



HAL
open science

Neural Networks for Energy-Efficient Self Optimization of eNodeB Antenna Tilt in 5G Mobile Network Environments

Muhammad Nauman Qureshi, Muhammad Khalil Shahid, Moazzam Islam
Tiwana, Majed Haddad, Irfan Ahmed, Tarig Faisal

► **To cite this version:**

Muhammad Nauman Qureshi, Muhammad Khalil Shahid, Moazzam Islam Tiwana, Majed Haddad, Irfan Ahmed, et al.. Neural Networks for Energy-Efficient Self Optimization of eNodeB Antenna Tilt in 5G Mobile Network Environments. IEEE Access, 2022, 10, pp.61678 - 61694. 10.1109/access.2022.3181595 . hal-03846208

HAL Id: hal-03846208

<https://hal.science/hal-03846208>

Submitted on 10 Nov 2022



HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Received May 17, 2022, accepted June 5, 2022, date of publication June 9, 2022, date of current version June 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3181595

Neural Networks for Energy-Efficient Self Optimization of eNodeB Antenna Tilt in 5G Mobile Network Environments

MUHAMMAD NAUMAN QURESHI¹, MUHAMMAD KHALIL SHAHID¹², (Member, IEEE),
MOAZZAM ISLAM TIWANA¹, MAJED HADDAD³, IRFAN AHMED¹², (Senior Member, IEEE),
AND TARIG FAISAL²

¹Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 45550, Pakistan

²Department of Electrical Engineering, Higher Colleges of Technology, Abu Dhabi, United Arab Emirates

³Department of Electrical Engineering, University of Avignon, 84000 Avignon, France

Corresponding author: Muhammad Khalil Shahid (kshahid@hct.ac.ae)


This work was supported in part by the HCT Research Grant from the Higher Colleges of Technology (HCT), Abu Dhabi, United Arab Emirates, under Grant 2307.

ABSTRACT In this paper, we present an energy-efficient Self Organizing Network (SON) architecture based on a tunable eNodeB (eNB) antenna tilt design for macrocells in a mobile network environment. This is an imperative element of mobility management in high speed and low latency wireless networks. The SON architecture follows a fully distributed approach with optional network information exchange with neighboring cells and core network. Antenna tilt directly affects its radiation pattern thus changes in eNB antenna tilt can be used to optimize cell coverage and reduce interference in mobile networks. We apply and compare two reinforcement machine learning techniques for optimizing the eNB antenna tilts, i.e., Deep Q-learning using Artificial Neural Network (ANN) and a simple Stochastic Cellular Learning Automata (SCLA). ANN is well known for its ability to learn from a vast number of inputs, while the stochastic learning technique relies on a simple action based probability vector updated based on system feedback. Neighboring cells for any one cell in the network environment are selected based on their separation distance and antenna orientation. We validate the data call performance of the network for edge users as they directly impact the Quality of Service (QoS) in the mobile environment. Our simulated results show that ANN performs better for edge users as compared to SCLA. The model also satisfies the SON requirement of scalability and agility. This work is a follow-up to our earlier work, where we showed that SCLA performs better than Q-learning in a similar network environment and optimizing strategy due to its low complexity, but within the same Q-learning algorithm more input learning parameters gave better performance.

INDEX TERMS 5G, antenna tilt, artificial neural networks (ANN), deep Q-learning, energy efficiency, HetNet, self organizing networks, self optimization, stochastic cellular learning automata (SCLA).

I. INTRODUCTION

The challenge to keep up with high traffic demands and increasing requirements of High Definition (HD) multimedia services, has led the research community to find innovative technologies and ways to boost coverage, capacity, and energy efficiency (EE) of the 4th Generation (4G) and upcoming 5th Generation (5G) mobile networks [1]. These new ideas and innovations further add to the complexity of the network. Thus, finding some solution for managing the

The associate editor coordinating the review of this manuscript and approving it for publication was Nurul I. Sarkar¹.

limited available network resources becomes a formidable task. There exists a need to make the network intelligent enough so that it can autonomously learn and adapt to the varying network requirements and changing environment scenarios [2]. To address this requirement, 3GPP introduced the concept of SON in 4G Networks [3], and it is a core component of the 5G networks. The main functionality of SON comprises self-configuration, self-optimization, and self-healing in the cellular networks. SON presents an intelligent network that optimizes its parameters autonomously with the dynamics of the network such as a change in traffic demand, congestion, interference, and network entity breakdown.

Coverage adaptation is one primary use case of self-optimization in SON and remains an active research area for future mobile networks [2]. One way to achieve coverage adaptation is by intelligently changing tilt angles of eNB antennas according to the network environment requirements [2], [4], [5]. The tilt angle of an antenna has a direct bearing on its radiation pattern, and thus for an eNB, its antenna tilt can be effectively used to optimize its coverage and improve the overall network capacity [6].

A way to realize antenna tilt optimization is by using machine learning techniques. These techniques give the system the ability to learn from its environment and optimize its performance by intelligently tuning some defined parameters [2]. The upcoming 5G networks will have the ability and capacity to store real-time network information due to emerging technologies and ideas like mobile edge networks [7]. At any time, an eNB cell can get the latest information about its network environment from either the core network or its neighboring cells directly linked through the standard X2 inter-cell interface line [8]. Thus, optimization techniques that can take advantage of more input learning network information can give better performance compared to ones that can't.

In [9], a low complexity quality of experience driven antenna tilt optimization is proposed for the cellular network. The presented solution uses steepest descent algorithm for the clustering of the coverage area to minimize the mutual interference. In cellular network the capacity and coverage optimization is a complicated problem. There are diverse RF parameters and power control requirements for macrocell and small cells. Wang *et al.* [10] tackle the self-optimization problem for RF parameters and power control in heterogeneous networks using RL, with BS as an agent that learns the strategies to control the parameters.

Recent research work shows the trend toward self-optimization of the antenna tilt to enhance the EE using various artificial intelligence techniques. The challenging capacity and coverage optimization problem in a HetNet is investigated in [11]. This paper uses exploration-based reinforcement learning to adapt the antenna tilt for optimal coverage and per-user throughput. The proposed scheme increases the EE by at least 21% compared to the fixed strategies. Authors in [12] present a 3D electrical antenna tilt in mmWave massive MIMO homogeneous and heterogeneous networks. The EE is formulated as a function of antenna tilt. Due to the high complexity of the optimal solution, they obtained the low complexity near optimal solution. The optimal antenna tilt angle is calculated by the bisection algorithm. Parera *et al.* [13] apply transfer learning to train the machine learning model for the self optimization of antenna downtilt in the cellular networks. Extensive preprocessing has been done on the input data like data augmentation to improve the prediction performance. Transfer learning models reduce the training time but neural network design from scratch for a particular scenario/setup is advantageous and renders better performance. A reinforcement learning-based

antenna electrical downtilt has been presented in [14]. This RL-based scheme increases the signal-to-interference and noise ratio (SINR) for cell-edge users. But the presented solution is only for the sparse population in the suburban areas. In [15], the authors present a self-optimization and self-healing process model for the 5G network. The process model consists of optimization, precoding, and big data architecture. The given model only represents the abstraction level solution without the implementation details. A latest survey on the coverage enhancement [16] concludes that SON is becoming popular in the cellular mobile networks due to the inclusion of artificial intelligence, machine learning, deep learning, and reinforcement learning (RL) algorithms. SON for antenna tilt can be used for coverage improvement and to mitigate the coverage holes. In general, cellular network self-optimization procedures enhance network flexibility and scalability. In [17], the authors propose a two-step algorithm for jointly optimizing antenna tilt angle and vertical and horizontal half-power beamwidths of the macrocells in a heterogeneous cellular network. A multi-agent mean-field RL algorithm is first utilized in the offline phase to transfer features for the second (online) phase single-agent RL algorithm. The results show that the performance of proposed algorithm comes close to the multi-agent RL performance, with only hundreds of online trials. It performs much better than a single agent RL. Furthermore, the proposed algorithm empirically appears to provide a performance guarantee regardless of the extent of the environmental dynamics. Authors in [13] propose a transfer learning method based on Feed-Forward Neural Networks to predict the strength of the radio signal in a reference tilt configuration. It then transfers the acquired information to a new neural network in order to obtain the best predictions in the target tilt arrangement. It has been shown [18] that the network output balances the received signal strength and the interference intensity to achieve the maximum coverage probability for a given base station density using the best antenna down tilt. It can also significantly improve area spectral efficiency, explicitly regarding base station density. It can delay the area spectral efficiency crash by almost one order of magnitude. Analytical results showed that three components are determining the optimal antenna down tilt, i.e., LOS links, the NLOS links, and the noise.

In our earlier work [19], we showed that a simple RL technique like Stochastic Cellular Learning Automata (SCLA) performed better than a more complex Q-learning one. In RL, a learning agent self-learns from its environment without requiring explicit training data. SCLA can quickly adapt because it is based on a small probability vector with a dimension equal to the number of actions possible and gets updated based on the feedback from the network with each time step. On the other hand, Q-learning consists of a large input state-action matrix that gets updated for every state-action combination and the feedback from the network. Also, as the state in the Q-learning matrix is based on the number of input learning parameters, so the order of the Q Matrix grows exponentially with the increase in the number of inputs and actions

possible. Thus compared to the SCLA, Q-learning required more time to train its state-action matrix and adapt, resulting in poor performance. Moreover, we found that within the same Q-learning technique, if we increase the number of input learning parameters, we find some slight performance improvement. Thus a better alternative to matrix learning was required.

One specific class of machine learning techniques that can learn from a diverse number of inputs and benefit from an information-rich environment is ANN [20]. ANNs are particularly suited for pattern recognition and finding optimal solutions to complex relationships [20]–[22]. Used in combination with Q-learning as in deep Q-learning they promise to address the problem of the state-action matrix in basic Q-learning technique.

Motivated from the above-mentioned concerns in the current research work and the potential of our proposed dynamic self-optimization downtilt scheme, in this paper, we look into enhancing the network performance using coverage adaptation with electronically steerable eNB antennas and self-optimization with deep Q-learning in a homogeneous 4G/5G environment. A homogeneous cellular network consists of the planned deployment of base stations. All base stations have same transmit power. They serve roughly the same number of users, and all users have similar QoS requirements. In particular, we look at File Transfer Protocol (FTP) call performance for edge users, especially handovers. We also compare the results with SCLA to validate our earlier work.

The SON model we present in our work is fully distributed with an optional neighborhood interaction and is an enhancement to our earlier model presented in [19]. Each eNB has a 120° antenna and focuses on optimizing its own edge user data throughput and file transfer time performance. Additionally, an eNB can also interact with its neighbors to learn about their network situation and their selected tunable parameter patterns for further optimization. The selection of neighborhood cells is based on a simple distance separation criteria and antenna direction so that only those eNBs are considered that can have overlapping coverage. In the previous work, we did not consider antenna orientation as one criterion for selecting the neighborhood cells. This approach allows inter-cell information exchange without compromising the scalability of the network.

The recent distributed SON architectures most pertinent to our work have been proposed in [8] and [23]. The most relevant paper compared to our work is presented in [8]. The authors propose a fuzzy neural network optimization model based on RL in a distributed architecture. It has an option of sharing learning experiences from a central management server to control both the power and tilt of SON entities. The authors claim that the model meets self-optimization requirements in a dynamic model, but the application of fuzzy logic in the learning process weakens its application for complex environments. In [23] a distributed architecture with coordination and communication between the Base Stations (BS) is presented. RL algorithm is used in each BS to

optimize antenna tilt. The approach shows network performance improvements but suffers from two issues; (a) Exploration and exploitation steps of the learning process are done separately, with the exploration step in the order of hours, and (b) All BS in the network are not optimized simultaneously, but one after the other making it less scalable. Thus in the context of SON, the approach fails to achieve desired characteristics of scalability and agility.

Contributions

The main contributions of our work are summarized as under:

- This paper shows the feasibility of realizing a fully distributive and cooperative SON network architecture capable of adapting to complex mobile environments with rapidly changing network user scenarios and operator requirements, with desired SON characteristics of scalability, stability, and agility [2].
- Additionally, we also show that machine learning techniques like ANN that can assimilate more information from the environment can give better results compared to learning techniques that are limited in their learning capacity like SCLA.

The paper is organized as follows. Section II presents the network system model using tunable eNB antennas and our objective function. Section III presents the design of SON elements using either the ANN or SCLA learning technique. Section IV describes the KPIs used in the paper to evaluate the system performance for edge users. Section IV describes the network simulation design and results achieved based on the simulation. Section V concludes the paper.

TABLE 1. List of abbreviations.

Abbreviations	Description
3GPP	3rd Generation Partnership Project
ANN	Artificial Neural Network
AAS	Active Antenna Systems
BS	Base Station
CA	Cellular Automata
eNB	eNodeB
KPI	Key Performance Index
LTE	Long-Term Evolution
MDP	Markov Decision Process
NMS	Network Management Server
OFDMA	Orthogonal Frequency Division Multiple Access
PRB	Physical Resource Blocks
QoE	Quality of Experience
QoS	Quality of Service
RL	Reinforcement Learning
SCLA	Stochastic Cellular Learning Automata
SON	Self-Organizing Networks
UE	User Equipment

II. SYSTEM MODEL

A. DESIGN OVERVIEW

We consider a homogeneous mobile network in a fully urban environment with a hybrid SON architecture as shown in Fig. 1. Each eNB cell has a sectored antenna and acts as an independent learning entity. A group of three co-located eNBs covers the complete 360° for a particular location.

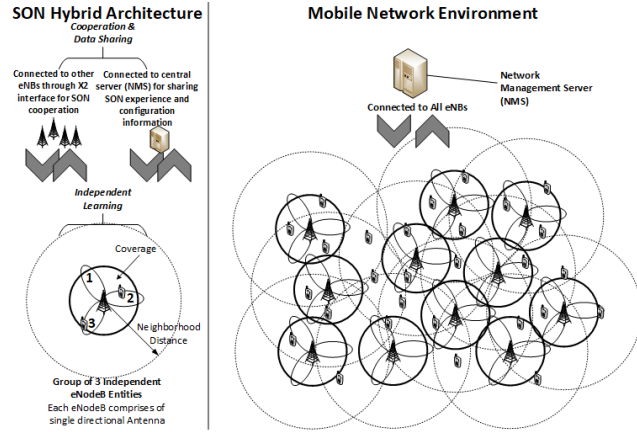


FIGURE 1. System model based on SON hybrid architecture.

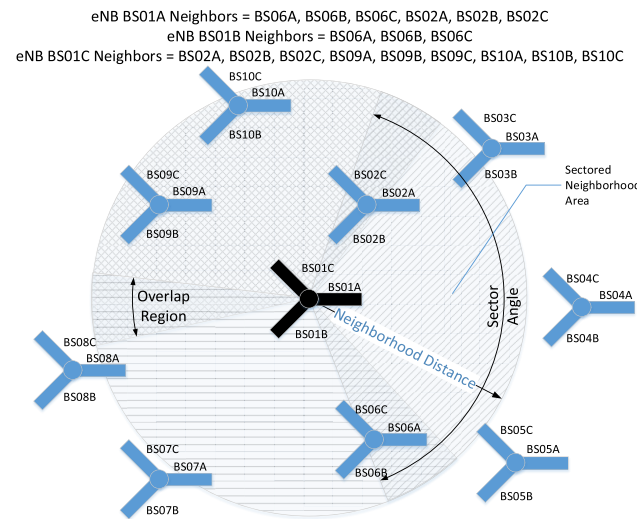


FIGURE 2. Neighborhood cell selection.

Any eNB can interact with other eNBs through the standard X2 interface. We assume a central Network Management Server (NMS) where all cells share their learning experiences, settings, configurations, and locations. Any eNB can find its interfering neighboring cells by requesting their positions and antenna orientation data from NMS. Once an eNB determines its interfering neighboring cells it can directly interact with them through the X2 interface, thus saving valuable time coordinating through the central NMS. Thus, a fully distributed SON architecture is realized with information sharing between cells.

B. NEIGHBORHOOD CELLS SELECTION

Neighborhood eNBs selection shown in Fig. 2, is based on fixed distance separation and eNBs antenna orientation. In our implemented system model, we consider eNBs neighbourhood separation distance of $d_{nsep} = 550$ m, and with sector coverage of $120 \pm 5^\circ$. The 550 m distance is selected to consider neighbours that are just beyond the minimum

inter-site cell distance set to $d_{minsep} = 500$ m, $d_{nsep} = 1.1 \times d_{minsep}$. The extra $\pm 5^\circ$ over 120° is kept to compensate for sectoral boundary coverage conditions between adjacent three eNBs at one point location (Fig. 1). Any eNB more than 550 m with the reference eNB will not cause interference to the users of the reference eNB. For example, consider the sector (or cell) A of BS01, i.e., BS01A in Fig. 2. The users in this cell experience interference from BS02A, BS02B, BS02C, and BS06A, BS06B, BS06C. BS03 will not cause any interference to BS01A because it is outside the neighbourhood distance $d_{nsep} = 550$ m.

ANTENNA TILT MODEL

Altering antenna tilt directly impacts the eNB cell coverage due to changes in the radiation pattern. In our design, we optimize the antenna beam for providing better coverage to edge users, in a way that the interference with neighboring cells is brought to a minimum. In this work, we have limited the changes to antenna tilt angle θ in fixed steps only.

The gain of an antenna at a fixed location is computed using antenna elevation angle ψ and azimuth angle ϕ (Refer Fig. 3). Elevation angle ψ is calculated from the antenna height and ground distance between eNB and that User Equipment (UE)¹ location point. Azimuth angle ϕ is calculated from the antenna direction and the UE location point coordinates. For a trisectorial site, 3GPP defines azimuth, elevation, and the total radiation patterns at location (ψ, ϕ) given respectively by [24]:

$$A_H(\phi) = -\min \left[12 \left(\frac{\phi}{\phi_{3dB}} \right)^2, A_m \right] \quad (1)$$

$$A_V(\psi) = -\min \left[12 \left(\frac{\psi - \theta}{\psi_{3dB}} \right)^2, SLA_v \right] \quad (2)$$

$$A(\phi, \psi) = -\min [-[A_H(\phi) + A_V(\psi)], A_m] \quad (3)$$

where;

A_m : backward attenuation factor in the horizontal plane and taken as 25 dB

SLA_v : backward attenuation factor in the vertical plane and taken as 20 dB

ϕ_{3dB} : half power azimuth beamwidth

ψ_{3dB} : half power elevation beamwidth

The total radiation pattern $A(\phi, \psi)$ is used to compute the eNB antenna gain $G = \zeta A$ for any location, where ζ is the ratio of transmit power at the input of the antenna and the transmit power of eNB.

C. INTERFERENCE MODEL

We consider OFDMA (Orthogonal Frequency Division Multiple Access) as access technology on the air interface. OFDMA subdivides the bandwidth into many subcarriers [25]. The smallest carrier bandwidth allocated to a user in unit time is in the form of Physical Resource Blocks (PRBs). Each PRB is exclusively assigned to a single user at a particular time, eliminating the intra-cell interference. Thus,

¹UE and user are used interchangeably

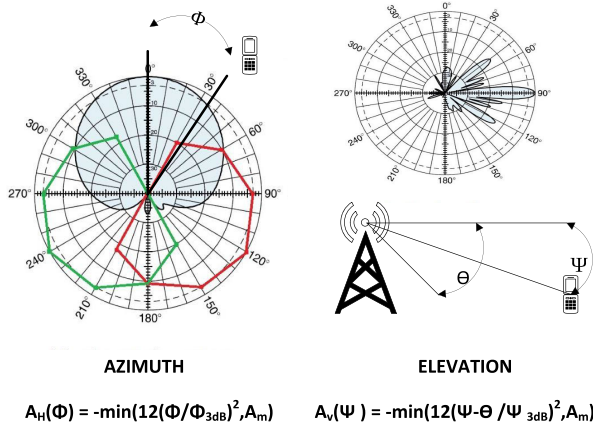


FIGURE 3. Antenna tilt model.

the interference suffered by a UE is the inter-cell interference only. Now, assuming UE u is attached to eNB e , the average interference I_{ue} observed by u per sub-carrier can be given as:

$$I_{ue} = \sum_{i=1, i \neq e}^k M(i, e) \times v_i \frac{P_i G_i}{\xi_{iu}} \quad (4)$$

where;

$M(i, e)$: 1 if eNB i and eNB e use the same frequency otherwise it is 0

v_i : ratio of allocated PRBs to total available PRBs in eNB i

P_i : transmit power of eNB i

G_i : gain of antenna for eNB i

ξ_{iu} : link loss between eNB i and UE u

Note that the link loss includes losses like path loss and fading. The signal to noise ratio observed by the UE u attached to eNB e or $SINR_{ue}$ can be given as follows, based on the interference received I_{ue} .

$$SINR_{ue} = \frac{P_e \times G_e}{\xi_{ue}(I_{ue} + \delta^2)}, \quad (5)$$

where;

P_e : transmit power of eNB e

G_e : gain of antenna for eNB e

ξ_{ue} : link loss between eNB e and UE u

δ^2 : thermal noise power per carrier

The throughput of user e attached to eNB u is given by

$$TH_{ue} = B_{ue} \times BW_{eff} \times \log_2 \left(1 + \frac{SINR_{ue}}{\Gamma} \right), \quad (6)$$

where;

B_{ue} = total bandwidth corresponding to the total number of PRBs assigned to user u by eNB e

BW_{eff} = Bandwidth Efficiency

Γ = SINR efficiency

The throughput of one eNB is given as $TH_e = \sum_{u=1}^N TH_{ue}$.

The utility function U for UE u served with a finite number of PRBs by eNB e can be given as [26]:

$$U_{ue} = \frac{B_{ue} \times BW_{eff} \times \log_2 \left(1 + \frac{SINR_{ue}}{\Gamma} \right)}{P_e / \eta + P_c}, \quad (7)$$

where;

P_c = Power dissipated in all circuit blocks

η = Efficiency of the transmit power amplifier The utility function is the EE based on the modified Shannon capacity theorem [27] and power consumption is based on the LTE model [28]. The value of BW_{eff} (Bandwidth Efficiency) and Γ (SINR efficiency) are set to 0.56 and 2 respectively [27].

We can now define the utility function of one SON entity or eNB U_e as the sum of EE when it is delivering data to its associated UEs. The objective of our system is to maximize this cell utility function, which is also its effective EE.

$$U_e = \sum_{u=1}^N U_{ue} \quad (8)$$

where;

U_e = utility function for eNB e

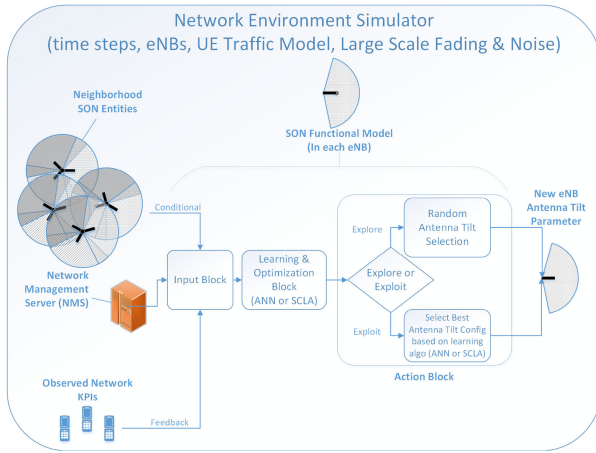
III. SELF OPTIMIZATION MODEL

SON ENTITY

Each SON entity or eNB in the simulated network runs its optimization function independently. The functional model of the SON entity and its interaction with its environment is shown in Fig. 4. The self-optimization model consists of: (i) an input block that accepts different system inputs, (ii) a learning and optimization block that learns and predicts optimum eNB antenna tilt based on the inputs and past feedback, and (iii) an action block that decides either to explore the neighbourhood environment and select a random antenna tilt, or exploit the learning done and accept the proposed tilt angle by the previous block. The system inputs comprise its own neighborhood antenna tilts, mobile users' data, and different selected KPIs. The simulated environment runs for predetermined time steps to get feedback in the form of KPIs observed. The details of the algorithms used in the self-optimization process are explained in the subsequent subsections.

A. REINFORCEMENT LEARNING

RL is a branch of machine learning, where an agent repeatedly interacts with its environment to learn which action yields the maximum reward in a given state while targeting a long-term objective function. The task of RL can be described as a Markov Decision Process (MDP), where the state space, explicit transition probability, and reward function are not necessarily defined [29]. Thus, RL can handle scenarios that mimic real-world complexity, like the LTE environment [30]. We implement the network as a multi-agent RL system, whereby each eNB has an associated agent. We compare two RL schemes, i.e., Q-learning using ANN and SCLA.


FIGURE 4. SON entity.

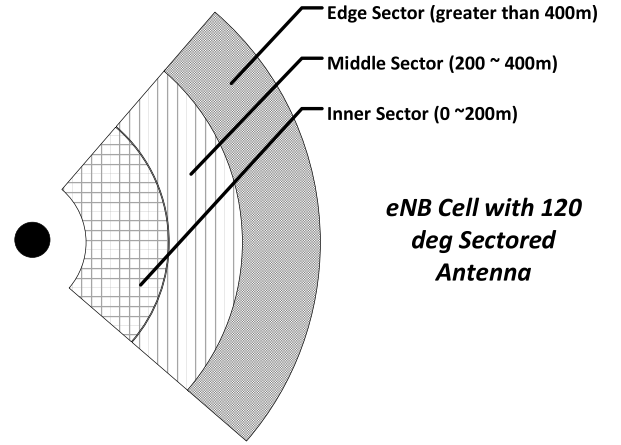
We maximize the eNB throughput by selecting the optimum tilt angle for any given input state through the RL agent.

The first scheme, $ANN = f(\theta, u_e(x, y), AvNTH_e, U_e)$, has three inputs i.e., own antenna tilt position θ , the associated mobile user positions $u_e(x, y)$ and mean of neighborhood cells throughput $AvNTH_e$ (Refer to Eq. 9). The last variable is the feedback U_e which is the utility of eNB e from Eq. 8. In the second scheme $SCLA = f(\theta, U_e)$, we only consider the tilt angle θ and the associated user positions $u_e(x, y)$. Thus, ANN is a complex learning scheme as compared to $SCLA$.

$$AvNTH_e = \frac{\sum_{i=1}^k TH_i}{k} \mid d_{intercell} \leq NeNBRange \quad (9)$$

We consider an eNB e , that interacts with its environment \mathbb{E} . The state space is given as $s \in S$, where S is the combination of all possible states. For the case of $ANN = f(\theta, u_e(x, y), AvNTH_e, U_e)$, the state space is made up $6 \times 8 \times 5 \times 5 = 1200$ combinations; First 6 are for the antenna tilt position $\theta = 6, 8, 10, 12, 14, 16$; Second 8 combinations of 3 bits represents the three sectors i.e. near, mid and edge having active UEs or not (Refer Fig. 5) and last two 5 values represent five throughput ranges of $AvNTH_e$ and TH_e form 0 to 1400 kbit/sec. Each eNB e , selects a particular θ tilt angle $a \in A = a_1, a_2, \dots, a_n, n \in \mathbb{N}$ depending on the feedback U_e . This interaction of an eNB e with its environment \mathbb{E} , where outcomes are partly random and partly based on a decision maker, is formally known as MDP and is a 5-tuple $(S, A, Pr_a(s, s'), Re_a(s, s'), \gamma)$. Here, $Pr_a(s, s') = Pr(s_{t+1} = s' \mid s_t = s, a_t = a)$, is the probability that the action a in state s at time t will bring agent to state s' at time $t + 1$. $Re_a(s, s')$ is the reward computed after moving to new state s' . $\gamma \in (0, 1]$ is the discount factor that discounts the rewards steadily as the system moves into next states. The typical range of γ is from 0 to 1. The problem of MDP is finding a decision making policy Π that yields maximum reward while in a state s .

RL aim is to learn an optimal action-selection policy Π that maximises the cumulative reward over time. Given E as


FIGURE 5. eNB User Coverage Regions.

expectation, this can be given in the following equation.

$$V^\Pi(s) = E \left[\sum_{t=0}^{\infty} \gamma^t r_t \mid s_0 = s \right] \quad (10)$$

Given that the Markov property defines that any future state depends only on the current state regardless of the previous states, we can rewrite Eq.10 as Eq.11, given $R_{s,a}$ as the mean of the immediate reward r_t .

$$V^\Pi(s) = R_{s,a} + \gamma \sum_{s' \in S} Pr_a(s, s') V^\Pi(s') \quad (11)$$

Therefore, if the cumulative reward is to be maximized, a maximum policy Π^* is required which can be found if $R_{s,a}$ and $Pr_a(s, s')$ are known.

$$V^{\Pi^*}(s) = \max_{a \in A} \left[R_{s,a} + \gamma \sum_{s' \in S} Pr_a(s, s') V^{\Pi^*}(s') \right] \quad (12)$$

B. Q-LEARNING

Q-learning is a model-free RL technique that can be used to find an optimal action-selection policy Π for any given (finite) MDP, when $R_{s,a}$ and $Pr_a(s, s')$ are not known. Q-learning typically fits our scenario, where each eNB has control over its antenna tilt but not over its neighboring eNB's antennas. Q-learning works by learning an action-value function while following an optimal selection policy to achieve the desired utility in small incremental steps. The Q-function used in Q-learning is defined as:

$$Q^\Pi(s, a) = R_{s,a} + \gamma \sum_{s' \in S} Pr_a(s, s') V^\Pi(s') \quad (13)$$

Since Q-function depends on discounted cumulative rewards, so it will be maximum when the action selection policy is optimal i.e.

$$Q^{\Pi^*}(s, a) = R_{s,a} + \gamma \sum_{s' \in S} Pr_a(s, s') V^{\Pi^*}(s') \quad (14)$$

The discounted cumulative state function is thus:

$$V^{\Pi^*}(s) = \max_{a \in A} [Q^{\Pi^*}(s, a)] \quad (15)$$

From Eq. 15, if we find the maximum Q-function, we also find the optimal policy. Usually, Q-function is found recursively using Eq. 16, also known as the Bellman Equation [31]. It is a simple value iteration update that assumes the old Q value and makes a correction based on the feedback observed after moving to the next state. In the start, Q returns an (arbitrary) fixed value for any state-action pair. In our case, we initialize Q to zero matrix. The correction on any $Q(s, a)$, is based on the computed reward, learning rate $\alpha \in (0, 1)$ and future rewards discount factor $\gamma \in (0, 1]$.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha_t [r_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (16)$$

where;

r_{t+1} is the reward observed after performing action a_t in state s_t .

The recursive update from the above Eq. 16 will ultimately achieve the optimal required Q value function i.e. $Q(s, a) \rightarrow Q^*(s, a)$ as done in Algorithm 1. For ease of understanding, we define the following two terms in this paper based on Eq. 16.

$$qtarget_t = r_t + \gamma \max_a Q(s_{t+1}, a) \quad (17)$$

$$qprediction_t = Q(s_t, a_t) \quad (18)$$

The advantages of Q-learning to our system model are; (i) It can compare the expected utility of the actions available without requiring a model of the environment. (ii) It can handle problems with stochastic transitions and rewards without requiring any adaptations. (iii) It is proven that for any finite MDP, Q-learning eventually finds an optimal policy [32].

1) Q-LEARNING ALGORITHM

The Q-learning algorithm for each eNB entity or learning agent is given in Algorithm 1. The algorithm is characterized by states S and a set of actions per state A . The algorithm calculates quantity $Q(s, a)$ or reward of a state-action pair combination (s, a) i.e. $Q : S \times A \rightarrow \mathbb{R}$. In our case, the state is defined by the number of inputs we want the agent to learn, and the actions are various antenna tilt positions. By performing an action $a \in \mathcal{A}$, the agent moves from one state to the next. A reward $r \in \mathbb{R}$ is computed after an agent takes action and observes the effect on desired system response.

The reward r is a numerical real value that depends on the effect an action has on the utility function given in Eq. 8. We consider r positive for an increase in the utility and negative in case a decrease in the utility is observed as given in Eq. 19. Over time, rewards add up for each state-action element in the Q matrix. The agent learns and can decide which action is optimal for any given state based on the total or cumulative reward for any state-action pair. The cumulative reward is a weighted sum of the expected values of

Algorithm 1 Q-Learning Algorithm for eNB Antenna Tilt

Define:

$s \in S$ the state space, $a \in A$ the action space or tilt angles, $T_{Episode} = Episode Time Period$, $alpha = Learning Rate$, reward function $R = f(KPIs)$, $\gamma = Discount Factor$

Initialize:

matrix $Q(s, a) = 0$, time $t = 0$, $Episode = 1$, Set s , α , γ to fix values, *Random select a*

repeat

if $t \bmod T_{Episode} == 0$

Apply a and observe KPIs, compute reward r from R and get new state s'

Select a' based on $\epsilon - greedy$ strategy

if *Selection = Random*

Random select a'

else

Select a' from $Q(s, a)$ matrix based on *max Q*

Update Q Table using Eq. 16

$s \leftarrow s'$, $a \leftarrow a'$

until $t = End Simulation$

the rewards of all future steps starting from the current state, where the weight for a step from state Δt steps into the future is calculated as $\gamma^{\Delta t}$.

$$r_t = \begin{cases} Positive, & \text{if } U_e(t) \geq U_e(t-1) \\ Negative, & \text{otherwise } U_e(t) < U_e(t-1) \end{cases} \quad (19)$$

where;

$U_e(t)$ = utility for eNB e at time step t

2) NEURAL NETWORK AS $Q(s, a)$ FUNCTION APPROXIMATION

Simple Q-learning using a $Q(s, a)$ matrix for the iterative update has two main drawbacks; an exponential increase in dimensionality with the addition of new inputs and a lack of ability to estimate Q value for states that are not visited. ANNs are typically used in non-linear statistical data modeling cases where complex relationships between inputs and outputs are observed or may have some behavior patterns. They can learn optimum near approximations of these non-linear input-output relationships given sufficient training cases or time. In our simple Q-learning case, the training of a large $Q(s, a)$ matrix through Eq. 16 becomes difficult as the number of states s and actions a increase. Thus, ANN as a tool is well suited for implementing $Q(s, a)$ matrix as a function to select the best action a for any given SON entity state s [33]. The block level design of Q-learning with ANN is shown in Fig. 6.

ANNs consist of interconnected layered groups of nodes that process information in a similar pattern as neurons in the human nervous system of function [34]. The first layer of ANN takes in weighted inputs, and the last layer delivers the outputs required by the learning agent. In between, the input and output layers, are the middle hidden layers that serve to provide different linear and non-linear combinations of

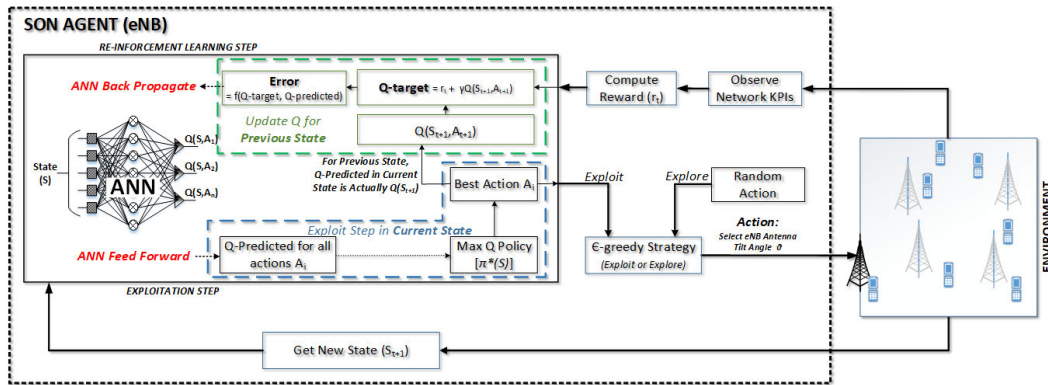


FIGURE 6. Q-learning with ANN Block Diagram.

inputs. Input signals travel from the input to the output layer, possibly after crisscrossing these hidden layers many times. In our $ANN = f(\theta, u_e(x, y), AvNTH_e, U_e)$ implementation, we have considered simple three-layer ANN structure i.e. one input, one hidden and one output layer with rectified linear unit as activation function. This basic ANN structure offers minimum complexity and requires less time for processing and training. Let Nd_i , Nd_h and Nd_o denote the number of nodes in the input, hidden and output layer respectively. For our case $Nd_i = 6$, i.e. one for θ , three for indicating each near, mid or edge UE active sectors, one for $AvNTH_e$ and one for U_e . $Nd_o = 6$, giving six Q-values for each possible antenna tilt position θ . No specific rule exists for selecting Nd_h , and varies with the problem and system design; however, the following thumb rule is generally used in choosing Nd_h [35].

$$Nd_h = \sqrt{Nd_i + Nd_o} + NDConst \quad (20)$$

where;

$$NDConst = \text{constant between 1 and 10}$$

3) ANN BACK PROPAGATION ALGORITHM

In a typical ANN design, the signal at the interconnecting branch between two artificial neurons is a real number while the output of each artificial neuron is a non-linear weighted sum of its inputs (Fig. 7). These weights are updated during the learning process to reach an optimum output. One of the ways ANN can learn and update its weights is through the back-propagation algorithm using gradient descent and sigmoid function [34]. Unlike for training samples in supervised learning in RL case, the rewards are used as target errors for weight updates in the back-propagation algorithm. This can be expressed by the following equation:

$$Error_t = \frac{(qtarget_t - qprediction_t)^2}{2} \quad (21)$$

We have implemented ANN-based Q-learning the same way as the Google DeepMind project implemented for its

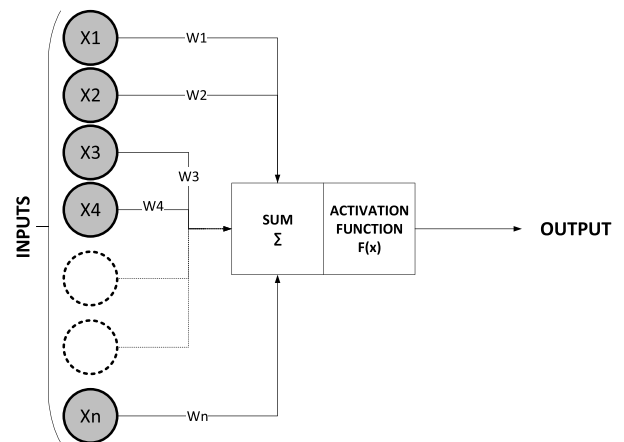


FIGURE 7. Typical structure of a neural network node.

Atari playing algorithm [33]. Specifically, DeepMind built a network that accepts a state and outputs separate Q values for each possible action in its output layer instead of one specific output. The difference in implementation of our particular work is depicted in Fig. 8. In a typical implementation, ANN would be used to learn Q matrix based on input state s and action a to give one fixed $Q(s, a)$ value. Thus in this case, while deciding which action to prefer, we would have to feed forward the ANN for all the possible actions and then compare their output Q values to find the best choice. In Google DeepMind the implementation of ANN is designed to give Q values for all possible actions, as shown in Fig. 8. Thus, with this improvement, we feed forward the ANN only once and then compare the output Q values to select the best action possible. This modified ANN design has fewer inputs and reduces processing time. The update frequency of algorithm is 4 [33, table 1], i.e., after taking 4 actions, this algorithm updates its weights. Regarding the memory requirement, RL agent

Algorithm 2 Back-Propagation ANN Q-Learning Algorithm for eNB
Define:

RL input = $\langle \vec{S}, \overrightarrow{QTarget} \rangle$, $\vec{S} = (a, UE_{Sec_{near}}, UE_{Sec_{mid}}, UE_{Sec_{edge}}, AvNTH_e, U_e)$ is the state vector input to ANN, where $UE_{Sec_{near}}$, $UE_{Sec_{mid}}$ and $UE_{Sec_{edge}}$ indicate presence of UEs in near, mid and edge sectors respectively.

$\overrightarrow{QTarget}$ is the vector of target network Q output values corresponding to each action a , α = Learning Rate, Nd_i = Number of ANN inputs, Nd_h = Number of nodes in hidden ANN layer, Nd_o = Number of ANN outputs, x_{ji} = input from node j to node i , w_{ji} = input weight from node j to node i .

Initialize:

α = small value e.g., (0.05), ANN feed forward network with Nd_i inputs, Nd_h hidden layer nodes and Nd_o outputs. All weights w_{ji} are set to random small values between -0.05 to 0.05.

repeat until end of simulation

For each RL input $\langle \vec{S}, \overrightarrow{QTarget} \rangle$, do Perform Feed forward step

 Feed input state vector \vec{S} and compute the ANN output Q value $\overrightarrow{QPrediction}$

 Select the antenna tilt action a corresponding to $\max_a \overrightarrow{QPrediction}$

 Execute action a and observe reward r

 Observe new state \vec{S}' and compute reward from Eq. 19

 Compute $\overrightarrow{QTarget}$ using Eq. 21

 Compute and back propagate the error in the network based on sigmoid function derivative

 For each ANN output node v , compute error δ_v
 $\delta_v \leftarrow \overrightarrow{QPrediction}_v(1 - \overrightarrow{QPrediction}_v)(\overrightarrow{QTarget}_v - \overrightarrow{QPrediction}_v)$

 For each hidden node h , compute error δ_h
 $\delta_h \leftarrow \overrightarrow{QPrediction}_h(1 - \overrightarrow{QPrediction}_h) \sum_{v \in outputs} w_{vh} \delta_v$

 Update each network weight w_{ji}

$w_{ji} \leftarrow w_{ji} + \delta w_{ji}$

 where;

$\delta w_{ji} = \alpha \delta_j S_{ji}$

randomly chooses mini-batch size experience samples from the replay buffer and performs the loss calculations to predict the action, which minimizes this loss. The experience sample consists of state, action, reward, and next state. Since there are 1200 possible states (as mentioned in sec. III-A), 6 actions, 2 rewards (positive or negative), therefore $11 + 3 + 1 + 11 = 26$ bits are required to store one state transition experience. The replay buffer size is 1000000; therefore, the memory size at each node would be 26M bits. The complete Q-learning with ANN back-propagation algorithm is given in Algorithm 2.

C. OPTIMIZATION USING STOCHASTIC CELLULAR LEARNING AUTOMATA METHOD

In order to compare the performance of reinforcement Q-learning with multiple inputs with technique that takes fewer inputs, we have used SCLA. In our earlier work [19], [36], hlwe have applied this technique for selecting orthogonal component carriers for neighboring femtocells in HetNet environment. We also have successfully optimized antenna tilt of macrocells in the same environment model as in this paper. Particularly to HetNet, we have shown that SCLA approach meets the SON requirements of scalability, stability, and agility. SCLA follows a distributed architecture that allows quick adaptability to changes in the environment.

SCLA stems from the idea of applying stochastic learning techniques in cellular automata (CAs). CAs are mathematical models for systems consisting of large numbers of simple identical components with local interactions. The simple components act together to produce complex emergent global behavior. By combining machine learning capability like stochastic learning automata to the plain CA, the new model formed is known as SCLA. Stochastic learning automata is a finite state machine and can learn from both stationary and non-stationary environment requiring only environment feedback to achieve better performance [37]. As there are no predetermined relationships between stochastic learning automata actions and the responses, so there is no requirement for a closed-form system model. Further details on CA, stochastic learning automata and SCLA can be found in [36]. Similar to our earlier work [19], SCLA picks out one eNB antenna tilt angle or action a from all possible positions or actions i.e., $\Lambda = |A|$. This selection is done according to the probability vector $p_t^e = [p_t^e(1), p_t^e(2), \dots, p_t^e(j), \dots, p_t^e(\Lambda)]$ for eNB e at time t . Once the tilt angle has been selected we update the probability vector by using the Discrete Pursuit Reward Inaction (DPRI) pursuit algorithm [38]. SCLA learns on the basis of feedback from the environment. A positive or negative reinforcement signal r is based on the earlier reward criteria we have set for the Q-learning algorithm (Refer to Eq. 19). The eNB probability vector is updated according to the following equation:

$$p_{t+1}^e(j) = \begin{cases} p_t^e(j) + (k-1) \times \alpha & , \text{if } r_t = \text{Positive} \\ p_t^e(j) - \alpha & \text{otherwise} \end{cases} \quad (22)$$

The pseudo-code of the SCLA algorithm is given in Algorithm 3. As the simulation time step t progresses each eNB learning entity will continue to learn and improve on its antenna tilt angle selections and reach an optimal level where the probability of the best tilt angle will almost reach unity. Reaching this stable condition is desirable if the neighbor eNBs do not change their tilt selections. However, the model maintains its dynamic nature and can respond to any new change in the neighborhood environment.

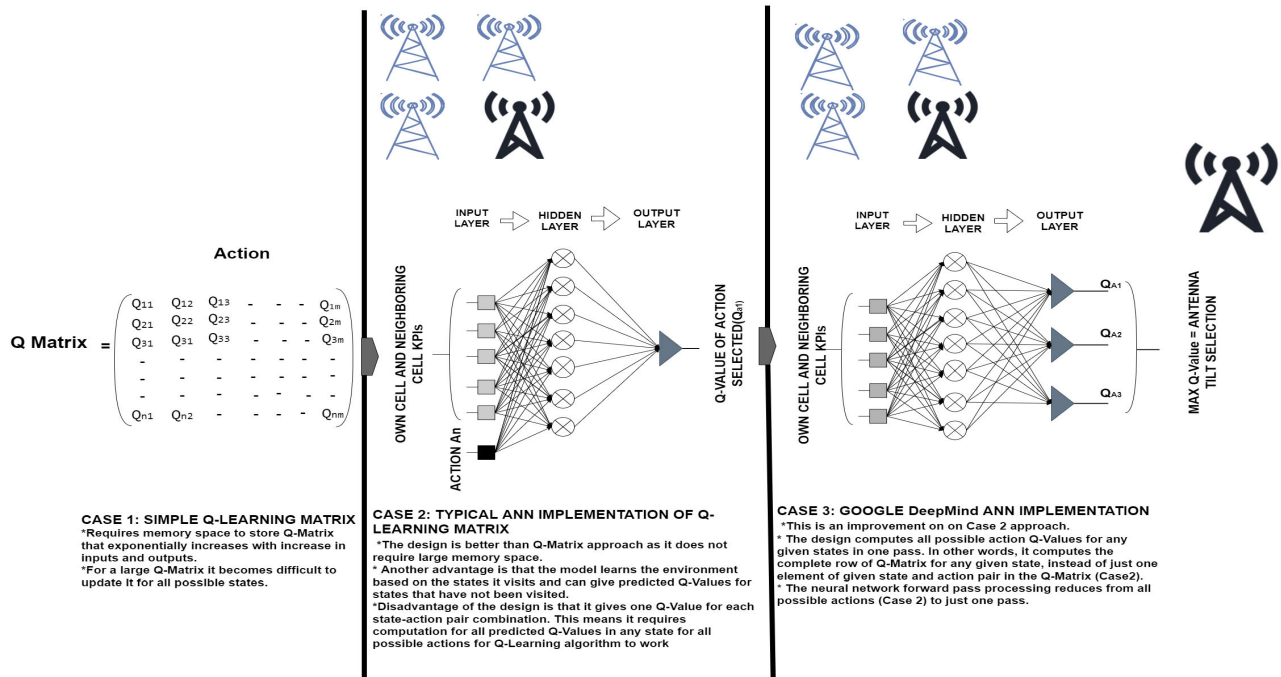


FIGURE 8. Comparison of Simple Q-learning with Q-learning with ANN and our Google Q-learning with ANN Implementation.

Algorithm 3 Stochastic Automata eNB Antenna Tilt Selection

Define: eNB Antenna tilt vector, $A = [a(1), a(2), \dots, a(j), \dots, a(\Lambda)]$, eNB Antenna probability vector, $p_t(j) = \frac{1}{\Lambda} \forall j \in \{1, \dots, \Lambda\}$, utility U_e , reward r , time t , Episode $T_{Episode} = 1$, Averaging Time Period (AvP),
Initialization: $U_e = 0, r = 0, t = 0, T_{Episode} = 1, T_{Period} = AvP$
 Random select a from A
repeat
 if $t \bmod AvP == 0$, and $T_{Episode} > 0$
 Apply a and observe KPIs using moving averaging filter
 Calculate reward r using Eq. 19
 Select a based on *epsilon – greedy* strategy
 if *Selection = Random*
 Random select a from A
 else
 Update probability vector using 22
 Select $a(j)$ from A based on $\max p_t(j)$
 Increment $T_{Episode}$
 else
 Increment t
until $t = End \text{ Simulation}$

D. EPSILON GREEDY STRATEGY - EXPLORATION VERSUS EXPLOITATION

RL systems converge to a good optimal action selection policy if there is a balance between the amount of exploration and exploitation. This balance also helps in achieving the

required agility in RL agents so that they can quickly follow developments in their environment. The ability to continually explore the environment while exploiting the learned data is an essential factor in determining the reactivity of a system. However, it is difficult to explore and exploit at the same time, so a balance has to be made between exploration and exploitation. In our work, we desire that network elements should be agile enough to adjust and adapt to changing environment scenarios. For this, we select the $\epsilon - greedy$ strategy, which allows exploration ϵ times and exploitation $(1 - \epsilon)$ times. Exploration is done with a uniform selection, without preference for any particular action. In the exploitation step, we choose the action corresponding to the best Q value in Algorithm 1 or probability in Algorithm 2.

E. COMPUTATIONAL COMPLEXITY

In the Q-table based reinforcement learning, an action may be selected in constant time $\mathcal{O}(1)$. The Q-table update procedure executed after receiving the reward have also constant time complexity. But due to the drawbacks of the Q-table in the presence of the non-linear input signal, we replace the Q-table with an ANN as a function to select the best a for any given SON entity state s as shown in the Fig. 6. In general, the matrix multiplication $A_{m \times n} * B_{n \times p}$ has the complexity of $\mathcal{O}(mnp)$. Since the activation function is an element-wise function, with n inputs, it has run-time complexity of $\mathcal{O}(n)$. The computational complexity of the feed-forward propagation in our ANN-based Q-learning model is $\mathcal{O}(Ts * (Nd_i Nd_h + Nd_h Nd_o))$. The ANN uses backpropagation to learn and update its weights with the computation

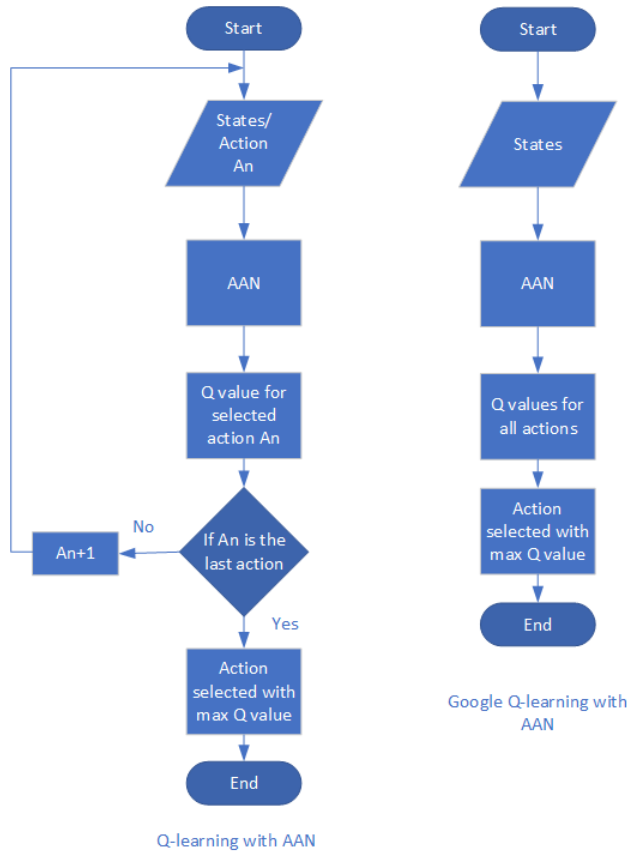


FIGURE 9. Flowchart diagram of Q-learning with ANN and Google Q-learning with ANN.

complexity of the order of $\mathcal{O}(Ts * N * (Nd_i Nd_h + Nd_h Nd_o))$, where Ts is the number of training samples and N is the number of episodes [39].

IV. KPIS USED FOR COMPARING SON TECHNIQUES

For data-carrying multimedia content, it is more desirable to have a constant downlink data flow to existing users even at cell boundaries, instead of dropping their calls in between. Call rejections, and slow data links are more acceptable to a user instead of call drops [40]–[42]. Based on this assumption, we define the following KPIS to gauge the performance of a SON technique.

- Average eNB EE for edge users in the system. This determines the ability of an optimization scheme in an eNB cell to yield optimum throughput in the face of a given environmental situation and hence is one measure of network QoS.
- Average system EE. This shows the ability of a network to provide an optimum bit rate to a user under a given environmental situation. Higher user data rate and cell throughput translate into better network quality experience by users or Quality of Experience (QoE) as most applications now require HD quality service.

- Average file transfer time recorded by users in the network. Lower file transfer time is preferred indicating the capacity of the network for delivering multimedia content with less buffering time.
- Maintain Rate (MR) is defined as the ratio of the number of UEs that are able to download the complete data file (NUDComplete) to the total number of UEs that were granted access to the network (NUAccept). Note that not all UEs, that get access to the network will be able to download the complete data file because in some cases calls may get dropped when a UE moves from one cell coverage to the next, i.e. handover case. Call drops in handover can occur due to lack of coverage or lack of available resources in the new cell. Thus, higher MR translates to more satisfied users and hence better user QoE.

$$MR = \frac{NUDComplete}{NUAccept} \tag{23}$$

- Call rejection rate due to low coverage in handover case (RH), is defined as the ratio of the total number of UEs rejected in handover due to coverage (NURejHO), to the total number of accepted UEs for the call by the network for any given simulation time frame. This inversely shows the capacity of a network to support a running call when a UE randomly moves from one cell area to another given availability of PRBs. Thus, lower RH means better network performance.

$$RH = \frac{NURejHO}{NUAccept} \tag{24}$$

- UE rejection rate due to coverage issues in network access phase denoted as RA. This is defined as the ratio of the total number of UEs rejected in network access phase due to coverage issues (NURejAC), to the total number of UEs accessing the network for any simulation time frame (NUAccess). This inversely indicates the coverage capacity of the network to accept new UEs given the availability of PRBs. Thus, lower RA means better network performance.

$$RA = \frac{NURejAC}{NUAccess} \tag{25}$$

SIMULATION AND RESULTS

A. SIMULATION ENVIRONMENT

The mobile network simulator described in [43] has been modified for this work. This is a dynamic link-level simulator tool developed using Matlab in Orange Labs. The simulator performs correlated snapshots to account for the time evolution of the network. At the end of each time step which can typically vary from a tenth to one second, the new mobile positions are updated, new users are admitted and some other users leave the network (end their communications or are dropped) and handover events are processed. The complete workflow of KPIS calculations is shown in Fig. 11.

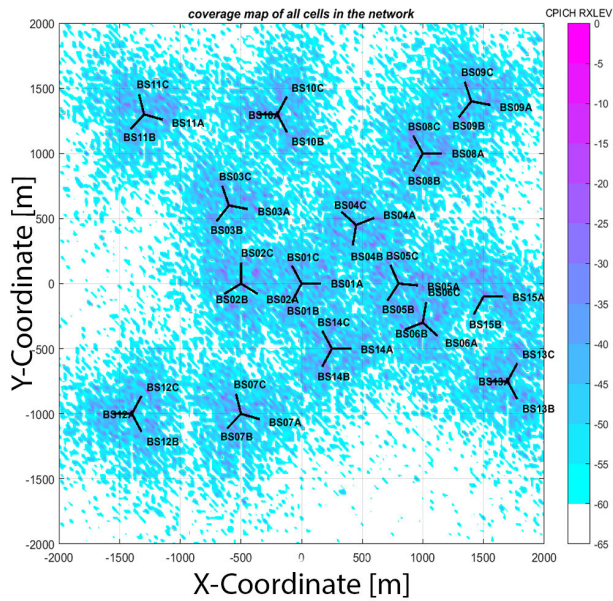


FIGURE 10. Simulated network environment with deployed eNB cells, such that three adjacent eNBs cover complete 360° at one location.

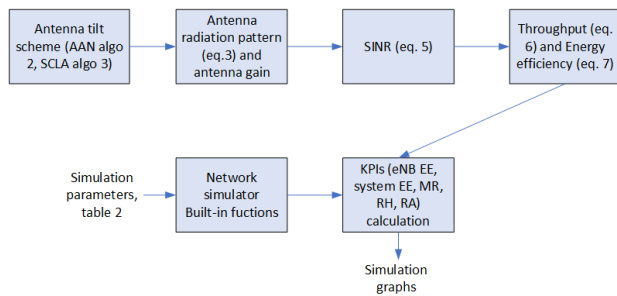


FIGURE 11. KPI measurement workflow.

A dense urban environment has been simulated as shown in Fig. 10. The ratio of mobile users to eNB is kept high so that all eNBs operate at full capacity and have comparatively more edge users accessing the network. Downlink FTP traffic is considered. UEs are deployed uniformly with random velocities. The downlink interference model includes thermal noise and large scale fading. Uplink traffic is not considered. The detail of simulation parameters is given in Table 2. The simulator proceeds in fixed time steps by taking correlated Monte Carlo snapshots. Users arrive following a Poisson process. With every time step, new users are added to the environment. Old users are checked for their FTP downloads, and if the file has been completely transferred these users are removed from the environment. A Call Admission Control (CAC) procedure based on eNBs resource availability and received signal strength by mobile users has been implemented. A user explores and selects the eNB with the highest Reference Signal Received Power (RSRP). The selected eNB then accepts the mobile if it has at least one PRB in spare. A call is dropped if the mobile enters an area with low cellular

TABLE 2. Network Simulator Parameters.

Parameter	Value
System BW	5 MHz
Cell Layout	45 eNBs, 120 deg sector
Mobiles	600
Inter-Site distance	0.5 - 2 Km
Sub Carrier Spacing	15 kHz
Thermal Noise Density	-173 dBm/Hz
PRBs per eNB	15
Pathloss	$128.1 + 37.6(\log_{10} R)$, R is in Km
Shadowing Standard Deviation	6 dB
PRBs assigned to one mobile	1 - 4 on first come first serve basis
Percentage of mobiles in motion	70%
Mobile speed	Max 15 m/s
Mobile Antenna Gain	1.5 dBi
Mobile Body Loss	1 dB
Traffic Arrival Rate λ	4 to 16
Traffic Model	FTP
File Size	57000 Kbits
Antenna Tilt	6 to 16 degrees
Antenna Tilt Steps	6
SCLA Learning Rate	0.004
ANN Hidden Layers	1
ANN Hidden Nodes	10
ANN Learning Rate	0.05
ANN Q Discount Factor	0.8
ϵ	0.01

coverage. KPIs are updated on every time step; however, the optimization algorithms run after fixed periods to account for normalization. The resulting KPI graphs and tables are based on the moving average filter of 500 steps. The initial 200 steps are excluded to avoid transient effects.

To get the average eNB cell edge throughput performance of the network we have divided the coverage area of a 120 deg sectored eNB into three regions and users in the range of 400m and beyond are considered edge users (Fig. 5). The simulation was run for 4000 time steps and results were collected for traffic arrival rates from 4 to 16. The mapping of the states to antenna tilt angles is given by the set $S = \{6, 8, 10, 12, 14, 16\}$ in which antenna tilt angle varies from 6 to 16 degrees in six steps.

B. RESULTS

In our earlier work [19], we showed that a simple technique like SCLA performed better than a more complex Q-learning one. In RL, a learning agent self-learns from its environment without requiring explicit training data. SCLA can quickly adapt because it is based on a small probability vector with a dimension equal to the number of actions possible and gets updated based on the feedback from the network with each time step. On the other hand, Q-learning consists of a large input state-action matrix that gets updated for every state-action combination and the feedback from the network. Also, as the state in the Q-learning matrix is based on the number of input learning parameters, so the order of the Q Matrix grows exponentially with the increase in the number of inputs and actions possible. Thus compared to the SCLA, Q-learning required more time to train its state-action matrix and adapt, resulting in poor performance. Moreover,

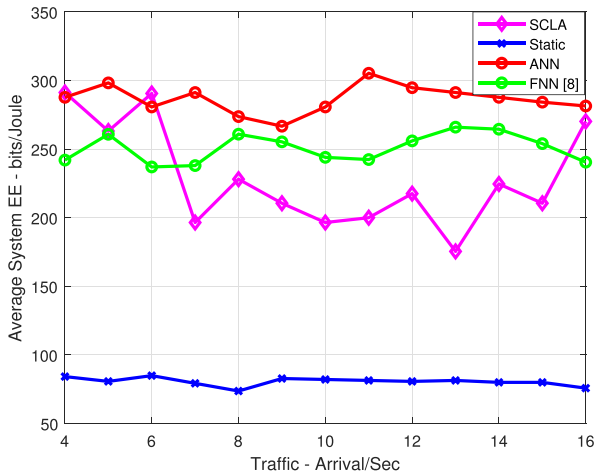


FIGURE 12. System average EE comparison.

TABLE 3. Performance comparison of system EE averaged over arrival rates $\lambda = 4, \dots, 16$.

Scheme	Average EE	Improvement
ANN	286.44	3.56
FNN [8]	250.95	3.12
SCLA	228.84	2.84
static	80.54	1

we found that within the same Q-learning technique, if we increase the number of input learning parameters, we find performance improvement. Figure 12 shows the average system EE in the network as a function of traffic arrival rate, for SCLA, ANN, fuzzy neural network (FNN) [8] and no-optimization or fixed static antenna tilt position case. It can be seen that all optimization schemes perform 2 to 4 times better than the non-optimized static case over the complete range of traffic arrival rate. Our simulation scenario is highly dynamic with moving users, adding/removing users, and handover are taking place. The static scheme does not apply optimization technique for antenna tilt even with the change of the positions and data transfer requirements of the users. In dynamic environment, some users are dropped and some users are added during the simulation time stamps but there is no electrical tilt in the static scheme. In ANN, SCLA, and FNN, antenna tilt controlled by the optimization of various network parameters through RL. The large improvement is due to the optimized RF configuration. First, the coverage of cells is optimized and the antenna beams are focused on the serving UEs while minimizing the interference to other cells. Secondly, due to the system-wide optimization transmit power to cell-edge users is minimized because of the reduced inter-cell interference.

In ANN case the users record about 25% and 14% better EE as compared to SCLA and FNN, respectively, for the range of $\lambda = 7 \dots 14$. The extremities of traffic arrival rate λ , are a region where either the resources are much more than the users requesting access to the network or the resources become constrained such that optimization does not give any

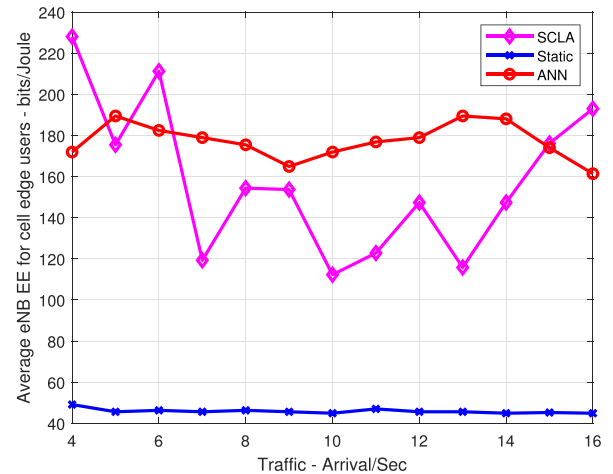


FIGURE 13. Average eNB EE for cell-edge users.

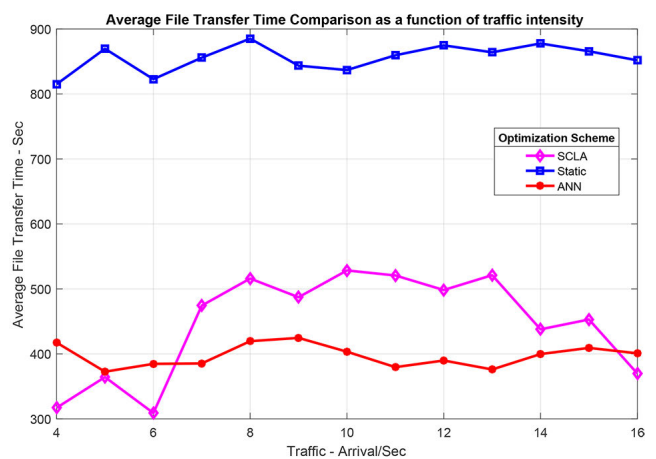


FIGURE 14. Average user file transfer time comparison plot.

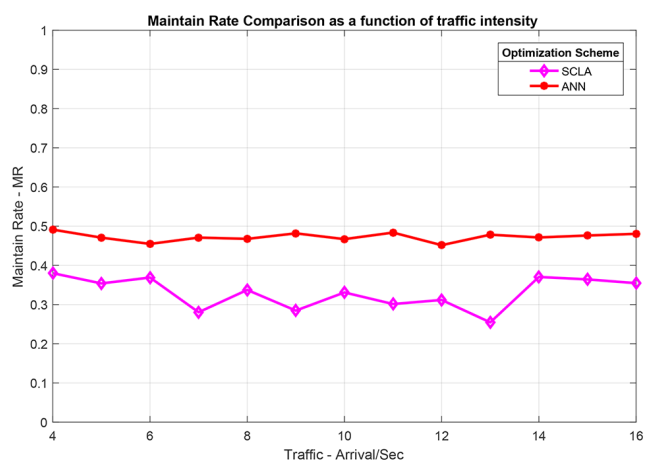


FIGURE 15. Maintain rate comparison of SCLA with ANN.

benefit. The above observation is also supported by the graph in Fig. 13, that gives a comparison of network eNBs edge EE performance as a function of traffic arrival rate for SCLA,

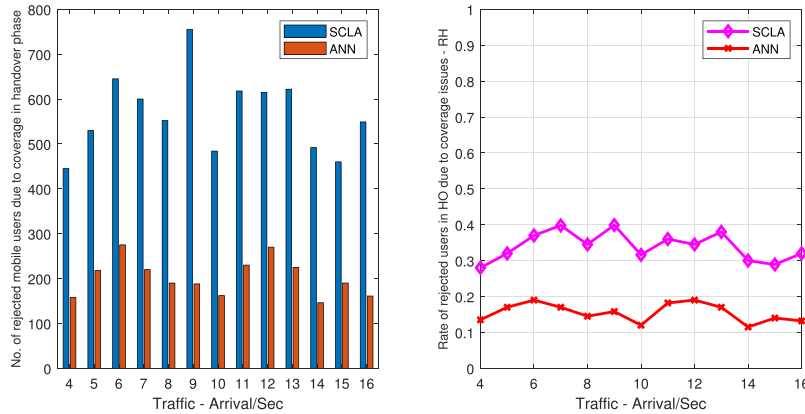


FIGURE 16. Comparison of users rejected in handover due to coverage issues for SCLA and ANN.

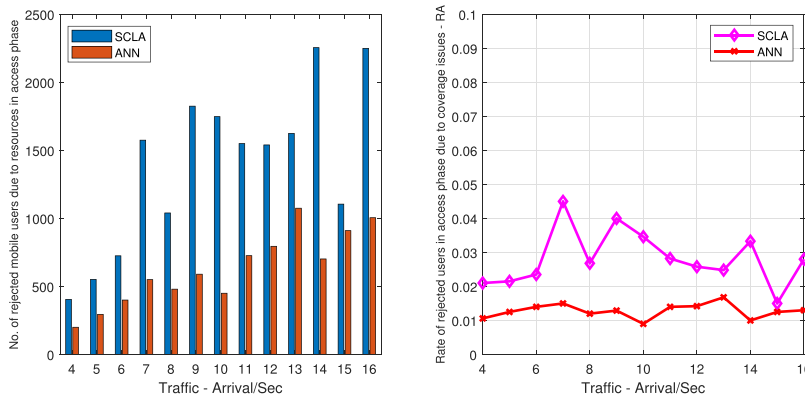


FIGURE 17. Comparison of users rejected in access phase due to coverage issues for SCLA and ANN.

ANN, and static cases. The figure shows that in the same range of $\lambda = 7 \dots 14$ the edge performance of eNB cells in ANN is more as compared to cells in the SCLA case. Thus a better cell edge performance gives a better data rate for users in the network. This performance improvement in ANN is due to the increase in the number of input dataset features as well as the number of input samples. From the graphs, it can be seen that both antenna tilt schemes do not deteriorate the performance with the increasing connection arrival rate but ANN responds well to the non-linearities of the system parameters of the cellular system as compared to the SCLA.

Figure 14, shows the observed file transfer time for network users as a function of traffic arrival rate for different SON schemes. The results, here again, are consistent with our previous figures, showing approximately 25% higher file transfer time for SCLA as compared to ANN for the range of $\lambda = 7 \dots 14$. Thus, gain in cell edge performance also lowers network file transfer time, which in turn means freeing up resources to accommodate more users in the network. This observation is further validated by MR, RH, and RA comparison graphs shown in Fig. 15, Fig. 16, and Fig. 17, respectively.

Fig. 15 shows the MR as a function of arrival rate. Maintain Rate (MR) is defined as the ratio of the number of UEs that are able to download the complete data file to the total number of UEs that were granted access to the network. Note that not all UEs, that get access to the network will be able to download the complete data file because in some cases calls may get dropped when a UE moves from one cell coverage to the next, i.e. hand over case. The MR graph shows an average 15% advantage of ANN over SCLA for the complete range of traffic arrival rate. This is because of ANN’s ability to learn the complex non-linear relationship between a large number of input features and the output function. We use a three-layered shallow neural network with an input layer consisting of six neurons, a hidden layer of ten neurons, and an output layer of six neurons. This is a one-vs-all logistic regression where only one output is 1 at a time corresponding to the predicted antenna tilt (out of possible six tilts). Though ANN gives $MR \approx 0.5$ but with minimal variance over the entire range of arrival rate.

Similarly, RH graph in Fig. 16 shows a 20% better performance of ANN over SCLA. RH is the call rejection rate due to low coverage in hand-over case and is defined as the ratio of the total number of UEs rejected in handover due

to coverage, to the total number of accepted UEs for the call by the network for any given simulation time frame. This inversely shows the capacity of a network to support a running call when a UE randomly moves from one cell area to another. The bar graph at arrival rate 4 shows $RH = 150$ and $RH = 430$ for ANN and SCLA, respectively. This is expected because of the low arrival rate network dynamics change slowly and performance is high. Then, it starts increasing up to an arrival rate of 6 and then decreasing and keeps changing harmonically. The ANN responds quickly to the change in the network dynamics and always outperforms the SCLA. The line graph exhibits that ANN results in $RH = 15\%$ as compared to the SCLA with the $RH = 35\%$ over the range of arrival rate.

Fig. 17 shows the call rejection rate in the network access phase RA versus arrival rate. This is defined as the ratio of the total number of UEs rejected in network access phase due to coverage issues, to the total number of UEs accessing the network for any simulation time frame. This inversely indicates the coverage capacity of the network to accept new UEs. The bar graph shows the number of rejected mobile users in the access phase due to the lack of PRB availability. For a constant number of PRB and increasing arrival rate, the number of call rejection increases for both SON schemes but ANN performance is better than the SCLA due non-linear capabilities of ANN. For RA case in the line graph, the improvement of ANN over SCLA is 1% which although not much but shows the capacity of ANN to reject less number of users requesting access to the network due to coverage as compared to SCLA.

V. CONCLUSION

In this paper, we have addressed the problem of optimizing eNB antenna tilt by proposing a SON architecture using tools from machine learning, i.e. ANN Q-learning and SCLA. Modern mobile networks collect real-time network statistics and have a standard mechanism of inter-cell communication defined by 3GPP. Taking advantage of this information resource as a source of learning, the SON architecture presented is fully distributed with optional information exchange with neighbors having overlapping antenna coverage. SCLA is a simple technique that learns from its action and the resulting feedback, whereas ANN learns from more inputs taken from its own eNB, network, and neighboring eNBs along with the feedback from its actions. The network simulation results show that, while the overall network performance is better than static or no optimization case, the proposed SON model based on ANN Q-learning gives better edge performance as compared to SCLA for FTP traffic. This shows that ANN can take advantage of the available neighborhood network information due to its ability to quickly learn from more number of input variables. These results offer hope that machine learning techniques like ANN, can be used for diverse network optimization tasks that depend on many input parameters.

REFERENCES

- [1] F. Ian Akyildiz, M. David Gutierrez-Estevez, R. Balakrishnan, and E. Chavarria-Reyes, "LTE-advanced and the evolution to beyond 4G (B4G) systems," *Phys. Commun.*, vol. 10, pp. 31–60, Mar. 2014.
- [2] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 1, pp. 336–361, 1st Quart., 2013.
- [3] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3GPP heterogeneous networks," *IEEE Wireless Commun.*, vol. 18, no. 3, pp. 10–21, Jun. 2011.
- [4] M. Sharsheer, B. Barakat, and K. Arshad, "Coverage and capacity self-optimisation in LTE-advanced using active antenna systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2016, pp. 1–5.
- [5] Y. Kishiyama, A. Benjebbour, H. Ishii, and T. Nakamura, "Evolution concept and candidate technologies for future steps of LTE-A," in *Proc. IEEE Int. Conf. Commun. Syst. (ICCS)*, Nov. 2012, pp. 473–477.
- [6] R. Shafin, L. Liu, V. Chandrasekhar, H. Chen, J. Reed, and J. Zhang, "Artificial intelligence-enabled cellular networks: A critical path to beyond-5G and 6G," *IEEE Wireless Commun.*, vol. 27, no. 2, pp. 212–217, Apr. 2020.
- [7] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, "A survey on mobile edge networks: Convergence of computing, caching and communications," *IEEE Access*, vol. 5, pp. 6757–6779, 2017.
- [8] S. Fan, H. Tian, and C. Sengul, "Self-optimization of coverage and capacity based on a fuzzy neural network with cooperative reinforcement learning," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 57, Apr. 2014.
- [9] P. A. S. Ordonez, S. Luna-Ramirez, and M. Toril, "A computationally efficient method for QoE-driven self-planning of antenna tilts in a LTE network," *IEEE Access*, vol. 8, pp. 197005–197016, 2020.
- [10] J. Wang, M. Yu, X. Zhang, and F. Jiang, "A reinforcement learning approach for self-optimization of coverage and capacity in heterogeneous cellular networks," *IEICE Trans. Commun.*, vol. 104, no. 10, pp. 1318–1327, 2021.
- [11] W. Guo, S. Wang, Y. Wu, J. Rigelsford, X. Chu, and T. O'Farrell, "Spectral- and energy-efficient antenna tilting in a HetNet using reinforcement learning," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2013, pp. 767–772.
- [12] M. Baianifar, S. M. Razavizadeh, H. Akhlaghpasand, and I. Lee, "Energy efficiency maximization in mmWave wireless networks with 3D beamforming," *J. Commun. Netw.*, vol. 21, no. 2, pp. 125–135, Apr. 2019.
- [13] C. Parera, Q. Liao, I. Malanchini, C. Tatino, A. E. C. Redondi, and M. Cesana, "Transfer learning for tilt-dependent radio map prediction," *IEEE Trans. Cognit. Commun. Netw.*, vol. 6, no. 2, pp. 829–843, Jun. 2020.
- [14] S. R. Samal, N. Dandanov, S. Bandopadhyaya, and V. Poulkov, "Adaptive antenna tilt for cellular coverage optimization in suburban scenario," in *Biologically Inspired Techniques in Many-Criteria Decision Making*. Cham, Switzerland: Springer, Dec. 2020, pp. 240–249.
- [15] T. Omar, T. Ketsoglu, and I. Naffaa, "A novel self-healing model using reinforcement & big-data based approach for 5G networks," *Pervasive Mobile Comput.*, vol. 73, Jun. 2021, Art. no. 101365.
- [16] R. Borrhalho, A. Mohamed, A. U. Quddus, P. Vieira, and R. Tafazolli, "A survey on coverage enhancement in cellular networks: Challenges and solutions for future deployments," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 1302–1341, 2nd Quart., 2021.
- [17] E. Balevi and J. G. Andrews, "Online antenna tuning in heterogeneous cellular networks with deep reinforcement learning," *IEEE Trans. Cognit. Commun. Netw.*, vol. 5, no. 4, pp. 1113–1124, Dec. 2019.
- [18] J. Yang, M. Ding, G. Mao, Z. Lin, D. G. Zhang, and T. H. Luan, "Optimal base station antenna downtilt in downlink cellular networks," *IEEE Trans. Wireless Commun.*, vol. 18, no. 3, pp. 1779–1791, Mar. 2019.
- [19] M. N. Qureshi, M. I. Tiwana, and M. Haddad, "Distributed self-optimization techniques for heterogeneous network environments using active antenna tilt systems," *Telecommun. Syst.*, vol. 70, no. 3, pp. 379–389, Jul. 2018.
- [20] G. Zhang, B. Eddy Patuwo, and M. Y. Hu, "Forecasting with artificial neural networks: The state of the art," *Int. J. Forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

- [21] J. Bourquin, H. Schmidli, P. van Hoogevest, and H. Leuenberger, "Advantages of artificial neural networks (ANNs) as alternative modelling technique for data sets showing non-linear relationships using data from a galenical study on a solid dosage form," *Eur. J. Pharmaceutical Sci.*, vol. 7, no. 1, pp. 5–16, Dec. 1998.
- [22] A. A. Periola and O. E. Falowo, "Immuno-neural network for spectrum prediction," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2014, pp. 1–6.
- [23] N. Dandanov, H. Al-Shatri, A. Klein, and V. Poulkov, "Dynamic self-optimization of the antenna tilt for best trade-off between coverage and capacity in mobile networks," *Wireless Pers. Commun.*, vol. 92, no. 1, pp. 251–278, Jan. 2017.
- [24] *Evolved Universal Terrestrial Radio Access (E-UTRA); Further Advancements for (e-UTRA) Physical Layer Aspects*, 3GPP, document TR 36.814, 2006.
- [25] *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN), Overall Description; Stage 2*, 3GPP, document TS 36.300, 2012.
- [26] A. Zappone and E. A. Jorswieck, "Energy-efficient resource allocation in future wireless networks by sequential fractional programming," *Digit. Signal Process.*, vol. 60, pp. 324–337, Jan. 2017.
- [27] P. Mogensen, W. Na, I. Z. Kovacs, F. Frederiksen, A. Pokhariyal, K. I. Pedersen, T. Kolding, K. Hugi, and M. Kuusela, "LTE capacity compared to the Shannon bound," in *Proc. IEEE 65th Veh. Technol. Conf. (VTC-Spring)*, Apr. 2007, pp. 1234–1238.
- [28] X. Xiao, X. Tao, and J. Lu, "Energy-efficient resource allocation in LTE-based MIMO-OFDMA systems with user rate constraints," *IEEE Trans. Veh. Technol.*, vol. 64, no. 1, pp. 185–197, Jan. 2015.
- [29] H. Yi Ong, K. Chavez, and A. Hong, "Distributed deep Q-learning," 2015, *arXiv:1508.04186*.
- [30] Y. He, Z. Zhang, F. R. Yu, N. Zhao, H. Yin, V. C. M. Leung, and Y. Zhang, "Deep-reinforcement-learning-based optimization for cache-enabled opportunistic interference alignment wireless networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 11, pp. 10433–10445, Nov. 2017.
- [31] T. Kamihigashi, "Elementary results on solutions to the Bellman equation of dynamic programming: Existence, uniqueness, and convergence," *Econ. Theory*, vol. 56, no. 2, pp. 251–273, Jun. 2014.
- [32] C. J. C. H. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
- [33] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Mar. 2015.
- [34] T. Mitchell, *Machine Learning*. New York, NY, USA: McGraw-Hill, 1997.
- [35] H. Xiao, L. Liao, and F. Zhou, "Mobile robot path planning based on Q-ANN," in *Proc. IEEE Int. Conf. Autom. Logistics*, Aug. 2007, pp. 2650–2654.
- [36] M. N. Qureshi and M. I. Tiwana, "A novel stochastic learning automata based SON interference mitigation framework for 5G HetNets," *Radio-engineering*, vol. 25, no. 4, pp. 763–773, Sep. 2016.
- [37] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, no. 1, pp. 237–285, Jan. 1996.
- [38] G. I. Papadimitriou, "A new approach to the design of reinforcement schemes for learning automata: Stochastic estimator learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 6, no. 4, pp. 649–654, Aug. 1994.
- [39] *StackExchange, Machine Learning—What is the Time Complexity for Training a Neural Network Using Back-Propagation?—Artificial Intelligence Stack Exchange*, StackExchange, New York, NY, USA. Accessed: May 22, 2022. [Online]. Available: <https://ai.stackexchange.com/a/5730/45919>
- [40] P. Gill, M. Arlitt, Z. Li, and A. Mahanti, "Youtube traffic characterization: A view from the edge," in *Proc. 7th ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, 2007, pp. 15–28.
- [41] M. Zink, K. Suh, Y. Gu, and J. Kurose, "Characteristics of Youtube network traffic at a campus network—measurements, models, and implications," *Comput. Netw.*, vol. 53, no. 4, pp. 501–514, Mar. 2009.
- [42] M. Haddad, E. Altman, R. El Azouzi, T. Jiménez, S. E. Elayoubi, S. Benjemaa, A. Legout, and A. Rao, "A survey on Youtube streaming service," in *Proc. 5th Int. ICST Conf. Perform. Eval. Methodol. Tools*, 2011, pp. 300–305.
- [43] R. Nasri and Z. Altman, "Handover adaptation for dynamic load balancing in 3gpp long term evolution systems," 2013, *arXiv:1307.1212*.



MUHAMMAD NAUMAN QURESHI received the B.E. degree in avionics from the College of Aeronautical Engineering, National University of Science and Technology (NUST), Pakistan, in 1997, and the M.S. degree in information security from Sichuan University, Chengdu, China, in 2007. He is currently with the Department of Electrical Engineering, COMSATS University, Islamabad, Pakistan. His research interests include 5G heterogeneous networks, self organizing networks, artificial intelligence, and fog networks.



MUHAMMAD KHALIL SHAHID (Member, IEEE) received the B.E. degree in electrical engineering from the University of Engineering and Technology, Lahore, in 1998, the M.S. degree in electrical engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2005, and the Ph.D. degree in engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. He is currently working as an Assistant Professor with the Higher Colleges of Technology. He has over 15 years of experience in telecommunication industry. He is the author of several journals and conference papers in the field of communications and information technology. He worked on LTE MiFi Clouds, Hotspots, Wingles, USB Dongles, Drive testing for CDMA/EVDO Network for checking QoS parameters using NEMO Analyzer, and Genex Probe. His current research interests include wireless communication, 5G communications, optical wireless communications, fiber optic systems and networks, optical transmission, optical fiber access networks, technology management, operational management, project management, and industrial organization.



MOAZZAM ISLAM TIWANA received the B.Sc. degree in electrical and electronics engineering from the University of Engineering and Technology, Taxila, Pakistan, in 2001, the M.Sc. degree in digital telecommunication systems from ENST, Paris, France, in 2007, and the Ph.D. degree in mobile communications from Telecom Sud-Paris Paris, France, in 2010. His Ph.D. work was with the Research and Development Group, Orange Laboratory, France Telecom. He has more than nine years of industrial and academic experience with research publications in the reputed international journals.



MAJED HADDAD received the Diploma degree in electrical engineering from the National Engineering School of Tunis, Tunisia, in 2004, the master's degree from the University of Nice Sophia Antipolis, France, in 2005, and the Ph.D. degree in electrical engineering from the Eurecom Institute, in 2008. In 2009, he joined France Telecom Research and Development as a Postdoctoral Research Fellow. In 2011, he joined the University of Avignon, France, as a Researcher Assistant. From 2012 to 2014, he was a Research Engineer at INRIA Sophia-Antipolis, France, under an INRIA Alcatel-Lucent Bell Laboratory Fellowship. He has been an Assistant Professor with the University of Avignon, since 2014. He has published more than 50 research papers in international conferences, journals, book chapters, and patents. His research interests include radio resource management, heterogeneous networks, green networks, complex networks, and game theory. He also acts as the TPC chair, a TPC member, and a reviewer for various prestigious conferences and journals.



IRFAN AHMED (Senior Member, IEEE) received the B.E. degree in electrical engineering and the M.S. degree in computer engineering from the University of Engineering and Technology, Taxila, Pakistan, in 1999 and 2003, respectively, and the Ph.D. degree in telecommunication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2008. From 2011 to 2017, he worked with Taif University as an Assistant Professor/an Associate Professor.

He was a Postdoctoral Fellow with Qatar University, from April 2010 to March 2011, where he worked on two research projects, wireless mesh networks with Purdue University, USA, and radio resource allocation for LTE with Qtel. He has also been involved in the National ICT Pakistan funded research project “Design and Development of MIMO and Cooperative MIMO Test-Bed” with Iqra University, Islamabad, Pakistan, from 2008 to 2010. He is currently working as an Associate Professor with the Higher Colleges of Technologies, United Arab Emirates. His research interests include wireless LAN (WLAN) medium access control (MAC) protocol design and analysis, 5G communications, mmWave and massive MIMO communications, performance analysis of wireless channels, energy-constrained wireless networks, and radio resource allocation. He served as the Session Chair for the IEEE Wireless Communications, Networking and Mobile Computing Conference held in Shanghai, China, in September 2007 and IEEE ICC 2016. He is an Active Reviewer of IEEE, Springer, and Elsevier journals and conferences. He is an Associate Editor of IEEE ACCESS journal.



TARIG FAISAL received the master’s degree in mechatronics engineering from IIUM University, in 2006, and the Ph.D. degree in signal processing from the University of Malaya, Malaysia, in 2011. He has been the Dean of Academic Operations with the Higher Colleges of Technology, since 2018. He has been a reviewer for multiple journals, including IEEE, Elsevier, Taylor & Francis, and Springer Nature. He has more than 20 years of academic and industry experience of which he worked

as an Engineering, an Assistant Professor, the Programs Chair, the Head of Department, the Division Chair, and the Campus Director. He is also a Chartered Engineering and a Senior Fellow of Higher Education Academy. His research interests include biomedical signal processing, intelligent systems, robotics, control, embedded system design, the IoTs, machine learning, and outcome-based education.

• • •