-Prelcic, S. Rangan, W. Roh, and A. M. Sayeed. "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems". In: 10.3 (Apr. 2016), pp. 436–453. ISSN: 1932-4553. DOI: 10.1109/JSTSP.2016.2523924.

[21] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Timothy P. Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. "Asynchronous Methods for Deep Reinforcement Learning". In: (2016). DOI: 10.48550/ARXIV.1602.01783. URL: https://arxiv.org/abs/1602.01783.

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. *Proximal Policy Optimization Algorithms*. 2017. DOI: 10.48550/ARXIV.1707.06347. URL: https://arxiv.org/abs/1707.06347.

[23] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. "Trust Region Policy Optimization". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.html.

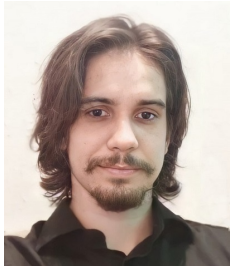
[24] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. *Playing Atari with Deep Reinforcement Learning*. 2013. DOI: 10.48550/ARXIV.1312.5602. URL: https://arxiv.org/abs/1312.5602.

[25] Ronald J. Williams. *Simple statistical gradient-following algorithms for connectionist reinforcement learning*. 1992. DOI: 10.1007/BF00992696.

[26] Yadan Luo, Zi Huang, Zheng Zhang, Ziwei Wang, Jingjing Li, and Yang Yang. *Curiosity-driven Reinforcement Learning for Diverse Visual Paragraph Generation*. 2019. DOI: 10.48550/ARXIV.1908.00169. URL: https://arxiv.org/abs/1908.00169.

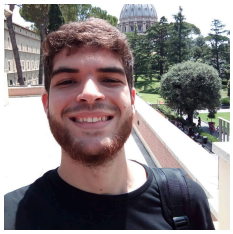
AUTHORS

Ilan S. Correa received a Bachelor's degree (UFPA, Brazil, 2012) in computer engineering, M.Sc. (UFPA, Brazil, 2014) and Ph.D. degrees (Universidade Federal do Pará, UFPA, Brazil, 2020) in electrical engineering. Currently, he is a professor at UFPA and is an associate professor at the 5G and IoT Research Group at LASSE/UFPA. He works in research and development projects related to 5G communications, embedded systems and electronics.



Ailton Oliveira is a B.Sc candidate in electrical engineering at Universidade Federal Pará, Brazil. He is currently a research student at the Telecommunications, Automation and Electronics R&D Center (LASSE/UFPA). His achievements were recognized with the outstanding undergraduate researcher award from LASSE/UFPA, and had an article awarded in 2020, by the Brazilian Telecommunications Society (SBrT), with research focused on machine learning applied to beam-selection. His current research interests include digital communications, 5G/B5G networks, MIMO systems, data science and machine learning.

Bojian Du received a B.E. degree in electronic information engineering from Beijing University of Technology, Beijing, China, in 2017. He received his Master's and is pursuing his doctorate degree in electrical engineering at The University of Tokyo, Tokyo, Japan, in 2020. His current research interest is primarily in time series data analysis.



Cleverson V. Nahum received a B.Sc. degree in computer engineering from the Federal University of Pará (UFPA), Belém, Pará, Brazil, in 2019. He received his Master's and is pursuing his doctorate degree in electrical engineering with emphasis on telecommunications in the Electrical Engineering Graduate Program at UFPA, in 2021. He is part of the Research and Development Center for Telecommunications, Automation and Electronics (LASSE) since 2016. His current research interests include network slicing, radio resource management, and artificial intelligence applied on mobile communication systems.

Daisuke Kobuchi received B.E. and M.E degrees in electrical engineering from the University of Tokyo, Tokyo, Japan, in 2019 and 2021, respectively. He is currently pursuing a Ph.D degree at the same university. His current research interests include wireless power transfer and wireless communications. He is a student member of the IEEE and IEICE.

Felipe Bastos received a technical degree in telecommunications from Instituto Federal do Pará in 2015, and a B.Sc. degree in computer engineering from Universidade Federal do Pará (UFPA) in 2020. He also participated in an inter-university exchange at École Supérieure d'Informatique, Électronique, Automatique (ESIEA) through the BRAFITEC program. He was an intern at the European Organization for Nuclear Research (CERN) in 2020. Currently, he is with the Telecommunications, Automation and Electronics R&D Center (LASSE) and pursues a Master of Science degree in electrical engineering at Universidade Federal do Pará (UFPA). He is interested in embedded systems, Internet of Things, 5G networks, and artificial intelligence.

Hirofumi Ohzeki received a B.E. degree in electrical engineering from Kyoto University, Kyoto, Japan, in 2019. He is pursuing a Master's degree in electrical engineering and information systems at The University of Tokyo, Tokyo, Japan, in 2021. His current interests are primarily in radio resource management and machine learning.

João Borges received the B.Sc. degree in computer engineering (2019), and is currently pursuing a Master's degree in computational intelligence with the Federal University of Pará (UFPA), Brazil. He has been a member of the Research and Development Center for Telecommunications, Automation and Electronics (LASSE), since 2017. His current research interests includes artificial intelligence applied on mobile communication systems.

Mohit Mehta is pursuing a B.Tech in electrical engineering from the Indian Institute of Technology, Indore, 2018. His research interests include artificial intelligence, machine learning, 5G networks, and MIMO communications.

Pedro Batista received the B.S., M.S., and Ph.D. degrees from the Electrical Engineering Graduate Program, Federal University of Pará, Brazil. He is currently a researcher with Ericsson. His research interests are in optimization of future mobile networks, particularly, using machine learning and machine reasoning, and future Internet architectures.

Ryoma Kondo received B.E. and M.E. degrees in information engineering from Tokyo Denki University, Tokyo, Japan, in 2015 and 2017, respectively. He is currently pursuing a Ph.D. degree at The University of Tokyo. His research interests include web database systems and data science.

Sundesh Gupta is pursuing a B.Tech degree in computer science and engineering from the Indian Institute of Technology Indore, 2018. His research interests include machine learning and computer vision.

Vimal Bhatia received a Ph.D. degree from the Institute for Digital Communications with The University of Edinburgh, Edinburgh, U.K., in 2005. He is currently working as a professor with the Indian Institute of Technology (IIT) Indore, India, and is an adjunct faculty at IIT Delhi and IIIT Delhi, India. He has over 300 peer-reviewed publications and has filed 13 patents (with four granted). His research interests are in the broader areas of communications, non-Gaussian non-parametric signal processing, machine/deep learning with applications to communications, and photonics. He is currently an associate editor for IETE Technical Review, Frontiers in Communications and Networks, Frontiers in Signal Processing, and IEEE Wireless Communications Letters.

Aldebaro Klautau received a Ph.D. degree from the University of California at San Diego (UCSD) in 2003, and is currently a full professor at the Federal University of Para (UFPA), Brazil. At UFPA he is also the ITU Focal Point and directs the LASSE Research Group. He is a senior member of the IEEE and of the Brazilian Telecommunications Society (SBrT), and a researcher of CNPq and INESC Brazil.