Deep Reinforcement Learning for Joint Energy Saving and Traffic Handling in xG RAN

Khoa Dang, Hicham Khalifé, Mathias Sintorn, Dag Lindbo, Stefano Secci

Deep Reinforcement Learning for Joint Energy Saving and Traffic Handling in xG RAN

Khoa Dang*‡, Hicham Khalifé*, Mathias Sintorn†, Dag Lindbo†, Stefano Secci‡
*Ericsson R&D, Massy, France. {anh.khoa.dang, hicham.khalife}@ericsson.com
†Ericsson, Kista, Sweden. {mathias.sintorn, dag.lindbo}@ericsson.com
‡Cnam, Paris, France. {anh-khoa.dang, stefano.secci}@cnam.fr

***Abstract*—In this paper, we formulate the traffic-aware mobile nodes sleeping with traffic offloading as a Markov Decision Process (MDP) and solve it using Deep Reinforcement Learning (DRL). Our model characterizes jointly the energy saving actions due to base stations entering in sleep mode as well offloading options to neighboring nodes of the turned off gNodeB. To solve this problem, the Proximal Policy Optimization (PPO) integrated with action masking is leveraged. Our validation results, when training the model with open source datasets, show a potential of reducing up to 16% of the network energy consumption without negatively affecting traffic coverage.**

***Index Terms*—Deep Reinforcement Learning (DRL), Base Station Sleep Control, Traffic Handling, Energy Saving.**

I. Introduction

A recent report by the UK government [1] indicated that Information and Communication technologies (ICT) energy consumption (excluding televisions) constituted 4-6% of global electricity usage in 2020, with an anticipated increase over the next 5-10 years. To this end, the demand to address the energy consumption challenge in the ICT sector and the transition towards sustainable practices becomes essential. Mobile networks, serving as the primary consumers of ICT energy, are witnessing significant commitment from mobile network operators (MNOs) to achieve Net Zero targets (in alignment with limiting global warming to 1.5°C) by 2050 or earlier, as reported by GSMA [2]. This commitment signifies a collective effort to address the environmental impact and strive for a more sustainable future.

5G, the current deployed mobile technology standard, is expected to bring a revolutionary leap in performance, capabilities, and user experiences [3]. Nevertheless, to meet these ambitious requirements in practice, MNOs are increasing their deployment density, positioning access nodes closer to end users. This leads to geographical regions now covered by one coverage base station or BS (operating on lower bands) and possibly dozens of small capacity BSs or cells offering high throughput but limited geographic coverage [3]. As a consequence, BSs becomes the main power consumers, accounting for about 60% of the total power consumption in mobile networks [4].

Starting from the observation that traffic load experiences temporal and spatial variations (as illustrated in Fig. 1), many studies have focused on load-adaptive solutions that deactivate nodes during low or no traffic periods to reduce energy wastage [4]. More specifically, the problem is usually framed as determining the optimal binary states (ON/OFF) for each BS in the network while assuring QoS constraints. While analytical [6] or optimization-based [7] approaches can be derived to address this issue, they often entail high computational costs and demand repeated computations for every time slot, making them less viable as the network expands. Recently, DRL has emerged as a promising alternative method [8]–[12], adapting efficiently to dynamic network conditions and lowering computational complexities. Inspired by these DRL approaches, we propose extending the traditional BS sleep control problem beyond the conventional binary state (ON/OFF) paradigm. Particularly, we present an MDP formulation that incorporates the notion of load offloading to neighboring nodes as a part of the action space.

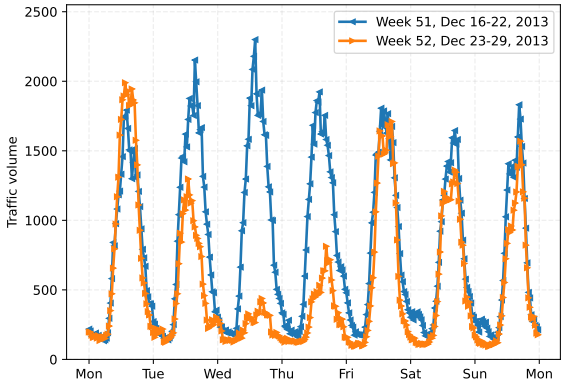


Fig. 1: The weekly mobile traffic fluctuations at *Duomo di Milano* [5]. Week 51 reflects standard traffic patterns, while week 52 contains the Christmas holiday.

By leveraging the latest breakthroughs in DRL, the main contributions can be summarized as follows:

- Differing from [8]–[12], where actions solely control the binary state of BSs, our approach introduces a novel MDP formulation for nodes sleep control, where actions directly migrate loads to neighboring nodes. We then solve this problem using PPO [13] integrated with logit-level action masking for efficient policy discovery.
- Moreover, an extensive performance evaluation is carried out using a realistic mobile dataset from Milan City. The results show that the obtained/proposed policy can achieve about 16% in energy savings gain without compromising traffic. Interestingly, the policy exhibits some

notable generalization properties, effectively accommodating the holiday scenario.

The rest of the paper is organized as follows. Section II defines the system model. In Section III, the BS activation problem is formulated, highlighting the direct influence of actions on neighboring BSs. An extensive performance evaluation is then presented in Section IV. Finally, Section V concludes the paper and outlines future directions.

II. System model

1) Network scenario: This work considers a large-scale network comprising a Macro Base Station (MBS) offering extensive coverage across a wide area and multiple Small Base Stations (SBSs) responsible for managing high-demand data capacity within specific regions. In this deployment scenario, we assume that SBSs can connect to MBS via optical fiber links, which enables the MBS to effectively monitor and take centralized control over all the SBSs [3]. The SBS can enter sleep mode, transferring its load to neighboring SBSs with handover [14]. Nearby SBSs can employ cell shaping techniques to ensure coverage for the migrated load [15].

We consider a geographically defined sub-network constituted of $N+1$ total nodes, composed of N SBSs associated with a single MBS. Let i denote the ith base station in this defined part of the network with $i \in \{1, 2, \ldots, N+1\}$. In the given scenario, each SBS has links and shares overlapping coverage with up to 4 neighboring SBSs (Fig. 2). To prevent energy waste during varying mobile traffic, BSs must adjust their operation modes by entering sleep mode when underutilized. Given migration and hardware costs, we choose less frequent BS activation by dividing time into fixed 30-minute slots. Let $\delta_t^{(i)}$ be the variable that denotes the activation state of the ith BS in time step t, with $\delta_t^{(i)} = 1$ indicating that node i is active and $\delta_t^{(i)} = 0$ otherwise.

The MBS must consistently remain operational (δ_t^{macro} is always 1) to provide constant coverage and host the DRL agent that coordinates the activation and traffic-shifting strategy of all managed SBSs. In our solution, we not only take into account the binary state (ON/OFF) of SBSs but also consider the impact of load shifting on neighboring SBSs and MBS. With this consideration, we gradually (de)activates SBS (one by one) to guarantee the stability of the network.

2) Network energy consumption: The power consumption of BS follows the Energy Aware Radio and neTwork tecHnologies (EARTH) profiling [16], with the power utilization $P^{(i)}$ for every base station i calculated as [10]:

$$P^{(i)} = \begin{cases} P_o^{(i)} + \eta^{(i)} \lambda_t^{(i)} P_T^{(i)} & \text{if } 0 < \lambda_t^{(i)} \leq 1 \\ P_s^{(i)} & \text{if } \lambda_t^{(i)} = 0 \end{cases} \tag{1}$$

where $P_o^{(i)}$ and $P_s^{(i)}$ are the fixed operational and sleeping power consumption, each, $\eta^{(i)}$ is the load-dependent power consumption slope, and $P_T^{(i)}$ is the transmission power. $\lambda_t^{(i)} \in [0, 1]$ is normalized traffic load of i at time step t, defined as:

$$\lambda_t^{(i)} = \frac{u_t^{(i)}}{C_i} \tag{2}$$

where the resources utilization from i at time t is denoted by $u_t^{(i)}$ and C_i is the total available resources of base station i ($u_t^{(i)} \leq C_i$).

3) Network load re-association: When an SBS enters sleep mode, it is necessary to transfer its load to *one or up to 4* neighboring SBSs or directly to the MBS in order to maintain uninterrupted service.

Considering a scenario when a node j deactivates and migrates its load to k, the factorized loads of k and j after traffic re-association becomes:

$$\lambda_t^{(k)\prime} = \lambda_t^{(k)} + \rho_t^{(j)} \phi_{j,k} \lambda_t^{(j)} \tag{3}$$

$$\lambda_t^{(j)} = 0 \tag{4}$$

where $\rho_t^{(j)}$ corresponds to the percentage of load share from j at time step t, and $\phi_{j,k}$ represents the capacity ratio of j traffic in the total node k load, such that:

$$\phi_{j,k} = \frac{C_j}{C_k} \tag{5}$$

Given that different BSs may have varying maximum capacities, this ratio characterizes precisely the shifted traffic from node j to k. In fact, if node k has in turn to enter sleep mode, and offload to station l, the capacity ratio still holds since:

$$\phi_{j,k} \times \phi_{k,l} = \frac{C_j}{C_k} \times \frac{C_k}{C_l} = \frac{C_j}{C_l} = \phi_{j,l} \tag{6}$$

In practice, this other node l must share overlapping coverage with the original node j. In our case, l can only be the Macro base station as we do not allow traffic to be moved to nodes away from the orginal small base station.

In contrast, when node j re-activates, the factorized loads of j and k following traffic re-association can be written as:

$$\lambda_t^{(j)} = \frac{u_t^{(j)}}{C_j} \tag{7}$$

$$\lambda_t^{(k)\prime} = \lambda_t^{(k)} - \rho_t^{(j)} \phi_{j,k} \lambda_t^{(j)} \tag{8}$$

4) Objectives: The primary goal of this study is to find an optimal policy to **gradually** act on small base stations in the network, aiming to minimize the total power consumption **in the long run**, while also preventing service interruption. Since the Macro base station is always ON to provide wide-range coverage, the aim is to offload the traffic of sleeping SBSs to neighboring ones and potentially to MBS while not exceeding the Macro base station maximum capacity. Therefore, the objective function is defined as:

$$\begin{aligned} \min \quad & P_{\text{total}} = \sum_{i=1}^{N+1} (P_o^{(i)} + \eta^{(i)} \lambda^{(i)} P_T^{(i)}) \delta^{(i)} + P_s^{(i)} (1 - \delta^{(i)}) \\ \text{s.t.} \quad & \lambda^{\text{macro}} \leq 1 \end{aligned} \tag{9}$$

III. Problem Formulation

In this section, we formulate the problem of maximizing rewards for sequential small base stations activation as a Markov Decision Process (MDP), which can be expressed as a tuple $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the transition probability function, $\mathcal{R}$ is the reward function and $\gamma \in [0, 1]$ is the discount factor. The design of our MDP can be described as follows.

1) State space: At every time step t, the environment records the system state as an array of normalized traffic loads of the entire network topology.

$$s_t = [\lambda_t^{(1)}, \lambda_t^{(2)}, \lambda_t^{(3)}, \ldots, \lambda_t^{(N+1)}] \tag{10}$$

2) Action space: As stated previously, our proposed model goes beyond making binary decisions (ON/OFF) for SBSs. It also involves shifting traffic load to nearby base stations. In our action space, there are 17 action types per SBS, categorized into 3 classes: *(1)* activate; *(2)* deactivate and shift load to neighbors; *(3)* deactivate and migrate all load to MBS.

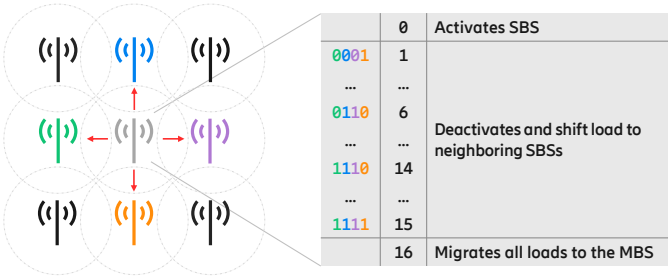


Fig. 2: Action types per SBS

Fig. 2 shows how these action types are encoded. Since each SBS have up to 4 neighbors in our model, 15 patterns of shifting load to neighbors are possible (each bit in the binary representation represents the SBSs to offload to). The load can be shifted equally in up to 4 neighbors (for example, if the action type is 15, then load is shared evenly between the 4 SBS neighbors with a $\rho_t = 25\%$). In addition to the 17 actions an SBS can take (activate encoded as 0, migrate to SBS neighbors encoded in one of the 15 possibilities of offloading and Migrate load to Macro node encoded as 16), we add a *do-nothing* action for the entire system, making the total action space of $17N+1$ possible actions. One can easily see that the size of the action space scales with N. At every time step t, the agent selects an action from : $a_t \in \{0, 1, 2, \ldots, 17N\}$.

It is important to note that when the load is shifted to a busy neighbor (i.e., $\lambda_t^{(i)} \approx 1$), the exceeded factorized load then relocates to the MBS. If the MBS also reaches its maximum capacity ($\lambda_t^{\text{macro}} = 1$), then the exceeded factorized load in the MBS is considered as *factorized traffic loss*. For practical considerations, since all loads are eventually shifted to the centralized coverage MBS, it is assumed that the MBS can calculate the overflow workload. We denote this factorized traffic loss at every time step t as l_t.

Action space masking: involves removing actions that are not feasible or allowed in a given state. This enables the agent to concentrate solely on relevant and valid actions within its current context, enhancing learning efficiency and preventing exploration of irrelevant or impossible actions.

Action masking is particularly relevant to our problem as realistic RAN configurations vary by geographical regions (e.g., some SBSs have only 2 neighbors, or for different N). This masking enables our proposed solution to generalize across diverse RAN scenarios, ensuring adaptability and applicability.

In addition, action masking enables the integration of specific constraints into the environment. Generally, the *action mask* in our model serves the following purposes:

- Preventing actuation on the same previous SBS (to avoid continuous on-off cycling on the same SBS).
- Enforcing only feasible actions in each state:
  - Prohibiting load shifting to a deactivated SBS.
  - Preventing re-deactivating a deactivated SBS.

With these considerations, the action mask in our model is computed based on s_t and a_{t-1} at each time t .

3) Reward function: The reward signal of the system is computed after taking action a_t and transit to time step $t+1$, comprised of energy consumption and traffic loss. Since the goal is to minimize excessive power consumption and penalize service disruption, our reward is defined as:

$$r_t = \mathcal{R}(s_t, a_t) = \sum_{i=1}^{b+1} (P_{\text{max}}^{(i)} - P_{t+1}^{(i)}) - \beta l_{t+1} \tag{11}$$

where $P_{\text{max}}^{(i)}$ is the maximum energy consumption of node i, $P_t^{(i)}$ is calculated at time t as in eq. (1), and β is the traffic loss penalty factor. The reward calculation is delayed by 1 time step to assess the impact of actions on the next state.

4) State transition and episodic configuration: At each time step t, the agent observes state $s_t \in \mathcal{S}$ and then decides an action $a_t \in \mathcal{A}$ to execute. Consequently, the agent sees a new state s_{t+1} and calculates reward signal r_t accordingly. Thus, the state transition probability function is represented as $\mathcal{P}(s_{t+1}, r_t | s_t, a_t)$. The system episode spans 7 days a week, starting on Monday, with 30-minute intervals, resulting in a total of 336 time (horizon) steps.

The next step is to train an agent to optimize policy π_θ that maximizes cumulative rewards. The aim is to enhance π_θ for maximum expected discounted returns in each episode:

$$J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{H-1} \gamma^t r_t \right] = \mathbb{E}_{\tau \sim \pi_\theta} [G_t] \tag{12}$$

where the trajectory τ is sampled from π_θ, with G_t representing the cumulative rewards over a time horizon (H).

IV. Performance Evaluation

A. Dataset and experimental setups

1) Dataset: To assess the feasibility of the proposed solution in a real network scenario, we utilize open-source real-world mobile traffic traces from Milan, Italy, released by

Telecom Italia [5] to obtain normalized load for each node i, $\lambda_t^{(i)}$. The dataset divides Milan into 100×100 grids, each 235m wide. Within these grids, real Call Detail Records (CDR) log calls, texts, and Internet activities every 10 minutes over a two-month period, from November 1^{st}, 2013. Each grid has $\text{grid}_{id} = (x_{id} + 1) + 100 \times y_{id}$ where $x_{id}, y_{id} \in [0, 99]$.

In this work, we calculate grid traffic volumes by aggregating CDR activities. We then normalize these volumes to derive $\lambda_t^{(i)}$. The 2-month dataset is divided into 30-minute intervals and grouped into 8 sets of weekly data (from week 45 to week 52 of 2013). The initial 5 weeks are for training, and the last 3 weeks are for testing. To match our design assumptions, we select 25 grids situated around the city center of Milan, with the iconic *Duomo di Milano* cathedral located in the center (grid_{5060}) to represent the Macro BS and 24 surrounding grids ($x_{5060} \pm 2, y_{5060} \pm 2$) to represent the Small base stations.

2) Experimental setups: The environment has been implemented using OpenAI Gym [17], utilizing the λ values collected from the Milan dataset, as discussed previously. The power consumption for each BS is determined based on the EARTH model from Table I, wherein this work solely concentrates on utilizing Macro and Micro values for MBS and SBS, respectively. We also assume that all BSs have the same maximum capacity ($\phi = 1$) for the sake of simplicity.

TABLE I: BSs power profiling from [16]

BS type	**Power consumption (W)**			**Slope** η
	Operational P_o	**Transmit** P_T	**Sleep** P_s	
Macro	130	20	75	4.7
Micro	56	6.3	39	2.6

We trained the policy with PPO, recognized for its proficiency in robustly handling complex environments with an emphasis on stability and transferability across diverse contexts, incorporating logit-level action masking for faster convergence to an efficient policy [18]. Fig. 3 shows the agent-environment interaction in PPO training. The training procedure is developed using Ray RLlib [19] with the configurations highlighted in Table II. Note that while the PPO policy employs stochastic sampling to encourage exploration, deterministic actions are taken when evaluating the performance of the learned policy.

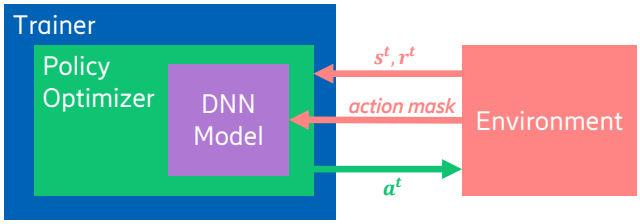


Fig. 3: High-level interaction of PPO training process

B. Benchmarking

To assess the performance of our DRL-based solution, we compare its performance with three baseline benchmarking methods. These methods comply with sequential (de)activation of BSs as formulated previously.

TABLE II: PPO parameters in Ray RLlib

training_iteration	425	*fcnet_hiddens*	[600, 600]
gamma γ	0.9	*vf_clip_param*	5000
learning_rate	10^{-4}	*clip_param* ϵ	0.3
kl_coeff	1.0	*entropy_coeff*	0.05
num_sgd_iter	25	*num_workers*	7
sgd_minibatch_size	1024	*num_envs_per_worker*	4
train_batch_size	34000	*num_gpus*	1

1) Ruled-Based (RB): A heuristic approach uses predefined thresholds (with T_1 set at 95% and T_2 at 25% in our case) as described in the algorithm below.

Rule-Based (RB) procedure

for $t = 0, 1, 2, \ldots, H-1$ **do**
  if normalized traffic load $> T_1$ **then**
    if the lowest $\lambda_t^{(i)} < T_2$ **then**
      Deactivates node i;
      Migrate $\lambda_t^{(i)}$ equally to all active neighbors;
    else
      Do nothing;
  else
    Activates node i with the highest traffic demand;

2) Action Scanning (AS): the AS strategy explores best actions within action masking in the following order.

Action Scanning (AS) procedure

- Initially, it identifies valid actions that result in the lowest value of l_t when applied to the environment.
- Among those actions, the search continues prioritizing actions with the least energy consumption.
- In case multiple actions remain, a tie-breaking mechanism is used to make a final selection among them.

3) All-ON method: In this method, all SBSs remain active, eliminating the need for traffic offloading and avoiding service disruption. However, it does not lead to any energy savings.

C. Performance metrics

This section introduces the metrics used to assess and compare the performance of the proposed solution against the benchmark algorithms.

1) Gain: This metric indicates the percentage gain of the total energy consumption (in Watts or W) of the system compared to the All-ON method.

2) Loss: This metric indicates the ratio between the total traffic loss over an episode and the All-ON method, as the latter always retains the original traffic loads.

3) Power consumption: The instantaneous energy consumption (W) over the week reflects the variations in network power consumption across different times of the week.

4) Normalized traffic load: This metric provides instantaneous normalized load at every timeslot t to demonstrate traffic preservation performance.

5) Number of deactivated SBSs: This metric reflects the behaviors of these given strategies throughout the week.

D. Evaluation results

Following the experimental setups, this section discusses the performance results of the proposed DRL solution compared with the designed benchmark strategies.

Fig. 4 illustrates the energy-saving gain and traffic loss under different policies in a typical week using the untrained dataset. It can be observed that the PPO policy with $\beta = 100$ (referred to as RL_100) exhibits the highest energy gain of 16.98% compared to all other methods. This is attributed to the effectiveness of minimizing energy consumption as described in (11). Still, due to a low traffic loss penalty, it incurs a loss of 6.73% of the total traffic. Upon a significant increase in β to 500, the policy (RL_500) becomes more conservative to maintain traffic loads, resulting in no observed traffic loss but causing a trade-off with a lower energy gain of 15.9%. Nevertheless, RL_500 still outperforms the two benchmark methods RB and AS, with slightly higher energy gains of 15.11% and 15.78%, respectively. Moreover, the two benchmark methods, RB and AS, still experience traffic losses of 6.03% and 12.95% each. This is because these methods select the best action that immediately benefits the current state, while the DRL policy is able to capture the impact over the entire week.

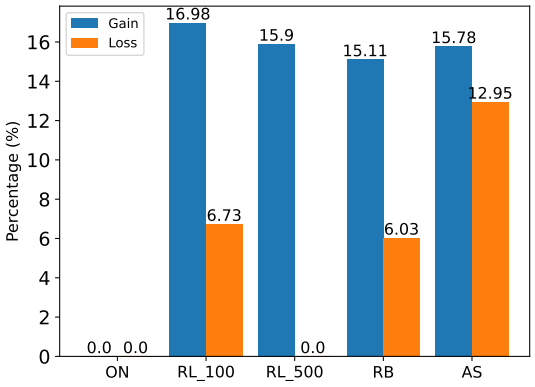
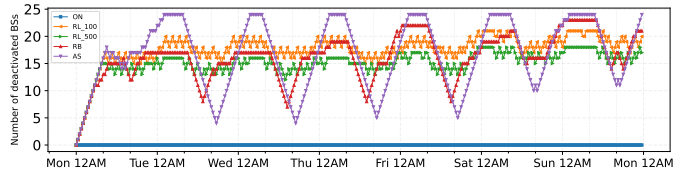
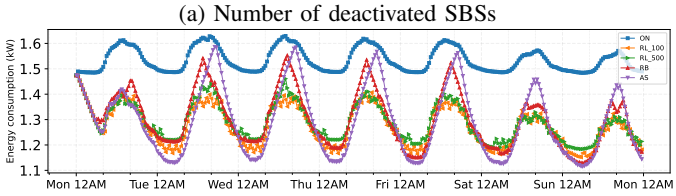


Fig. 4: Energy efficiency gain and traffic loss percentage during a regular week (Week 51).

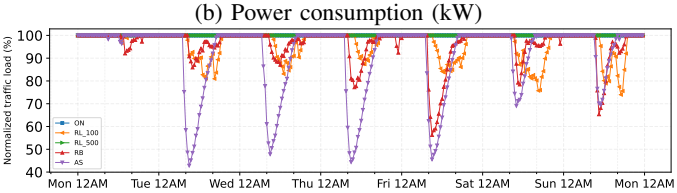
Further insight into the performance results throughout the week is presented in Fig. 5. It is evident from Fig. 5a that the number of deactivated SBSs in RL_500 is lower than in RL_100 due to RL_500 being more conservative regarding traffic continuity, leading to higher energy consumption compared to RL_100, as depicted in Fig. 5b. For the AS method, it attempts to deactivate as many SBSs as possible during midnight, which subsequently requires reactivating a significant number of them to handle high traffic during peak hours (Fig. 5a). As a result, the AS method exhibits minimal power consumption during off-peak hours and then compensates by consuming more energy during peak hours, as illustrated in Fig. 5b. While the RB method follows a similar sharp trend to AS, its behavior is more controlled due to the imposed thresholds. This characteristic also explains the lower traffic loss of RB compared to AS, as detailed earlier in Fig. 4. The normalized traffic load in Fig. 5c indicates that traffic losses primarily occur during peak hours, with the worst losses attributed to benchmark methods that failed to anticipate the traffic pattern in future time steps. In summary, DRL policies tend to result in fewer deactivated SBSs during peak hours compared to benchmark methods. The former exhibit consistent fluctuations, while the latter show sharper changes throughout the week. This distinction arises because DRL policies are designed for long-term benefit, whereas benchmark methods are inherently short-term in nature.



(a) Number of deactivated SBSs



(b) Power consumption (kW)



(c) Normalized traffic load (%)

Fig. 5: Performance results of a regular week in the test set (Week 51) under different strategies.

To evaluate how well the DRL policy generalizes to different scenarios, we test it during the holiday week (Week 52), where Christmas celebrations spanned from Tuesday to Thursday, as highlighted in Fig. 1. In Fig. 6, all sleeping strategies outperform their results from the standard week (Week 51, previously investigated in Fig. 4), achieving energy gains of up to 20% while reducing traffic loss to below 5.2%. This improved performance can be attributed to the significantly reduced mobile traffic demand around Christmas Day. With lighter traffic, there are naturally fewer losses, and more SBSs can be turned off. The DRL policies still manage to maintain good performance compared to benchmark methods, achieving the lowest traffic loss (none for RL_500) while ensuring reasonable energy savings.

Fig. 7 further illustrates performance results throughout the week. As depicted in Fig. 7a, the number of deactivated SBSs under DRL policies continues to exhibit consistent behavior even around Christmas Day. For the benchmark methods, a constant amount of deactivated SBSs is maintained throughout Christmas Day due to their reactive response to the given state, taking into account the low traffic during this period. In Fig. 7b, power consumption is similarly low during this

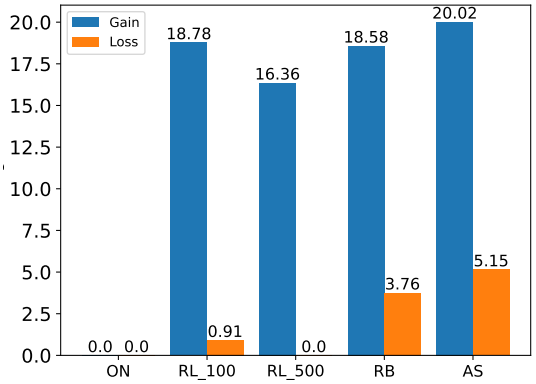
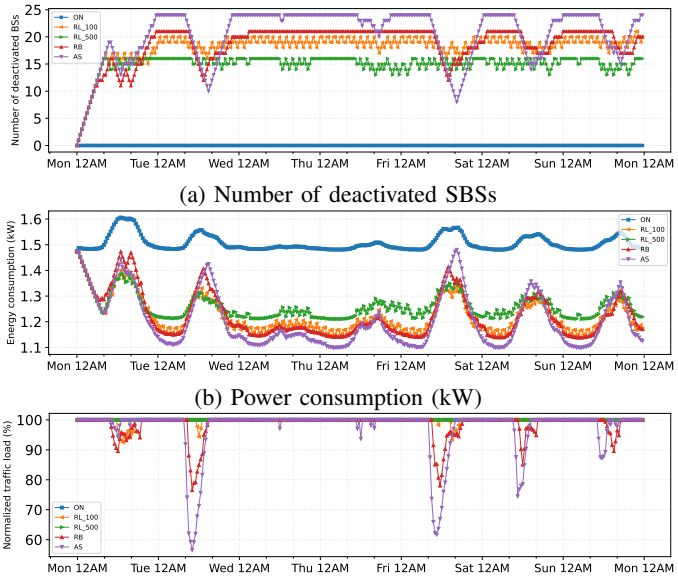


Fig. 6: Energy efficiency gain and traffic loss percentage during the Christmas week (Week 52).

time frame, since it aligns with the traffic loads defined in (1). Likewise, as expected in Fig. 7c, there is no considerable



(a) Number of deactivated SBSs

(b) Power consumption (kW)

(c) Normalized traffic load (%)

Fig. 7: Performance results of the Christmas week in the test set (Week 52) under different strategies.

traffic loss around Christmas Day. Notice that the behavior during the regular days of this week consistently follows the pattern illustrated previously in Fig. 4.

V. Conclusion and Future Work

In this paper, we present a DRL application tailored for the Base Station sleep control problem, featuring novel offloading options to neighboring nodes. Through the integration of PPO and action masking, the derived policy achieves an energy savings gain of approximately 16% while ensuring consistent traffic performance and showcasing notable generalization capabilities across holiday scenarios. This signifies a promising direction for sustainable 5G network operations.

In future work, we plan to utilize more recent mobile traffic patterns from live networks to reflect current network demands more accurately. We also intend to investigate a multi-agent reinforcement learning approach in which each BS operates as an independent agent. This approach can enable turning off multiple BSs simultaneously, enhancing energy efficiency. Additionally, we would consider formulating the sleep control at the carrier level, which could allow smoother service transitions and lower switching costs.

References

[1] POSTnote. Energy consumption of ICT. [Online]. Available: https://post.parliament.uk/research-briefings/post-pn-0677/

[2] GSMA. Mobile Net Zero: State of the Industry on Climate Action 2023.

[3] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE communications surveys & tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[4] D. López-Pérez, A. De Domenico, N. Piovesan, G. Xinli, H. Bao, S. Qitao, and M. Debbah, "A Survey on 5G Radio Access Network Energy Efficiency: Massive MIMO, Lean Carrier Design, Sleep Modes, and Machine Learning," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 653–697, 2022.

[5] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of Milan and the Province of Trentino," *Scientific data*, vol. 2, no. 1, pp. 1–15, 2015.

[6] H. Tabassum, U. Siddique, E. Hossain, and M. J. Hossain, "Downlink Performance of Cellular Systems With Base Station Sleeping, User Association, and Scheduling," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5752–5767, 2014.

[7] A. I. Abubakar, M. S. Mollel, M. Ozturk, S. Hussain, and M. A. Imran, "A lightweight cell switching and traffic offloading scheme for energy optimization in ultra-dense heterogeneous networks," *Physical Communication*, vol. 52, p. 101643, 2022.

[8] J. Liu, B. Krishnamachari, S. Zhou, and Z. Niu, "DeepNap: Data-driven base station sleeping operations through deep reinforcement learning," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4273–4282, 2018.

[9] Q. Wu, X. Chen, Z. Zhou, L. Chen, and J. Zhang, "Deep Reinforcement Learning With Spatio-Temporal Traffic Forecasting for Data-Driven Base Station Sleep Control," *IEEE/ACM Transactions on Networking*, vol. 29, no. 2, pp. 935–948, 2021.

[10] M. Ozturk, A. I. Abubakar, J. P. B. Nadas, R. N. B. Rais, S. Hussain, and M. A. Imran, "Energy optimization in ultra-dense radio access networks via traffic-aware cell switching," *IEEE Transactions on Green Communications and Networking*, vol. 5, no. 2, pp. 832–845, 2021.

[11] G. Sun, D. Ayepah-Mensah, R. Xu, V. K. Agbesi, G. Liu, and W. Jiang, "Transfer Learning for Autonomous Cell Activation Based on Relational Reinforcement Learning With Adaptive Reward," *IEEE Systems Journal*, vol. 16, no. 1, pp. 1044–1055, 2022.

[12] H. Ju, S. Kim, Y. Kim, and B. Shim, "Energy-Efficient Ultra-Dense Network With Deep Reinforcement Learning," *IEEE Transactions on Wireless Communications*, vol. 21, no. 8, pp. 6539–6552, 2022.

[13] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[14] M. Tayyab, X. Gelabert, and R. Jäntti, "A survey on handover management: from lte to nr," *IEEE Access*, vol. 7, pp. 118 907–118 930, 2019.

[15] Ericsson. Massive MIMO handbook. [Online]. Available: https://www.ericsson.com/massive-mimo

[16] G. Auer, V. Giannini, C. Desset, I. Godor, P. Skillermark, M. Olsson, M. A. Imran, D. Sabella, M. J. Gonzalez, O. Blume *et al.*, "How much energy is needed to run a wireless network?" *IEEE wireless communications*, vol. 18, no. 5, pp. 40–49, 2011.

[17] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "OpenAI Gym," *arXiv preprint arXiv:1606.01540*, 2016.

[18] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *arXiv preprint arXiv:2006.14171*, 2020.

[19] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "RLlib: Abstractions for distributed reinforcement learning," in *International conference on machine learning*. PMLR, 2018, pp. 3053–3062.