ML-Aided Traffic-Aware Base Station Sleep Threshold Design with User Throughput Guarantees

Ahmed AlAlwani*‖, Abdulrahman Itman*, Ayman Gaber§†, Mohamed Zaki†,
Mohammad Galal Khafagy§, Karim Banawan*, Karim Seddik*
* School of Sciences and Engineering, The American University in Cairo (AUC), New Cairo, Egypt.
§ Mobile Network Development, Technology Department, Vodafone Egypt, Smart Village, Giza, Egypt.
‖ Network Strategy & Engineering, Vodafone Intelligent Solutions (NSE _VOIS), Smart Village, Giza, Egypt.
† School of Information Technology and Computer Science, Nile University, Giza, Egypt.
E-mail: {ahmedalalwani, a_itman, karim.banawan, kseddik}@aucegypt.edu,
{ayman.gaber, mohamed.khafagy}@vodafone.com, moha.zaki@nu.edu.eg

***Abstract*—As the demand for mobile data continues to grow, the energy consumption of mobile networks becomes a major concern. Specifically, base stations account for over 76% of energy usage in mobile networks. We consider a two-tier cellular system, one offering basic coverage while the other offers extra capacity. To save energy, existing network features can opportunistically shut down capacity layer cells when physical resources are lightly utilized. Nevertheless, this is performed without guaranteeing coverage cells can maintain the sought user-centric service quality. To address this challenge, we propose a machine learning (ML)-aided search approach that dynamically designs energy-saving configurations for each cell in each hour while being constrained with a pre-defined quality of service (QoS) measure. This ML model was trained using data collected from 10,283 cells of a live network. We introduce two different approaches to provide these settings: Adaptive QoS Threshold Optimization Algorithm (AQTOA) and an Exhaustive Search (ES) baseline. AQTOA is a low complexity ML-aided search algorithm designed to determine the optimal shutdown threshold for capacity cells while ensuring that QoS requirements are met. Through extensive *live network* experimentation, the AQTOA results indicate a 1.8% improvement in energy savings compared to the earlier static settings models while maintaining a more strict QoS level than the one addressed in the previous work.**

***Index Terms*—cellular networks, energy saving, green mobile networks, machine learning, RAN intelligent controller, service management.**

I. Introduction

In the era of ubiquitous connectivity, mobile networks have become an indispensable part of modern life, enabling seamless communication, access to information, and a plethora of digital services. However, this surge in mobile network usage has come at a significant environmental cost. Mobile networks are responsible for up to 3% of the global energy consumption [1], a figure projected to further increase as network traffic continues to grow. This substantial energy consumption not only translates into higher operating costs for mobile network operators but also contributes to greenhouse gas emissions by approximately 2% [2] and exacerbates climate change. Radio access networks (RANs) consume over 76% of energy in mobile networks according to [3] with the power amplifier (PA) being the most energy-intensive component, followed by baseband processing, radio unit (RU) power requirements, and cooling systems. Given the pressing need for sustainability, energy efficiency has become a critical focus for network operators and researchers alike. Optimizing energy consumption in mobile networks involves finding a balance between network performance and energy savings. Although reducing energy consumption is essential, maintaining the quality of service (QoS) experienced by mobile users is vital. QoS includes key metrics such as data rate, latency, and call drop rate, which are crucial to ensure a satisfactory user experience.

Previous research has explored various techniques for energy saving in mobile networks, including network densification [4]. In [5], a SARSA-based algorithm was developed to choose the best sleep mode based on the time and traffic load of the base station while ensuring that the (de)activation time does not cause service interruption. A reinforcement learning (RL) approach that minimizes energy consumption in ultra-dense networks by intelligently switching off small cells based on traffic load without impacting QoS was proposed in [6]. In [7], we previously designed static energy-saving configurations tailored to fit the entire day, successfully demonstrating the effectiveness of machine learning (ML) in optimizing mobile network energy efficiency. The authors in [8] addressed the energy optimization problem from the point of view of multiple-input multiple-output (MIMO) resource usage. Two ML approaches, based on multilayer perception (MLP) and recurrent neural network (RNN), were developed to decide whether the MIMO feature is needed or to turn it off based on traffic load. However, these techniques often involve trade-offs between energy consumption and QoS. For example, network densification can improve coverage and capacity but increases the overall energy consumption of the network. Similarly, sleep modes for base stations can reduce energy consumption during periods of low traffic but can lead to reduced coverage and degraded QoS if traffic spikes unexpectedly. Static energy-saving settings result in conservative settings, potentially missing additional energy-saving opportunities since the thresholds are designed to accommodate varying traffic conditions throughout the day.

To the best of our knowledge, no prior work has addressed the problem of dynamically shutting off capacity layer cells throughout the day. In this paper, we propose an improved ML approach to dynamically optimize network settings for

each cell and hour. This dynamic approach addresses the limitations of static energy-saving techniques by adapting to real-time traffic patterns and maintaining QoS. Our ML model was tested on a live network and proved to aid the network proactively adjust its configuration by selectively turning off the capacity cells while maintaining the coverage layer cells.

The remainder of this paper is structured as follows: Section II investigates the system model and formulates the problem. Section III presents our ML algorithm as well as two distinct proposed approaches to address the problem. Section IV discusses the results of testing our algorithm of a live network and provides a comparative analysis with our previous work. Section V concludes the paper, and Section VI outlines potential directions for future research.

II. System Model and Problem Formulation

Our system model comprises a two-tier cellular network. Each tier of radio cells offers a distinct function, namely 1) basic coverage and 2) capacity. As the tier name may suggest, coverage cells are responsible for continuous network service availability utilizing a low-frequency range (e.g., 1800 MHz band). This corresponds to wider coverage and smaller bandwidth. Capacity cells, on the other hand, operate at higher frequency bands (e.g., 2600 MHz band), providing additional capacity through larger bandwidth but offering reduced coverage. In this work, we assume that, in each serving sector, a single coverage cell coexists with two capacity cells. Furthermore, we assume that the two capacity cells, hosted on the same RU, are simultaneously activated or deactivated offering an opportunity to shut down the most energy-consuming element in the base station, the PA. The system imposes a network feature that monitors the physical resource blocks (PRBs) used across all three cells and calculates utilization relative to the coverage cell's available PRBs. If this utilization falls below a specified threshold, both capacity cells are deactivated.

Our goal is to minimize energy consumption by opportunistically shutting down capacity cells whenever possible under a strict constraint that the coverage cells can handle all the traffic without compromising the mandated QoS levels defined by a minimum downlink (DL) average user throughput. Our objective is to determine the optimal threshold that maximizes energy savings while preserving QoS.

III. Proposed Approaches

Our approach to satisfy the aforementioned objective includes an ML-based predictive model accompanied by a search algorithm. The ML-based predictive model predicts the average DL throughput based on a subset of key performance indicators (KPIs), which are readily available in cellular networks. The search algorithm identifies an appropriate shutting-down threshold for the capacity cells. We propose two search algorithms, namely, Adaptive QoS Threshold Optimization Algorithm (AQTOA), and an Exhaustive Search (ES) baseline.

A. Dataset and ML-Based Throughput Predictive Model

To build our ML model, we collected a dataset from a 4G live network in Egypt. Data points were collected from 10,283 cells. Each data point includes the following KPIs, which were carefully selected based on correlation analysis and domain expertise. Although the correlation matrix is not included due to space constraints, the selected KPIs were found to have strong correlation to the target variable. The same approach can be easily applied to a 5G network.

- DL PRB Utilization (%)
- Average DL Active Users
- DL Traffic Volume [MB]
- Average Modulation and Coding Scheme (MCS)
- Average Rank Indicator (RI)
- Average DL User Throughput [Mbps]

We treat the first 5 KPIs as features (i.e., inputs) of the ML model, while the average DL user throughput is the target variable (i.e., prediction output). The number of used data points is 1,370,886. The dataset is divided as follows: 80% of the data points were used for the training phase, 10% for validation, and the rest 10% were used for testing.

The collected data is used to train and test an XGBoost regressor model for throughput prediction. XGBoost regressor is a scalable and efficient implementation of gradient boosting designed for regression tasks [9]. Through a careful process of fine-tuning, we settled for the following set of hyper-parameters; 1000 estimators, 0.08 learning rate, 10 maximum depth, and 0.9 subsample.

B. Adaptive QoS Threshold Optimization Algorithm (AQTOA)

Our proposed scheme uses the pre-trained regression model. This model is fed the aggregated KPIs of the coverage cell and the two capacity cells as input to predict the corresponding average DL user throughput. The reason for having this aggregation is to simulate the capacity cells shutdown case. Our algorithm then iteratively changes the aggregated DL PRB utilization (%) with a dynamic step to get the highest DL PRB utilization (%) threshold that meets our QoS constraint.

AQTOA, presented in Algorithm 1, is described as follows:

1) **Aggregation and Prediction**: Lines 6-11 aggregate the data of the coverage cell and capacity cells and predict the throughput of the aggregated cell. This throughput is checked whether it falls between the QoS constraint and the QoS constraint + prediction error (Δ). If so, the corresponding utilization is chosen as the needed threshold. If not, utilization is incremented/decremented depending on the predicted throughput by the initial step size.
2) **Threshold Design**: In lines 19-39, the utilization is incremented/decremented on each iteration depending on the predicted throughput and the previous step until the predicted throughput falls within the desired range or the utilization reaches the min/max values.
3) **Handling Infinite Loops**: During looping, all (utilization, throughput) pairs are saved into a dictionary, which is useful to avoid having an infinite loop. This is detected by checking whenever a previously visited pair is revisited. This case occurs when 2 consecutive utilization values with corresponding predicted throughput satisfy $QoS \leq throughput_{pred} \leq QoS + \Delta$. In such a case,

the smaller utilization value is chosen to be our desired threshold. This can be seen in lines 41-42

C. Exhaustive Search (ES)

Our aim is to implement an ES algorithm for the problem to serve as a baseline for performance evaluation. In this approach, we are defining a 5D space consisting of (DL PRB Utilization (%), Average DL Active Users, DL Traffic Volume, Average MCS, Average RI), Those features are fed as input to the ML predictor to get the corresponding average DL user throughput then we are able to set a threshold for each tuple.

We start by defining the sampling resolution of the aforementioned space. Here, n, m, l, p, and o represent the number of points in the parameter space for DL PRB Utilization (%), Average DL Active Users, DL Traffic Volume (MB), Average MCS, and Average RI, respectively. The sampling resolution was adjusted to balance the computational efficiency as well as match the level of detail required for our algorithm as follows:

- n: 100 uniform samples in $[0, 100]$.
- m: 60 uniform samples in $[0.25, 15]$.
- l: 60 uniform samples in $[0.5, 30]$.
- p: 28 uniform samples in $[1, 28]$.
- o: 21 uniform samples in $[1, 2]$.

We then predict the corresponding throughput for each point in that space using the regression model in subsection III-A. We then group and filter all the resulting data to get the highest possible threshold for each (Average DL Active Users, DL Traffic Volume (MB), Average MCS, Average RI) tuple that meets our QoS constraint.

D. Computational Complexities

Table I compares the computational complexities between the proposed AQTOA and the ES approach for designing the required thresholds over n_{data} hours.

The ES approach consists of two offline steps; 1) creating the table and 2) grouping all possible tuples/combinations. This requires significant computational resources and can be precomputed. Furthermore, the third step, which is performed online to calculate the thresholds, introduces a high computational cost. Notably, for small values of n_{data}, the overall complexity is dominated by the term $m \times l \times p \times o$, which grows rapidly with the size of the parameter space.

In contrast, the AQTOA offers a significantly lower computational complexity, as it eliminates the need for exhaustive exploration of the parameter space. By leveraging an adaptive iterative approach, AQTOA focuses on optimizing the thresholds directly during the online step, making it more efficient and scalable, especially when handling large datasets. This reduction in computational overhead highlights the superiority of AQTOA in terms of complexity.

IV. Performance Evaluation

In this work, and without loss of generality, we impose a DL average user throughput of 8 Mbps as the QoS constraint, which aligns with the market demands and application requirements of most operators and avoids any service-level violations. In addition, we also impose upper and lower bounds

Algorithm 1 Adaptive QoS Threshold Optimization (AQTOA)

1: Initialize $threshold = 0$
2: Initialize $LastAct = 0$
3: Initialize $U_{step} = 1$
4: Initialize $VisitedStates = \{\}$
5: Let Δ be the prediction error
6: Aggregate the coverage and capacity cells KPIs to get the aggregated cell (c_{agg}) KPIs.
7: Initialize $U = utilization_{aggregated}$.
8: Predict DL average user throughput of c_{agg} ($\hat{T}$).
9: **if** $QoS \leq \hat{T} \leq QoS + \Delta$ **then**
10: $threshold \leftarrow U$
11: **else**
12: **if** $\hat{T} \geq QoS + \Delta$ **then**
13: $U \leftarrow U + U_{step}$
14: $LastAct \leftarrow 1$
15: **else if** $\hat{T} \leq QoS$ **then**
16: $U \leftarrow U - U_{step}$
17: $LastAct \leftarrow 0$
18: **end if**
19: Predict $\hat{T}$
20: **while** $\sim (QoS \leq \hat{T} \leq QoS + \Delta)$ **do**
21: $VisitedStates[U] \leftarrow \hat{T}$
22: **if** $\hat{T} > QoS + \Delta$ **and** $LastAct = 1$ **then**
23: $U_{step} \leftarrow U_{step} \times 2$
24: $LastAct \leftarrow 1$
25: $U \leftarrow U + U_{step}$
26: **else if** $\hat{T} < QoS$ **and** $LastAct = 1$ **then**
27: $U_{step} \leftarrow 1$
28: $LastAct \leftarrow 0$
29: $U \leftarrow U - U_{step}$
30: **else if** $\hat{T} > QoS + \Delta$ **and** $LastAct = 0$ **then**
31: $U_{step} \leftarrow 1$
32: $LastAct \leftarrow 1$
33: $U \leftarrow U + U_{step}$
34: **else if** $\hat{T} < QoS$ **and** $LastAct = 0$ **then**
35: $U_{step} \leftarrow U_{step} \times 2$
36: $LastAct \leftarrow 0$
37: $U \leftarrow U - U_{step}$
38: **end if**
39: $U \leftarrow \max(U_{min}, U)$
40: $U \leftarrow \min(U, U_{max})$
41: **if** U in $VisitedStates$.keys() **then**
42: Check for 2 consecutive U values with corresponding $\hat{T} > QoS + \Delta$ and $< QoS$ respectively
43: $U \leftarrow \min_{k \in \{i, i+1\}} U_k$
44: break
45: **end if**
46: Predict $\hat{T}$
47: **if** $\hat{T} > QoS + \Delta$ **and** $U = U_{max}$ **then**
48: break
49: **else if** $\hat{T} < QoS$ **and** $U = U_{min}$ **then**
50: break
51: **end if**
52: **end while**
53: **end if**
54: $threshold \leftarrow U$

TABLE I: Computational Complexity

Approach		Complexity
AQTOA		$O(n_{data})$
ES	Creating Table	$O(n \times m \times l \times p \times o)$
	Grouping and filtering	$O(n \times m \times l \times p \times o)$
	Thresholds design	$O(n_{data} + m \times l \times p \times o)$

of 80% and 20%, respectively, for the threshold. To account for the ML model prediction error, we take $\Delta = 10\%$. In the evaluation process, we first compare the designed thresholds using both approaches. Then, we continue our assessment with our proposed algorithm AQTOA and focus on three key metrics:

1) **Shutdown time percentage for capacity cells**: This metric indicates the percentage of time, in which capacity cells are deactivated. A higher percentage reflects more energy-saving behavior.
2) **Energy consumption (kWh)**: Measurement of total energy consumed during the evaluation period.
3) **Average user throughput**: Assessing the user experience by monitoring the average data rate in the coverage cells. This metric ensures that the implemented energy-saving measures maintain the QoS constraint.

To evaluate the effectiveness of our proposed energy-saving framework, three distinct configurations were implemented on 10 different sectors across the network using the thresholds obtained from AQTOA.

1) **Low traffic window**: The energy-saving feature was activated only during the low-traffic period between 2 AM and 7 AM using a fixed threshold.
2) **Static settings**: A fixed threshold value was applied to the energy-saving feature throughout the entire day (the lowest obtained hourly threshold).
3) **Dynamic settings**: The energy-saving feature was activated based on hourly thresholds that adjust according to the traffic load and PRB utilization (%) per hour.

TABLE II: XGBoost model performance metrics

Model	MAE	MAPE	R^2
XGBoost	1.66	9.5%	0.928

Table II shows the performance metrics of the model used for throughput prediction.

Fig. 1 illustrates the average utilization thresholds obtained using both the proposed AQTOA and the ES approach. The thresholds shown represent the average across 10 sectors. The results demonstrate that both methods achieve nearly identical thresholds. However, AQTOA achieves these results with significantly lower computational complexity, making it a more efficient and practical solution compared to the ES.

Fig. 2 presents the hourly ratio of time during which all capacity cells were shut down for the three configurations. While both static and dynamic approaches outperformed the low traffic window, an increase in the capacity cell shutdown time by 5.23% is achieved using the dynamic settings.

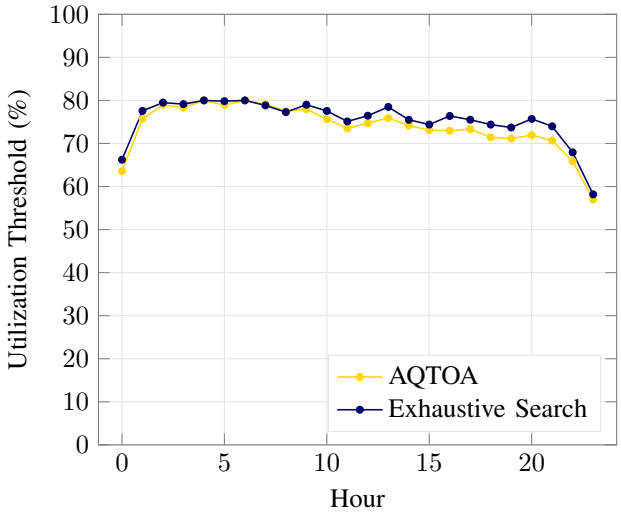

Fig. 1: Thresholds Comparison.

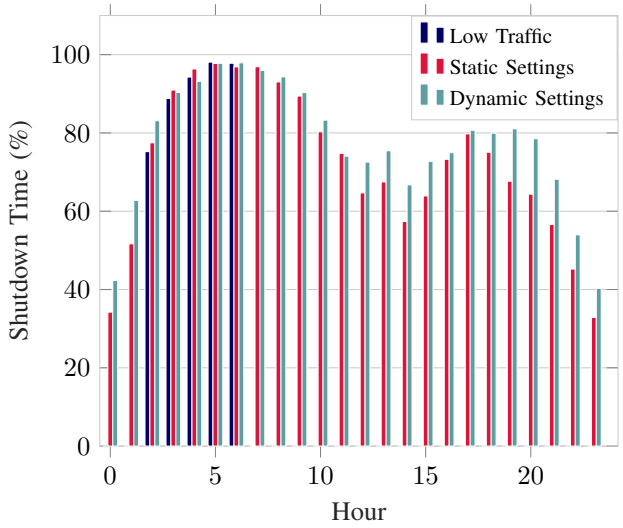

Fig. 2: Percentage of capacity cells shut-down time.

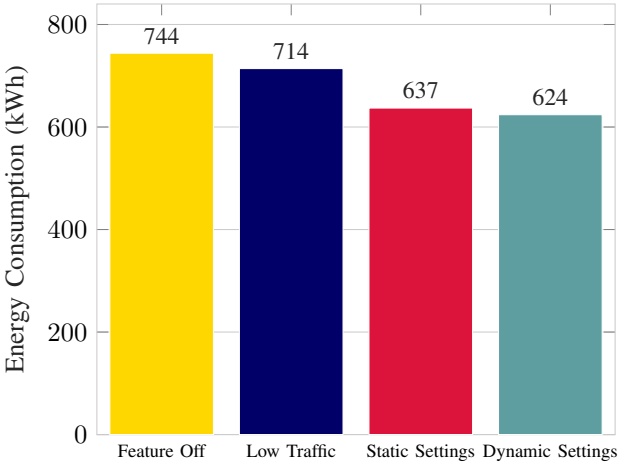

Fig. 3: Energy consumption (kWh).

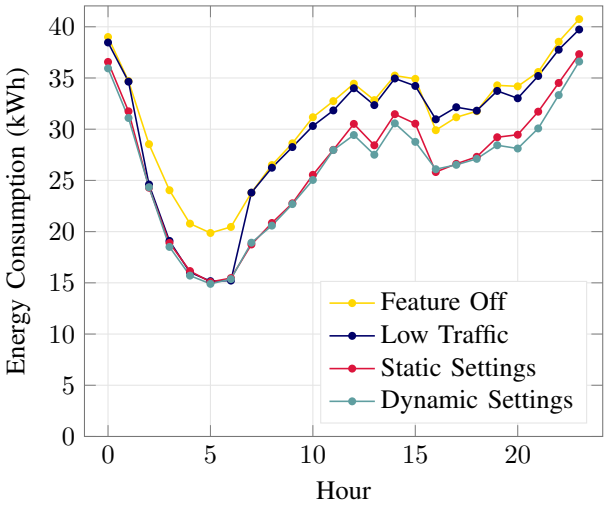

Fig. 4: Hourly energy consumption.

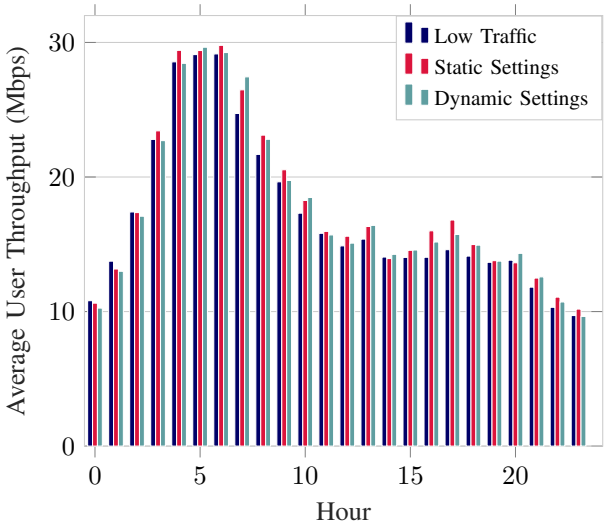

Fig. 5: Hourly average downlink user throughput.

Fig. 3 shows the total energy consumption for the coverage and capacity layers RUs which shows 16.1% (120kWh) overall energy saving using the proposed dynamic settings, compared to only 14.3% (107kWh) using the static settings. The dynamic settings approach achieved 1.8% more energy saving (13kWh) compared to the old static settings approach.

In Fig. 4 we compare the hourly energy consumption for the 4 scenarios. The figure shows that the proposed dynamic settings scenario offers less energy consumption than the static settings due to the increase in the shutdown time.

In Fig. 5 the hourly average DL user throughput over the coverage layer is demonstrated which shows that the gain in energy consumption and shutdown time comes only at a cost of 1.05% decrease in the average DL user throughput.

V. Conclusion

This paper presented a novel ML-based approach to optimize energy consumption in multi-layer mobile networks. By dynamically adjusting capacity layer shutdown thresholds on an hourly basis, our proposed algorithm demonstrated a 1.8% improvement in energy savings compared to the static threshold approach. This enhancement was achieved while rigorously adhering to predefined QoS constraints, ensuring an uninterrupted user experience. The results underscore the potential of intelligent, data-driven techniques in optimizing network performance and resource efficiency.

Future research directions can explore the broader environmental impact of these approaches and quantify the reduction in greenhouse gas emissions. Furthermore, investigating hierarchical energy-saving techniques, such as selectively shutting down MIMO branches before deactivating entire capacity cells, is a promising avenue for further optimization. Given the growing adoption of open radio access network (ORAN) architectures, adapting the proposed framework to the distributed and cloud-native nature of ORAN networks is an area for future research.

References

[1] M. Pickavet, W. Vereecken, S. Demeyer, P. Audenaert, B. Vermeulen, C. Develder, D. Colle, B. Dhoedt, and P. Demeester. Worldwide energy needs for ICT: The rise of power-aware networking. In *2008 2nd International Symposium on Advanced Networks and Telecommunication Systems*, 2008.

[2] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok. The global footprint of mobile communications: The ecological and economic perspective. *IEEE Commun. Mag.*, 49(8):55–62, 2011.

[3] P. Frenger, Y. Jading, and A. Nader. Energy performance of 6g radio access networks: A once in a decade opportunity. Technical report, Ericsson, 2024.

[4] A. Arbi and T. O'Farrell. Energy efficiency in 5G access networks: Small cell densification and high order sectorisation. In *Proc. 2015 IEEE International Conference on Communication Workshop (ICCW)*, London, UK, Jun. 2015.

[5] M. Masoudi, M. G. Khafagy, E. Soroush, D. Giacomelli, S. Morosi, and C. Cavdar. Reinforcement learning for traffic-adaptive sleep mode management in 5G networks. In *Proc. IEEE PIMRC'20*, London, UK, Oct. 2020.

[6] 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*, 5(2):832–845, 2021.

[7] M. Zaki, A. Gaber, M. G. Khafagy, M. Beshara, and N. Abdelbaki. Autonomous traffic-aware and QoS-constrained capacity cell shutdown for green mobile networks. In *Proc. 2023 IEEE 5th Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, Cairo, Egypt, 2023.

[8] M. Aboelwafa, M. Zaki, A. Gaber, K. Seddik, Y. Gadallah, and A. Elezabi. Machine learning-based MIMO enabling techniques for energy optimization in cellular networks. In *Proc. IEEE CCNC'20*, Las Vegas, NV, USA, Jan. 2020.

[9] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.