IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received April 21, 2020, accepted May 13, 2020, date of publication May 15, 2020, date of current version June 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2995057

Deep Convolutional Neural Network Assisted Reinforcement Learning Based Mobile Network Power Saving

SHANGBIN WU[1], YUE WANG[1], (Senior Member, IEEE), AND LU BAI[2]
[1]Samsung R&D Institute U.K., Staines-upon-Thames TW18 4QE, U.K.
[2]School of Cyber Science and Technology, Beihang University, Beijing 100191, China

Corresponding author: Lu Bai (lu_bai@buaa.edu.cn)

This work was supported by Samsung Research.

ABSTRACT This paper addresses the power saving problem in mobile networks. Base station (BS) power and network traffic volume (NTV) models are first established. The BS power is modeled based on in-house equipment measurement by sampling different BS load configurations. The NTV model is built based on traffic data in the literature. Then, a threshold-based adaptive power saving method is discussed, serving as the benchmark. Next, a BS power control framework is created using Q-learning. The action-state function of the Q-learning is approximated via a deep convolutional neural network (DCNN). The DCNN-Q agent is designed to control the loads of cells in order to adapt to NTV variations and reduce power consumption. The DCNN-Q power saving framework is trained and simulated in a heterogeneous network including macrocells and microcells. It can be concluded that with the proposed DCNN-Q method, the power saving outperforms the threshold-based method.

INDEX TERMS Power saving, deep convolutional neural network, reinforcement learning.

I. INTRODUCTION

A. BACKGROUND

In the era of data, information is flowing in an unprecedented way anytime everywhere. It is reported in [1], that the number of mobile broadband subscriptions will be approaching eight billion by 2025. The amount of mobile data traffic is anticipated to grow at an exponential pace, reaching 160 extrabyte (EB, 10^{18} bytes) per month within the same time period. New emerging applications such as augmented reality (AR), virtual reality (VR), vehicle to everything (V2X), and internet of things (IoTs) are projected to have increasing contribution to the massive growth of data traffic.

The fifth generation (5G) mobile network (MN) [2]–[4] has introduced groundbreaking technologies in order to satisfy this growing demand of data traffic. Millimeter-wave (mmWave), for instance, is a well-recognized solution as high bandwidths in mmWave are able to provide more available radio resources. In addition, the use of massive multiple-input multiple-output (MIMO), which equips base stations (BSs) and user equipments (UEs) with an increasing number of antennas, can reduce intercell interference and boost network throughput. Most importantly, reducing cell size and increasing cell density have been the main source of enhancing network throughput [5], [6]. There is no exception in 5G networks, as they are expected to significantly scale up cell densities.

However, denser cells come at the cost of larger MN power consumption, which increases green house gas emissions and accelerates global warming. Operators such as Vodafone, have targeted to reduce green house gas emission by 50% by 2025 [7]. Reducing power consumption can not only reduce green house gas emission, but also reduce operating cost of MNs. To tackle the problem of MN power saving, practical models for BS power consumption and data traffic as well as smart resource management techniques are required.

Authors in [8] measured BS power in real equipment and proposed a number of linear power models in terms of load for the remote unit (RU) only. In [9], power models were built for components in a BS, such as power amplifier and filter. It concluded that power consumption in downlink was dominant. Measurement of voice traffic was presented in [10]. More generally, the white paper [11] revealed traffic patterns of various applications in reality. Both measurement

The associate editor coordinating the review of this manuscript and approving it for publication was Ivan Wang-Hei Ho.

reports showed that network traffic volume (NTV) was normally higher during weekdays and lower during weekends.

There are a number of classic cell on/off algorithms, including optimizing user association, optimizing BS coverage, traffic prediction, and heterogeneous deployment [12]. In [10], a concept known as network-impact was proposed, which can be calculated by the maximum of sum of the original BS load and the additional load increments brought by neighboring BSs. The algorithm in [10] required heuristic parameters. In [13], the user association to BSs and dynamic BS operations were jointly optimized for the purpose of improving energy efficiency. The switching on and off of BSs relied on a greedy algorithm and heuristic parameters. Authors in [14] and [15] proposed algorithms to adjust cell coverage to reduce power consumption. Methods of traffic pattern and BS energy consumption pattern prediction were discussed in [16]. In [17], stochastic geometry was used to model distributions of macrocells and low-power cells. The minimum separation distance between a macrocell and a low-power cell was optimized to reduce interference and power consumption. Besides the discoveries in academia, industry has designed schemes to reduce power consumption as well. The 3GPP 5G new radio (NR) [18] has replaced the always-on cell-specific reference signal (CRS) in 4G long-term evolution (LTE) [19] with a novel reference signal framework, including demodulation reference signals (DMRSs) and channel state information reference signal (CSI-RSs). These are user-specific and flexibly configurable. As a result, power consumption is reduced when there is no traffic or measurement to certain UEs.

Besides classic methods, machine learning (ML) based methods have attracted researchers to explore new approaches to solve the MN power saving problem [20], [21]. Having assumed accessible location information, [22] proposed a reinforcement learning (RL) based method to predict movement of UEs and dynamically adjust the powers of the handover target cell and the original cell. Authors [23] used RL to optimize durations of different sleep modes to reduce power consumption. These RL based methods did not considered realistic power and traffic models. Also, loads of BS were not directly controlled. In this paper, a centralized deep RL based method is proposed, to intelligently control BS loads according to realistic power and traffic models. In a multi-cell mobile network, it is straightforward to expect a distributed architecture where each cell is equipped with one RL agent. As a result, multiple agents perform RL individually. However, a distributed architecture suffers from the moving target problem [24], where the behavior of each agent can impact on behaviors of other agents. On the contrary, the centralized architecture used in this paper assumes one agent only controlling all cells in the mobile network. This can accelerate convergence.

B. CONTRIBUTIONS

The contributions of this paper are listed as follows.

1) A power model and a NTV model for base stations are proposed. The power model is established based on measurement data in real-world base stations. More importantly, detailed power consumption in a data unit (DU) and a RU is shown. The NTV model is obtained from measurement data in the literature. These two models are able to provide realistic descriptions on the dynamics of network power consumption in terms of time.
2) A threshold-based power saving method is proposed. This method uses a cell load adaptation equation to update cell loads to adjust power consumption.
3) Most importantly, a deep learning approach, i.e., deep convolutional neural network based Q-learning (DCNN-Q), for power saving is proposed. The proposed method uses a centralized architecture and Q-learning to control cell loads, with the action-state function approximated by a DCNN. The DCNN not only takes a one-dimensional (1D) load vector as input, but also a two-channel two-dimensional (2D) image containing information of instantaneous NTV requirement and network throughput.

The rest of this paper is organized as follows. Section II proposes a power model based on measurement data and a NTV model based on literature data. Problem description and system model are presented in Section III. The benchmark method, i.e., the threshold-based method is investigated in Section IV. Our proposed DCNN-Q method is discussed in detail in Section V. Simulation/numerical results and analysis are presented in Section VI. Conclusions are drawn in Section VII.

II. POWER MODEL AND NETWORK TRAFFIC MODEL

A. POWER MEASUREMENT IN REAL-WORLD EQUIPMENT

The power measurement was conducted in our in-house lab on real LTE DU and RU equipment. Both power of DU and RU in terms of different settings of load were measured, by installing a power meter to both the DU and RU power cables. Load of the system is the ratio of the number of active physical resource blocks (PRBs) over the number of total available PRBs. This was configured using the orthogonal channel noise simulator (OCNS) functionality via command line interface (CLI) during the measurement. The flowchart of measurement is depicted in Fig. 1. A typical set of load settings, i.e., 0%, 50%, 100%, were configured. The total measurement period lasted for 10 hours and the readings of

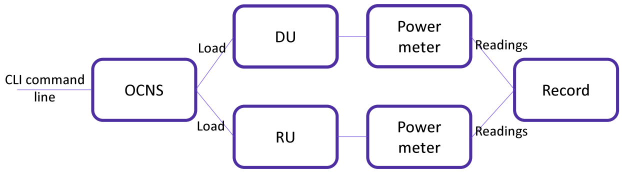


FIGURE 1. DU and RU power measurement flowchart.

power meter were recorded every 15 minutes. As a result, there were 41 measured samples in total.

B. POWER MODEL

After obtaining the measured power data, a power model can be established. In the paper, linear models for DU power P^{DU} and RU power P^{RU} based on measured data are proposed, i.e.,

$$\mathrm{P}^{\mathrm{DU}}(l) = 1.68l + 266.98 \tag{1}$$

$$\mathrm{P}^{\mathrm{RU}}(l) = 153.50l + 93.95 \tag{2}$$

where l represents the load of the unit. Both proposed models, the total power model $\mathrm{P}^{\mathrm{Total}}(l)$, and measured data are shown in Fig. 2. It can be observed that the power of DU is not sensitive to the change of load. On the other hand, the change of load can result in changing the power of RU from 94 W to 247 W.

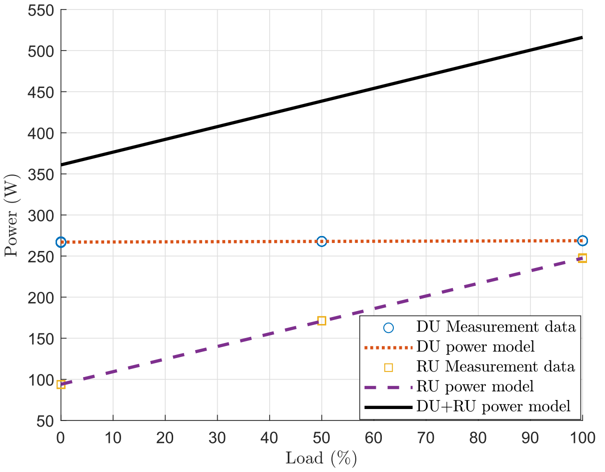


FIGURE 2. **Comparison between the proposed power model and measured data.**

Moreover, when load falls down to zero, switch-off DU and RU can be assumed. In this case, the total power is assumed to reduce to zero in the paper, although in practice there can be a small amount of energy consumption. Hence, the total power can be expressed as

$$\mathrm{P}^{\mathrm{Total}}(l) = \begin{cases} 0 & \text{if } l = 0 \\ \mathrm{P}^{\mathrm{DU}}(l) + \mathrm{P}^{\mathrm{RU}}(l) & \text{if } l > 0. \end{cases} \tag{3}$$

The power saving comes from two sources. First, each cell adapts its load according to current network traffic. Second, certain low load cells need to handover their traffic to other cells such that these low load cells can be completely switched off.

C. NETWORK TRAFFIC MODEL

Network traffic model in this paper was developed based on measured NTV data published in [11]. The measured NTV data were extracted by visual inspection. It can be observed in [11] that the shape of NTV in each single day is similar. However, the absolute NTV values in weekdays and weekends are different. Therefore, to establish the model, a two-step approach is used in this paper. First, a normalized NTV model for a single day is established, to characterize how NTV is varying in different hours of a day. Second, another model is established to characterize how NTV is varying in different days of a week.

The NTV model $V_1(t)$ for a single day can be expressed as a 20th-order polynomial as

$$V_1(t) = \sum_{n=0}^{20} a_n t^n, \tag{4}$$

where $t \in [0, 24)$ is the hour of a day and a_n is the coefficient of the nth-order term. Least-squared estimation was performed and the coefficients a_n can be found in Table 1.

Fig. 3 shows the comparison of normalized measured NTV in a day in [11]. It can be seen that the valley of NTV appears at approximately 4 am, which accounts for 15% of the peak of a day. The peak of NTV occurs at 9 pm. The proposed 20th-order polynomial model provides sufficient approximation to the real-world one-day measured NTV.

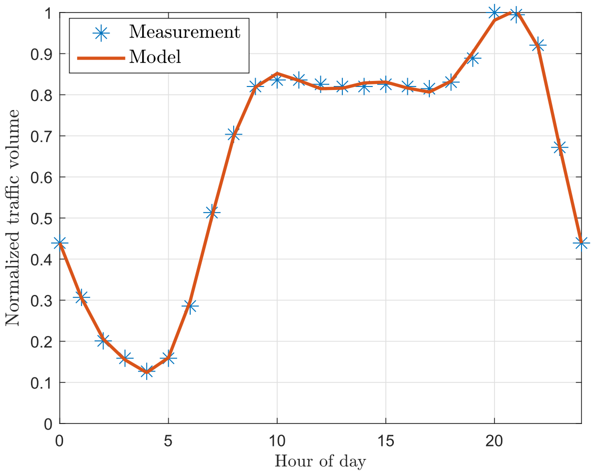


FIGURE 3. **Comparison between the single-day NTV model and measured NTV in [11].**

Next, to capture the variation of days within a week, the NTV model $V_2(\tau)$ for a week can be expressed as a 5th-order polynomial as

$$V_2(\tau) = \sum_{m=0}^{5} b_m \tau^m, \tag{5}$$

where $\tau = 1, 2, \ldots, 7$ represents Monday to Sunday and b_m is the coefficient of the mth-order term. The coefficients b_m are in Table 2. Then, with (4) and (5), the NTV of a week can be synthesized via

$$V(t) = \eta V_1(\mathrm{mod}(t, 24)) V_2((\mathrm{mod} \lfloor t/24 \rfloor, 7) + 1) \tag{6}$$

where t is the hour in a week ($t \in [0, 168)$), $\mathrm{mod}(\cdot)$ is the modulus operator, $\lfloor \cdot \rfloor$ is the flooring operator, and η is a

TABLE 1. Polynomial coefficients for NTV model in a day.

n	0	1	2	3	4	5	6
a_n	4.40×10^{-1}	-3.40×10^{-3}	-3.30×10^{-1}	3.00×10^{-1}	-1.30×10^{-1}	3.00×10^{-2}	-4.50×10^{-3}
n	7	8	9	10	11	12	13
a_n	3.72×10^{-4}	-1.62×10^{-5}	1.73×10^{-7}	1.22×10^{-8}	-2.97×10^{-10}	3.07×10^{-12}	-1.09×10^{-12}
n	14	15	16	17	18	19	20
a_n	6.98×10^{-14}	-3.42×10^{-15}	2.74×10^{-16}	-1.59×10^{-17}	5.03×10^{-19}	-8.15×10^{-21}	5.40×10^{-23}

TABLE 2. Polynomial coefficients for NTV model in a week.

m	0	1	2	3	4	5
b_m	4.42×10^{-1}	9.67×10^{-1}	-6.08×10^{-1}	1.72×10^{-1}	-2.24×10^{-2}	1.10×10^{-3}

scaling factor to scale the normalized NTV to a realistic NTV. The normalized NTV model is depicted in Fig. 4. It can be observed that during the weekdays, the NTVs are similar. However, during weekend, the NTVs drop from Saturday to Sunday.

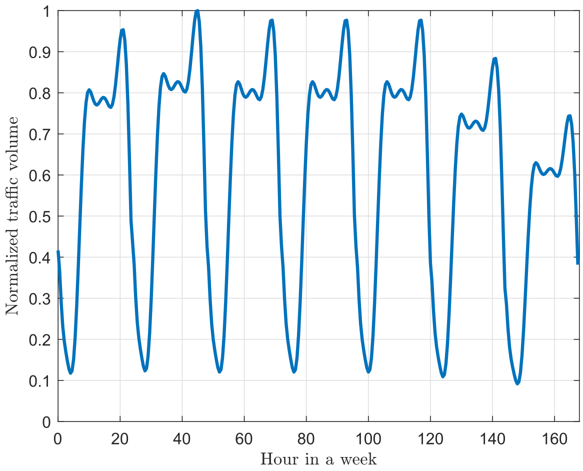


FIGURE 4. Normalized synthesized NTV model in a week.

Furthermore, we define a parameter γ called safety margin, which quantifies the largest rate of change of NTV between two adjacent time instances t_ν and $t_{\nu+1}$, i.e.,

$$\gamma \equiv \max_{\nu} \frac{V(t_{\nu+1}) - V(t_\nu)}{V(t_\nu)} \tag{7}$$

where ν is the sub-interval index when a certain length of observation interval is divided into equal-size sub-intervals.

A network with γ taken into consideration will be able to satisfy the period when the traffic increases at the steepest rate. From (6), it can be computed numerically that γ equals 0.4.

Assume that there are N_{user} users per cell with index $i = 1, 2, \cdots, N_{\text{user}}$, the user traffic volume (UTV) for the ith user in terms of time is modeled as

$$\text{UTV}_i(t) = \frac{V(t)}{N_{\text{user}}} \cdot Z_i \tag{8}$$

where Z_i is user-specific independent and identically distributed (i.i.d.) log-normal random variable, i.e., $\ln Z_i \sim \mathcal{N}(0, \sigma^2)\ \forall i$, to describe user-specific traffic variations. Parameter σ is the standard deviation of UTV among different users.

III. PROBLEM DESCRIPTION AND SYSTEM MODEL

A. PROBLEM DESCRIPTION

From Section II, the mobile network power saving problem is to adjust loads of cells according to the current NTV requirement. Furthermore, to save the largest amount of power, a handover mechanism needs to be considered such that certain cells can migrate its attached users to other cells and reduce its load to zero and switch off. However, since the number of cells can be massive in the area of interest (AOI), the solution space of this combinatorial optimization problem will be too large for exhaustive search, even for a single time instance. Moreover, the NTV is evolving in terms of time. The solution of the problem should be sufficiently flexible to handle the variation of NTV.

B. NETWORK DEPLOYMENT

In this paper, we consider an approximately 1km × 1km AOI which is covered by four frequency bands. Among these four frequency bands, three of them are for urban micro (UMi) and one of them is for urban macro (UMa). Settings of these four frequency bands are listed in Table 3. The UMi cells have carrier frequencies 2.1 GHz, 2.7 GHz, and 3.6 GHz, and they are two-ring hexagonal [25] and with 200m inter-site distance (ISD). For a two-ring hexagonal layout [25], each band has 19 three-sector sites, resulting in 57 cells in total. The UMa cell has carrier frequency 1.8 GHz and it is one-ring hexagonal with 500m ISD. For a one-ring hexagonal layout [25], each band has 7 three-sector sites, resulting in 21 cells in total. Users are uniformly and randomly dropped into the AOI for each band and the average total NTV for each band of each cell, i.e., the mean of the sum of all user traffic within a cell, equals (6) for a specific time t. It should be noticed that each band will fully cover the AOI. When a user is dropped inside the AOI, it will choose the cell in a certain band which provides the largest received power. Also, in this paper, for the sake of reducing power, handover between different bands is allowed. Namely, when cells in two different bands with similar coverage area, one cell in

TABLE 3. Settings of the four frequency bands covering the area of interest.

Band ID	Cell type	f_c	ISD	No. of cells	Layout
1	UMi	2.1 GHz	200m	57	Two-ring hexagonal
2	UMi	2.7 GHz	200m	57	Two-ring hexagonal
3	UMi	3.6 GHz	200m	57	Two-ring hexagonal
4	UMa	1.8 GHz	500m	21	One-ring hexagonal

Band 1 can migrate all of its traffic to the other cell in Band 2, provided that Band 2 will not overload. Then, the cell in Band 1 will have zero load and it can completely switch off.

C. SINR AND NETWORK THROUGHPUT CALCULATION

Since different bands will not interfere each other, band index is dropped in the following expressions. Consider a certain band, let P_k be the transmit power of the kth cell and let $\beta_{ik}^{(k)}$ be the large scale path loss between the kth cell and the ith user in the AOI, where the superscript $\cdot^{(k)}$ denotes that the user belongs to the kth cell. Furthermore, let N_{PRB} denote the total number of PRBs, let l_j be the load of the jth cell and let N_0 and $N_i^{(k)}$ denote the noise density and number of assigned PRBs to the ith user in the kth cell, respectively. Then, assuming the bandwidth of a PRB is B, the signal-to-noise-plus-interference ratio (SINR) $\rho_i^{(k)}$ of the ith user in the kth cell in the AOI can be expressed as

$$\rho_i^{(k)} = \frac{\beta_{ik}^{(k)} l_i P_k}{N_0 N_i^{(k)} B + \sum_{j \neq k} \beta_{ij}^{(k)} P_j \chi_{ij}} \tag{9}$$

where χ_{ij} is a coefficient representing the interference ratio between the ith base station and the jth base station. As the jth base station is only using $N_j^{(k)}$ PRBs, the interference power emitted by it is a fraction of its total power P_j. At the same time, the ith base station has only $N_i^{(k)}$ active PRBs, the interference power it receives is a fraction of the interference power emitted by the jth base station. As a result, χ_{ij} is a function of N_{PRB}, $N_i^{(k)}$, and $N_j^{(k)}$. Also, the number of PRBs allocated by a base station is assumed to be randomly distributed in $[0, N_{\text{PRB}}]$. The average number of PRBs selected by both the ith and the jth cells can be computed as $\frac{N_i^{(k)} N_j^{(k)}}{N_{\text{PRB}}}$. Therefore, the interference ratio coefficient can be expressed as

$$\chi_{ij} = \frac{N_i^{(k)} N_j^{(k)}}{N_{\text{PRB}}^2}. \tag{10}$$

The network throughput $T^{(k)}$ provided by the kth cell can be computed as

$$T^{(k)} = \mu \sum_i N_i^{(k)} B \log_2(1 + \rho_i^{(k)}) \tag{11}$$

where μ represents a factor accounting the overhead and number of layers during the transmission process. The area throughput in the AOI T^{area} is then the sum of the network throughput of all cells, i.e.,

$$T^{\text{area}} = \sum_k T^{(k)}. \tag{12}$$

From (9) and (11), it can be observed that the network throughput may not always be monotonically increasing with loads, because as loads increase, mutual interference among cells increases as well. Also, when a set of new loads are configured for all the cells, the SINR and throughput map of the AOI need to be updated.

IV. BENCHMARK METHOD: THRESHOLD-BASED POWER SAVING

Controlling problems like MN power saving are usually approached by threshold-based methods. Namely, a feedback loop is established and the feedback is mapped to a metric, such that actions will be taken accordingly based on whether the metric is higher or lower than a threshold. For power saving, these actions include scaling up or down the loads of cells, and handing over traffic to other bands and switching off cells whose loads are zero, and switching on cells. Let $V_X^{(k)}$, $T_X^{(k)}$, $l_X^{(k)}$ be the NTV, the network throughput, and the cell load of the kth cell in Band X, respectively. The adaptation of cell load $l_X^{(k)}$ at time t_{n+1} is expressed as

$$l_X^{(k)}(t_{n+1}) = l_X^{(k)}(t_n)(1+\gamma)\frac{V_X^{(k)}(t_n)}{T_X^{(k)}(t_n)} + \frac{V_Y^{(k)}(t_n)}{T_X^{(k)}(t_n)} \tag{13}$$

where γ from (7) is a safety margin such that the cell load is enough for the steepest NTV increase. It can be seen from (13) that the cell load at time t_{n+1} is the sum of two terms. The first term is a scaled version of load at the previous time t_n. The gap between two time instances is customizable and it is assumed half an hour in this paper. The second term is an additional load if Band Y is switched off and Band Y migrates its traffic to Band X.

When the load of the kth cell $l_X^{(k)}(t_{n+1})$ in Band X is less than a threshold ξ_1, i.e., $l_X^{(k)}(t_{n+1}) < \xi_1$ the cell will handover its traffic to another band then the cell can be switched off and $l_X^{(k)}(t_{n+1})$ will be set to zero. For simplicity, we assume that the handover is done by handing over from Band X to Band $X+1$. This is a feasible simplification if the traffic distributions in Band X and Band $X+1$ are statistically the same. On the other hand, let $\bar{l}_X^{\text{active}}$ denote the average load of active cells in Band X. If $\bar{l}_X^{\text{active}} > \xi_2$, meaning that current active cells have heavy load, then inactive cells in Band $X-1$ should be switched on to help handle traffic. The settings

TABLE 4. Parameter settings in the threshold-based power saving method.

Parameter	Description	Value
γ	Safety margin for NTV sudden jumps	0.4
ξ_1	Threshold for a cell to start traffic handover and switching off process	0.2
ξ_2	Mean load of active cells in a band to switch on inactive cells	0.8

of γ, ξ_1, and ξ_2 in this paper are heuristically determined and listed in Table 4.

The procedure of the threshold-based power saving method is shown in Fig. 5. The procedure starts with scaling the cell load only based on traffic variation. Then, each band starts to adjust the on and off situations. This is achieved by calculating the average load of active cells. If this load is larger than ξ_2, it requires more cells to offload upcoming traffic. Therefore, inactive cells in Band $X-1$ are switched on. If this load is less than ξ_2, the load of each cell in Band X will be compared to ξ_1. If a cell load is less than ξ_1, it means this cell has low load and can be switched off as soon as its traffic is handed over to Band $X+1$. Otherwise, the updated load is calculated according to (13).

```
Update cell loads with scaled version of the load at the previous time $t_n$, $l_X^{(k)}(t_{n+1}) = l_X^{(k)}(t_n)(1+\gamma)\frac{V_X^{(k)}(t_n)}{T_X^{(k)}(t_n)}$ $\forall X, k$;
for Band ID $X = 1, 2, 3, \ldots$ do
    Calculate the average load $\bar{l}_X^{\text{active}}$ of active cells in Band $X$;
    if $\bar{l}_X^{\text{active}} > \xi_2$ then
        Switch on inactive cells in Band $X-1$;
    else
        for each cell $k$ in the band do
            if $l_X^{(k)}(t_{n+1}) < \xi_1$ then
                Handover traffic to the $k$th cell in Band $X+1$;
                Switch off the $k$th cell in Band $X$;
            else
                Calculate updated load based on (13);
            end if
        end for
    end if
end for
```

FIGURE 5. Pseudo codes of threshold-based power saving.

V. DCNN-Q FOR MN POWER SAVING

A. RL REVIEW

RL is a trial-and-error machine learning technique, which samples the environment and takes actions to the environment. The environment is everything that cannot be arbitrarily modified by the RL agent and will provide a feedback containing the reward corresponding to the action to the RL agent. When the RL agent obtains a sample from the environment, this sample is known as a state. The RL agent attempts to make a sequence of decision on actions in order to achieve a certain goal. The difficulty of RL is that when an action is taken in each step, it will impact on actions in later stages.

A Markov decision process (MDP) provides widely used model for RL [26]. A MDP can be modeled by a 4-tuple $E = E\langle S, A, P, R\rangle$. State space S consists of all possible states of the environment. A state $s \in S$ is the perception of the environment of the RL agent. Action space A contains potential actions to be taken by the RL agent. Assume that the state is s, when action $a \in A$ is taken, the environment will transit to a new state s'. This transition is modeled by a hidden transfer function $P : S \times A \times S \mapsto \mathbb{R}$, which represents the transition probability. Moreover, in each state transition, a reward is produced and it is characterized by $R : S \times A \times S \mapsto \mathbb{R}$.

A policy π associates a state s to an action, which can be categorized as deterministic or randomized. A deterministic policy maps a state to an action $\pi : S \mapsto A$. On the contrary, a randomized policy maps a state to a probability distribution $\pi : S \times A \mapsto \mathbb{R}$, representing the probability of taking action $a \in A$ in state s. In the learning process, the state-action value function (Q function) $Q^\pi(s, a)$ stores the estimated values of accumulated discounted rewards using policy π.

When the model of the environment is accessible (model-based learning), i.e., the hidden transfer function P is known, the expected values of the Q function can be computed iteratively with dynamic programming. According to [26], the optimal policy satisfies the optimal Bellman equation and can be found by selecting the action maximizing the state-action value iteratively. The state-action values increase monotonically each time the policy is updated with the best action. Therefore, when the policy converges, it converges to the optimal policy.

However, in practice, it is usually difficult to obtain the model of the environment. Namely, the hidden transfer function P is unknown. In this case, model-free learning can be applied. Model-free learning assumes no knowledge of the environment and relies on approximating the Q function by sampling the environment, states, and rewards. A widely used model-free learning method is the Monte Carlo (MC) method [26], where the value function and policies are updated only when an episode of samples are finished. Another model-free learning method is the temporal-difference (TD) learning [26] where the value function and policies are updated in a step-by-step manner. A TD learning method known as Q-learning is used in this paper. The main characteristic of Q-learning is that the approximation of the Q function is independent of the policy being followed, which largely reduces complexity [26].

B. DCNN-Q ARCHITECTURE DESIGN

The overview diagram of DCNN-Q for mobile network power saving is depicted in Fig. 6. This RL problem can be divided into the design of state space, action space, policy, reward function, and Q function, which will be detailed later paragraphs.

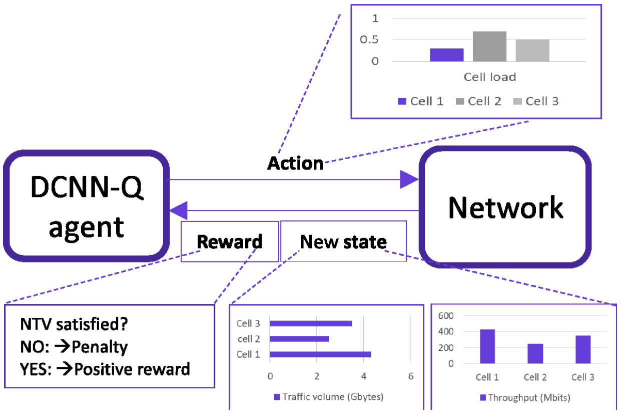


FIGURE 6. Diagram of DCNN-Q for mobile network power saving.

1) STATE SPACE

In a power saving problem, a state should be able to capture what the current requirement of NTV is and how well the system is responding to such requirement. Therefore, a state $s \in S$ is characterized by a traffic map, a throughput map, and a vector of current loads of all cells in the AOI, i.e., $s = \{\text{traffic map, throughput map, loads of all cells}\}$. It should be noticed that both maps are 2D while the vector of loads of all cells is 1D.

2) ACTION SPACE

The cardinality of the action space should be properly design. If the cardinality of the action space is too small, then the granularity of actions becomes coarse. Conversely, if the cardinality of the action space is too large, convergence of training will be too slow. To reach this balance, three constraints are taken in this paper. First, instead of continuous load, only discretized loads are considered. For example, a load can only be chosen in the set of $\{0\%, 25\%, 50\%, 75\%, 100\%\}$. However, even only discretized loads are considered, with 192 cells as shown in Table 3, there are still 5^{192} combinations. Therefore, the second constraint is that once a load is chosen, all the cells in the same band will be set to the same load. Then, for four bands, the number of combinations reduces to $5^4 = 625$. Third, certain combinations will be excluded in the action space as the pseudo codes shown in Fig. 7. In Fig. 7, the subscript k in w_k represents the band identification (ID). From the action space generation algorithm, it can be observed that the UMa band is always on to guarantee there will be coverage in the AOI while UMi cells can be switched off. This constrain is able to avoid coverage holes in the AOI when certain UMi cells are switched off. After the algorithm in Fig. 7, the cardinality of the action space is reduced to 140.

```
$A = \{\}$;
**for** $w_4 = 1, 2, 3, 4$ **do**
    **for** $w_1 = 0, 1, 2, 3, 4$ **do**
        **for** $w_2 = w_1, \cdots, 4$ **do**
            **for** $w_3 = w_2, \cdots, 4$ **do**
                $\mathbf{v} = 25\% \times \{w_1, w_2, w_3, w_4\}$;
                Push $\mathbf{v}$ into $A$;
            **end for**
        **end for**
    **end for**
**end for**
```

FIGURE 7. Pseudo codes of action space generation.

3) POLICY

A state is mapped to an action via policy $\pi(s)$. A commonly chosen policy is the ϵ-greedy algorithm ($\epsilon \in [0, 1]$) [26]. It consists of two phases. In the exploitation phase, which has probability $1 - \epsilon$, the RL agent selects the action with the highest Q value. In the exploration phase, which has probability ϵ, the RL agent will choose an action in a random manner in the action space with equal probability. The ϵ-greedy algorithm is presented as

$$\pi^{\epsilon}(s) = \begin{cases} \pi(s) & \text{if } U < 1 - \epsilon \\ a \in A \text{ uniformly} & \text{otherwise} \end{cases} \tag{14}$$

where U is a uniform random variable in [0, 1].

4) REWARD FUNCTION

There are two principles to design the reward function. First, it should be penalized if the current network throughput is not able to satisfy the NTV requirement. With such design, the RL agent will learn from experience to avoid corresponding actions. Second, as another goal of the problem is to save power, the reward should be monotonically increasing if the network consumes less power, provided that the required NTV is satisfied. As a result, the reward function is modeled as

$$r = \begin{cases} -20 & \text{throughput} < \text{NTV} \\ \sum\limits_{X,k} \exp\left(-\beta P_X^{(k),\text{Total}}\right) & \text{otherwise} \end{cases} \tag{15}$$

where β is a positive coefficient describing how fast the reward is decaying with the increase in power and $P_X^{(k),\text{Total}}$ is the total power of the kth cell in Band X. The choice of the exponential function is because it is continuous in its domain and able to handle interpolated power values. In this paper,

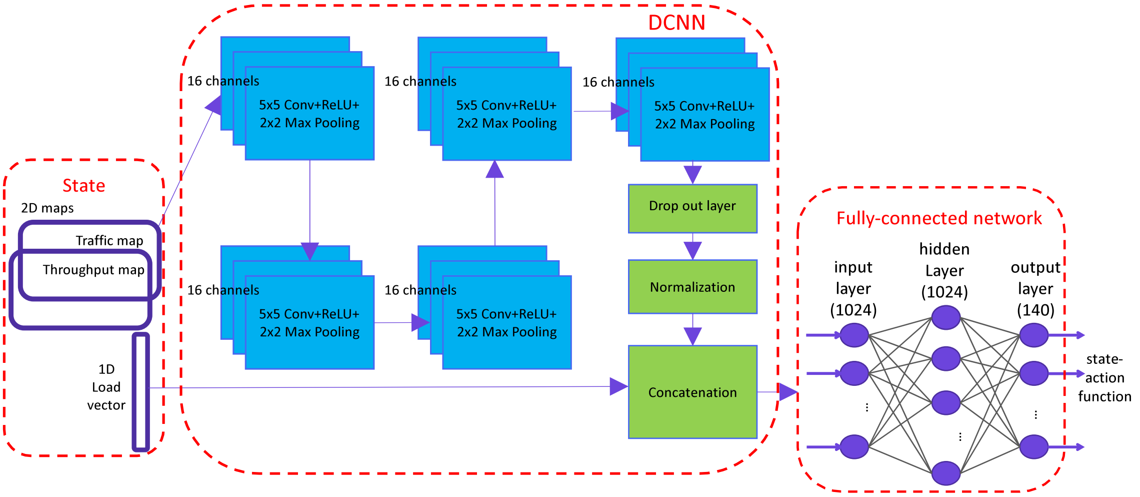


FIGURE 8. DCNN structure for Q function approximation.

β is set to be 0.004, which corresponds to a reward of 0.2 when the load is 25%.

5) Q FUNCTION

The accumulated reward of a state-action pair is recorded by the Q function $Q(s, a)$ and it is incrementally updated as the training progresses, i.e.,

$$Q(s, a) = Q(s, a) + \alpha(r + \delta Q(s', a') - Q(s, a)) \qquad (16)$$

where s' is the new state, a' is the action to the new state, α is the learning rate, and δ is the discounting factor. To approximate the Q function, both table-based and NN-based methods were used in the literature.

In this paper, the NN-based method is adopted. A NN-based Q function has two benefits. First, it does not need massive storage compared to a table-based method when the action space and state space are large. Second, it can handle complex inputs such as a mixture of 2D and 1D data and unseen states. A NN structure is proposed for approximation of the Q function. The NN accepts a state as input and outputs the Q value of each potential action. The NN is constructed by two parts which is shown in Fig. 8. The first part is a DCNN and the second part is a 3-layer fully-connected network. The DCNN maps a two-channel 2D image to a vector. One channel of the 2D image is the throughput map of the AOI and the other channel is the traffic map of the AOI. The two-channel image is then passed to five convolution blocks in serial and each convolution block consists of a convolution layer, a rectifier (ReLU) [27], and a pooling layer. The convolution layer is responsible for exacting high-level features of the 2D input. The ReLU is a typical non-linear activation in NNs. The pooling layer is responsible for reducing complexity and extracting dominate features. Then, after five convolution blocks, the output is passed into a drop-out layer to further reduce complexity and avoid overfitting. Since a state consists not only the 2D image but also a 1D vector storing the current loads of all the cells, the output of the drop-out layer is normalized and concatenated with the 1D load vector. The concatenated 1D vector is input to the second part of the NN, i.e., the 3-layer fully-connected network, the output of which is the the Q function. The objective function of the DCNN is the root mean squared error (RMSE) between the predicted Q value vector and the updated Q value vector.

To achieve the best performance of the DCNN, both the throughput map and the traffic map will be normalized before being input to the DCNN. Let 2D matrices $\mathbf{M}_1$ and $\mathbf{M}_2$ denote the throughput map and traffic map, respectively. The normalization of $\mathbf{M}_1$ includes cutting-off and scaling, i.e.,

$$\tilde{\mathbf{M}}_1 = \max\left\{\mathbf{M}_1, 20\right\}/20. \qquad (17)$$

The normalization of $\mathbf{M}_2$ is achieved by

$$\tilde{\mathbf{M}}_2 = \mathbf{M}_2/\eta. \qquad (18)$$

The DCNN-Q learning is described in pseudo codes in Fig. 9. To begin with, the policy is initialized with equal probability. In each iteration of the training, the RL agent chooses an action according to ϵ-greedy policy. As soon as the action is determined, it will be mapped to cell loads in all the bands. Then, all the cells will adjust their loads according to the action. After these processes, the situation of mutual inference is changed and hence SINR in the AOI needs to be re-calculated. Then, the throughput map needs to be updated and the new state is formed. Next, the reward is computed according to (15) and the next action is obtained from the policy function. The RL agent updates the Q function and the policy. These steps are then repeated until the maximum number of iteration is reached.

VI. RESULTS AND ANALYSIS

The proposed DCNN-Q power saving is trained and tested according to the parameters listed in Table 5. As comparisons,

```
$Q(s,a) = 0, \forall s, a; \pi(s,a) = \frac{1}{|A(s)|}, \forall a;$
for $t = 1, 2, 3, \dots$ do
    The RL agent chooses action $a$ according to $\pi^{\epsilon}(s)$;
    Cells in each band adjust their loads according to $a$;
    User handover and cell on/off if any;
    Re-calculate SINR in each band according to new loads;
    Re-calculate throughput;
    Form new state $s'$ with new throughput and traffic maps and new cell load vector;
    Calculate reward $r$ and compute $a' = \pi(s')$;
    The RL agent updates the state-action function
        $Q(s,a) = Q(s,a) + \alpha\left(r + \gamma Q(s',a') - Q(s,a)\right);$
    The RL agent updates policy $\pi(s) = \underset{u}{\operatorname{argmax}}\, Q(s,u)$;
    $s = s', a = a'$;
end for
```

FIGURE 9. Pseudo codes of DCNN-Q learning.

TABLE 5. Simulation parameter settings.

Parameter	Value
α	0.8
δ	0.5
ϵ	0.1
DCNN Optimizer	AdamOptimizer [28]
N_{user}	100
η	4×10^{11} bits
σ	0.97
Channel models	UMa, UMi [25]
Max. num. of PRBs	100
B	180 kHz
μ	2.8 (4 layers, 30% overhead)
N_{user}	100

the always-full-load method, the threshold-based method, and the DCNN-Q method are discussed. The always-full-load method means that all cells at all bands are operating with 100% loads.

Normalized mean reward with respect to the number of weeks trained is shown in Fig. 10. Normalization is down in terms of the mean reward after 10 weeks of training. It can be seen that there are fluctuations between 20 to 40 weeks, as the size of training data is still small. After 40 weeks of training, performance starts to improve. After 200 weeks of training, the result is 13% better than 10 weeks of training. As the length of training needs to reach balance between performance and training cost, we use the 200-week trained RL agent to test the proposed DCNN-Q performance.

NTV requirement and throughput provided by these three methods in terms of time within a week are illustrated in Fig. 11. The step size is half an hour. The always-full-load method provides a constant network throughput, which is 30% higher than the highest NTV peak during a week. This is the foundation for an intelligent power saving method. The threshold-based method is able to adjust its network throughput according to NTV. It can be seen that the threshold-based method is aggressive when the required NTV is low and is conservative when the required NTV is high. On the contrary, DCNN-Q does not behave like the threshold-based method. DCNN-Q is more conservative when NTV is low by reserving a larger safety margin, and more aggressive when NTV is high. It can also be observed that the change of configuration in DCNN-Q is sharper. This is because the action space of the DCNN-Q is discrete.

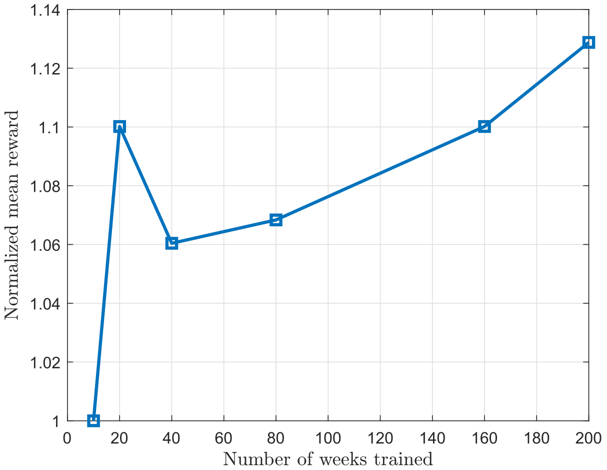


FIGURE 10. Normalized mean reward with respect to the number of weeks trained.

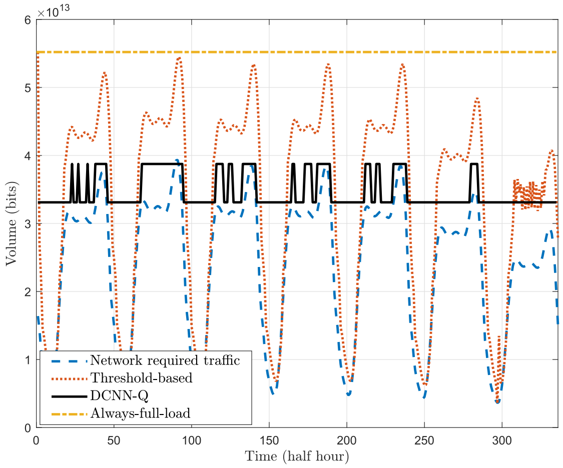


FIGURE 11. Network throughput comparison of always-full-load, threshold-based, and DCNN-Q methods in terms of time.

Network power consumption in terms of time within a week is illustrated in Fig. 12. The power consumption of the always-full-load method is constant. The threshold-based method has the lowest power consumption when NTV is low and had higher power consumption when NTV is high. Its range of power consumption is relatively wide.

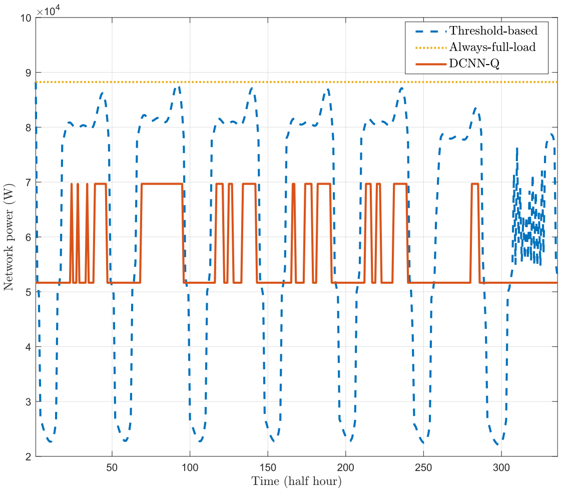


FIGURE 12. Network power consumption comparison of always-full-load, threshold-based, and DCNN-Q methods in terms of time.

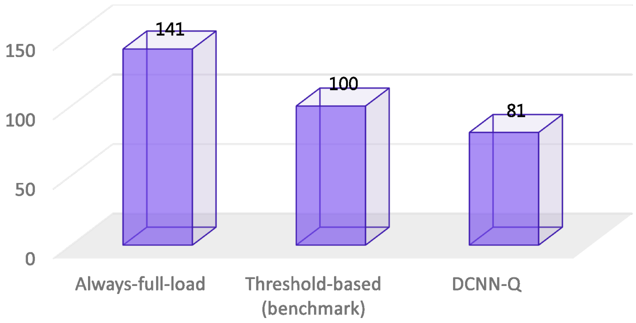


FIGURE 13. Normalized aggregate network power consumption comparison of always-full-load, threshold-based, and DCNN-Q methods.

Conversely, the provided network throughput by the DCNN-Q has limited range of power consumption values.

Fig. 13 depicts the normalized aggregate power consumption of the three methods. Normalization is done relative to the threshold-based method. Always-full-load consumes the most power as expected, which is 41% higher than the threshold-based method. The proposed DCNN-Q method is able to save 19% power compared to the threshold-based method and 42% compared to the always-full-load method. This demonstrates that the proposed method is able to achieve significant power saving.

VII. CONCLUSION

To investigate power saving for mobile networks, it is important to establish practical power and network traffic models. Based on our in-house measurement, linear models are sufficiently accurate to describe base station power consumption in terms of load. The power of RU is more sensitive to load change, whereas the power of DU is steady. Power reduction is achieved via the adaptation of loads of the network and dynamic switching on and off according to required NTV. A polynomial model for synthesizing NTV is proposed, describing traffic fluctuations over one week. The threshold-based method, which relies on heuristically set thresholds, serves as the benchmark and is able to reduce power consumption by 30% compared to always-full load. As a significant enhancement, the centralized DCNN-Q method is proposed. The DCNN-Q uses a DCNN, which accepts a joint input of 2D images and a 1D vector, to approximate the Q function in the Q-learning framework. The proposed DCNN-Q method is capable of saving 19% power compared to the threshold-based method. This demonstrates that DCNN-Q is a promising solution to confine mobile network power when both the data and the size of a network are soaring. For future work, instead of the centralized method proposed in this paper, a distributed learning framework would be another direction of research. Also, optimization on energy efficiency, i.e., bits/joule, is essential to consider for green network research.

REFERENCES

[1] Ericsson. White paper. (Nov. 2019). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/4acd7e/assets/local/mobility-report/documents/2019/emr-november-2019.pdf

[2] Samsung. White Paper. *5G Vision*. [Online]. Available: http://www.samsung.com/global/business-images/insights/2015/Samsung-5G-Vision-0.pdf

[3] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, "What will 5G be?" *IEEE J. Sel. Areas Commun.*, vol. 32, no. 6, pp. 1065–1082, Jun. 2014.

[4] *Study on New Radio Access Technology Physical Layer Aspects*, document 3GPP T.R. 38.802 V14.2.0, Sep. 2019.

[5] M. Dohler, R. W. Heath, A. Lozano, C. B. Papadias, and R. A. Valenzuela, "Is the PHY layer dead?" *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 159–165, Apr. 2011.

[6] M. Ding, D. Lopez-Perez, H. Claussen, and M. A. Kaafar, "On the fundamental characteristics of ultra-dense small cell networks," *IEEE Netw.*, vol. 32, no. 3, pp. 92–100, May 2018.

[7] Vodafone. White Paper. (2019). *Sustainable Business Report*. [Online]. Available: https://www.vodafone.com/content/dam/vodcom/sustainability/pdfs/sustainablebusiness2019.pdf

[8] J. D. Gadze, S. B. Aboagye, and K. A. Agyekum, "Real time traffic base station power consumption model for Telcos in Ghana," *Int. J. Comput. Sci. Telecommun.*, vol. 7, no. 5, pp. 6–13, Jul. 2016.

[9] C. Desset, B. Debaillie, V. Giannini, A. Fehske, G. Auer, H. Holtkamp, W. Wajda, D. Sabella, F. Richter, M. J. Gonzalez, H. Klessig, I. Gódor, M. Olsson, M. A. Imran, A. Ambrosy, and O. Blume, "Flexible power modeling of LTE base stations," in *Proc. WCNC*, Shanghai, China, Apr. 2012, pp. 1–5.

[10] E. Oh, K. Son, and B. Krishnamachari, "Dynamic base station switching-on/off strategies for green cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 5, pp. 2126–2136, May 2013.

[11] Alcatel-Lucent. White Paper. *Alcatel-Lucent 9900 Wireless Network Guardian*. Accessed: May 1, 2020. [Online]. Available: https://www.tmcnet.com/tmc/whitepapers/documents/whitepapers/2013/7451-alcatel-lucent-9900-wireless-network-guardian-powerful-mobile.pdf

[12] J. Wu, Y. Zhang, M. Zukerman, and E. K.-N. Yung, "Energy-efficient base-stations sleep-mode techniques in green cellular networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 803–826, 2nd Quart., 2015.

[13] K. Son, H. Kim, Y. Yi, and B. Krishnamachari, "Base station operation and user association mechanisms for energy-delay tradeoffs in green cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 8, pp. 1525–1536, Sep. 2011.

[14] S. Bhaumik, G. Narlikar, S. Chattopadhyay, and S. Kanugovi, "Breathe to stay cool: Adjusting cell sizes to reduce energy consumption," in *Proc. 1st ACM SIGCOMM Workshop Green Netw.*, New Delhi, India, Sep. 2010, pp. 41–46.

[15] R. Balasubramaniam, S. Nagaraj, M. Sarkar, C. Paolini, and P. Khaitan, "Cell zooming for power efficient base station operation," in *Proc. 9th Int. Wireless Commun. Mobile Comput. Conf. (IWCMC)*, Sardinia, Italy, Jul. 2013, pp. 556–560.

[16] A. Mosavi and A. Bahmani, "Energy consumption prediction using machine learning: A review," Mar. 2019, doi: 10.20944/preprints201903.0131.v1.

[17] S.-R. Cho and W. Choi, "Energy-efficient repulsive cell activation for heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 5, pp. 870–882, May 2013.

[18] *Physical Channels and Modulation*, document 3GPP T.R. 38.211, V15.6.0, Jun. 2019.

[19] *Physical Channels and Modulation*, document 3GPP T.R. 36.211, V14.0.0, Jun. 2016.

[20] Ericsson. (Feb. 2020). *Can AI Bring Down Network Energy Costs?* [Online]. Available: https://www.ericsson.com/en/blog/2020/2/ai-network-management-energy-costs

[21] Huawei. White Paper. (2019). *Enable Autonomous Driving Network*. [Online]. Available: https://carrier.huawei.com/~/media/CNBGV2/download/adn/Huawei-NAIE-White-Paper.pdf

[22] A. El-Amine, H. A. Haj Hassan, M. Iturralde, and L. Nuaymi, "Location-aware sleep strategy for energy-delay tradeoffs in 5G with reinforcement learning," in *Proc. IEEE 30th Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Istanbul, Turkey, Sep. 2019, pp. 1–6.

[23] F. E. Salem, Z. Altman, A. Gati, T. Chahed, and E. Altman, "Reinforcement learning approach for advanced sleep modes management in 5G networks," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Chicago, IL, USA, Aug. 2018, pp. 1–5.

[24] L. Kraemer and B. Banerjee, "Multi-agent reinforcement learning as a rehearsal for decentralized planning," *Neurocomputing*, vol. 190, pp. 82–94, May 2016.

[25] *Study on 3D Channel Model for LTE*, document 3GPP T.R. 36.873, V12.2.0, Jun. 2015.

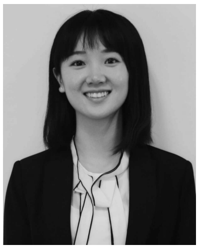
[26] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[27] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. Mach. Learn. Res.*, Fort Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[28] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, San Diego, CA, USA, May 2015, pp. 1–15.

SHANGBIN WU received the B.Sc. degree in communication engineering from South China Normal University, Guangzhou, China, in 2009, the M.Sc. degree in wireless communications (Hons.) from the University of Southampton, Southampton, U.K., in 2010, and the Ph.D. degree in electrical engineering from Heriot–Watt University, Edinburgh, U.K., in 2015. From 2010 to 2011, he worked as an LTE Research and Development Engineer responsible for LTE standardization and system-level simulation with New Postcom Equipment Ltd., Guangzhou. From October 2011 to August 2012, he was with Nokia Siemens Network, where he worked as an LTE Algorithm Specialist, mainly focusing on LTE radio resource management algorithm design and system-level simulations. He has been a 5G Researcher with the Samsung R&D Institute U.K., since November 2015.

YUE WANG (Senior Member, IEEE) received the Ph.D. degree from the University of Victoria. She is a Senior Technology Manager of the Samsung R&D Institute U.K. She is the Samsung Delegate of ETSI ISG Experiential Networked Intelligence (ENI), and the Secretary and Rapporteur of ENI. Prior to joining Samsung, she has worked in the U.S. and U.K. on various research roles in wireless communications. She has coauthored over 40 publications and is a named inventor of over 30 patents (and patent applications). Her current work focuses on AI for 5G networks and beyond, with topics span on the application of AI in communications systems and E2E networks. She is an Industry Advisory Board Member of the University of Sussex and King's College London, and is the Industry Supervisor of a five-year research program in the area of AI in 5G and beyond. She was awarded the ENI 2019 Award recognizing her extraordinary contributions to the ISG.



LU BAI received the B.Sc. degree in electronics information engineering from Qufu Normal University, China, in 2014, and the Ph.D. degree in information and communication engineering from Shandong University, China, in 2019. From 2017 to 2019, she was also a Visiting Ph.D. Student with Heriot–Watt University, U.K. She is a Research Fellow of Beihang University, China. Her research interests are (B)5G wireless communication channel measurements and modeling, including massive MIMO channel measurements and modeling, satellite-terrestrial integrated communication channel modeling, and machine learning-based channel modeling.