IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 24, 2020, accepted December 14, 2020. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2021.3051695

Reinforcement Learning-Based Resource Management Model for Fog Radio Access Network Architectures in 5G

NOSIPHO N. KHUMALO[1,2], OLUTAYO O. OYERINDE[1], (Senior Member, IEEE), AND LUZANGO MFUPE[2], (Senior Member, IEEE)

[1]School of Electrical and Information Engineering, University of the Witwatersrand, Johannesburg 2000, South Africa
[2]Next Generation Enterprises and Institutions, Council for Scientific and Industrial Research, Pretoria 0184, South Africa

Corresponding author: Nosipho N. Khumalo (nkhumalo3@csir.co.za)

ABSTRACT The need to cope with the continuously growing number of connected users and the increased demand for mobile broadband services in the Internet of Things has led to the notion of introducing the fog computing paradigm in fifth generation (5G) mobile networks in the form of fog radio access network (F-RAN). The F-RAN approach emphasises bringing the computation capability to the edge of the network so as to reduce network bottlenecks and improve latency. However, despite the potential, the management of computational resources remains a challenge in F-RAN architectures. Thus, this paper aims to overcome the shortcomings of conventional approaches to computational resource allocation in F-RANs. Reinforcement learning (RL) is presented as a method for dynamic and autonomous resource allocation, and an algorithm is proposed based on Q-learning. RL has several benefits in resource allocation problems and simulations carried out show that it outperforms reactive methods. Furthermore, the results show that the proposed algorithm improves latency and thus has the potential to have a major impact in 5G applications, particularly the Internet of Things.

INDEX TERMS Fifth generation, fog computing, Internet of Things (IoT), radio access network, reinforcement learning, resource allocation.

I. INTRODUCTION

The forthcoming ubiquity of the Internet of Things (IoT) in everyday life, combined with the continuously growing number of connected users and the increased demand for mobile broadband services, have created a challenge for current cellular networks and necessitate an essential change in the way in which wireless networks are designed and modelled [1]. This challenge, which is particularly eminent when considering the need to deal with the exponential amounts of data produced at the edge of the network, is further exacerbated by the current network state, which is both extremely heterogeneous and immensely fragmented [2].

Fifth generation wireless network technologies (5G) are the next generation in mobile communications, beyond the current fourth generation (4G) and Long Term Evolution (LTE) mobile networks, and promise to play a crucial role in enabling a better-connected networked society. 5G is anticipated to provide new opportunities that enable us to deliver unprecedented applications and services that can support new users and devices. These applications encompass massive machine-type communications (mMTC)- also known as the Internet of Things (IoT), enhanced mobile broadband (eMBB) requiring high data rates over a wide coverage area, and ultra-reliable and low-latency communications (URLLC) with stringent requirements on latency and reliability [3], [4].

The proposed architecture for 5G, in an effort to deal with the expanding amount of user traffic and the increasing number of IoT devices, is the cloud radio access network (C-RAN) architecture. In the C-RAN approach, the function of processing data is borne by a pool of centralised baseband units (BBU) inside the core network, which are characterised by a limited fronthaul [5]–[7]. In order for processing to take place in the centralised BBU pool, a high bandwidth fronthaul with low latency is required. However, the fronthaul

The associate editor coordinating the review of this manuscript and approving it for publication was Chi-Tsun Cheng.

in the C-RAN is prone to time-delay and capacity constraints, which presents several challenges for 5G applications, particularly when considering the massive traffic produced by IoT devices. Furthermore, CRAN does not exploit the storage and processing capacity of edge devices and may excessively burden the core network and consequently adversely affect the quality of service (QoS) experienced by the end users [8].

As a means to overcome the challenges in the CRAN effort, the notion of introducing fog computing in 5G RAN in the form of fog radio access network (F-RAN) has emerged as a promising architecture. The F-RAN approach emphasises bringing the computation capability to the edge of the network so as to enable a lower burden on the fronthaul and meet the demands of ultra-low-latency applications [9]. As a secondary advantage to reducing network bandwidth bottlenecks and improving latency, the F-RAN technique also bears great potential in very low Average Revenue Per User (ARPU) areas, particularly when the connection to the cloud is unavailable or limited [10]. These developing regions, which are characterised by a lack of adequate broadband infrastructure, are referred to as underserved areas.

Despite all these attempts to handle the growing demand of IoT applications, management of computational and network resources of processing entities in the 5G F-RAN (i.e. fog nodes) still remains a high priority goal for future network designs. Contrary to resources in the C-RAN, the resources at the network edge are inherently: (i) restricted in terms of computational resources - a constraint imposed by the limited processor size and power budget of edge devices, (ii) heterogeneous, and (iii) dynamic with variable workloads and applications contending for the limited resources [11]. Therefore, resource management continues to be one of the key challenges in 5G F-RAN.

Conventional legacy approaches to computational resource allocation in virtualised networks, such as the 5G F-RAN, are static mechanisms in which a fixed resource size pool (including storage resources, computing resources, and bandwidth resource) is allocated to each fog node when the network is configured [12]. However, the dynamic nature of F-RAN resources coupled with the heterogeneity and increasing complexity of IoT applications deem static allocation mechanisms insufficient for satisfying the needs of future mobile networks and necessitate dynamic allocation approaches that can predict changes in the workload and autonomously adjust resources accordingly.

Drawing from the literature review, there is a lack of studies in the area of designing dynamic and autonomous resource allocation mechanisms that allow each fog node to independently manage its computed power allocation. Nonetheless, learning-based resource allocation has been implemented in [13]–[15] through a centralised controller that acts as the primary decision maker for the service provider. Most research efforts aimed at addressing the computing resource allocation problem in 5G F-RAN are commonly focused on offloading data from the resource-constrained F-RAN to the core network, in order to meet the data rate and latency demands of eMBB and URLLC applications, respectively. For instance, the work in [16] considers offloading as a means to optimise latency. The major shortfall of the offloading approach is that it is ill-suited for networks in underserved areas, where the connection to the remote cloud is unavailable or limited. The issue of computing resource allocation for 5G F-RAN architectures in underserved regions is an area of research that has not been studied extensively in literature.

By taking this into account, this paper leverages the capabilities of machine learning and fog computing in order to address the computing resource allocation problem in 5G F-RAN architectures for mMTC services in underserved communities. Consequently, the main contributions of this paper include the following:

1) We propose reinforcement learning as a technique for dynamic and autonomous resource management and design an algorithm based on Q-learning.
2) We demonstrate, through simulation-based performance evaluation, the efficiency of the proposed algorithm.

The remainder of this paper is organised as follows. The state of the art is presented in Section II, along with an overview of the related work. Section III defines the system model and formulates the resource allocation problem. In Section IV, a reinforcement learning model for autonomous resource allocation in 5G F-RAN is presented based on proactive auto-scaling, including the learning parameters and the proposed Q-learning algorithm. After describing the simulation setup in Section V, the results are presented and discussed in Section VI. Finally, Section VII elaborates on the major contributions of this paper and gives a brief overview of possible extensions as part of future work.

II. RESOURCE MANAGEMENT TECHNIQUES IN 5G F-RANs

F-RANs have been presented as a promising architecture to provide high spectral and energy efficiency in future wireless networks. The potential of F-RANs has been highlighted in numerous relevant works, however, the cache resource optimization remains a challenging task due to the uncertainty and dynamics of user file requests. The work in [17] proposed a deep reinforcement learning (DRL) based algorithm as a means to improve this. In addition to cache resource optimisation, achieving ultra-low latency for emerging cellular networks is still challenging due to constrained fronthaul capacity. The work in [18] attempted to achieve ultra-low latency by presenting a distributed content sharing and computing mechanism combined with the greedy algorithm. The proposed approach, which was successful at optimising the transmission rate, was proven to be a sub-optimal solution.

In [19], an approach was devised based on DRL to minimise power consumption of the network in the long-term. The authors demonstrated that integrating transfer learning with DLR yields promising performance gains and requires much fewer interaction with the environment. The work in [20] made an effort to investigate the computation offloading problem by designing an algorithm to optimise the joint allocation of radio and computation resources. The proposed iterative algorithm showed promising performance gains, including computational complexity. Similarly, an iterative algorithm was adopted in [21] to optimise the CPU-cycle frequency, the transmit power control, and the offloading decision. The proposed solution, which was based on the conventional convex optimisation methods, minimised total energy consumption while guaranteeing that the restrictions on maximum latency tolerance and capacity are adhered to.

As part of the F-RAN resource management effort, the authors in [22] set out to design a resource allocation strategy based on differential game and bipartite graph multiple matching, and proposed a distributed uplink computation offloading strategy with Lyapunov theory and deviation update decision algorithm (DUDA). The proposed mechanism performed well in terms of system consumption and resource demand satisfaction rate. In [23], the resource allocation problem was formulated as a Markov Decision Process (MDP), for which an optimal decision policy was presented through reinforcement learning. The proposed resource allocation method learned from the IoT environment how to strike the right balance between two conflicting objectives of maximising the total served utility and minimising the idle time of the fog node. The transmission latency between fog nodes, node-to-UE, and fronthaul latency strongly depends on interference power from the undesired network element as well as end-users. At the same time, the computational latency increases with the queuing delay. In [24], a load balancing scheme was proposed to address the trade-off between transmission and computing latencies in F-RANs. The suggested method outperforms the greedy approach in terms of low latency and minimal task offloading to the cloud.

Considerable advances have been made in the area of designing algorithms for the management of resources in the 5G F-RAN architecture, with machine leaning paving the way for dynamic and autonomous mechanisms. Despite the promise, most of these efforts make the assumption that the resources are fixed and/or the network functions are executed on black boxes. The problem with these approaches is that they do not account for the dynamic formation of the 5G F-RAN. Furthermore, most approaches to the resource allocation problem adopt computation offloading, which may result in adverse consequences to the performance due to additional offloading delays in scenarios where the fog nodes are very resource constrained. There is a lack of studies that address the self-management of networks with softwarised and virtualised resources.

As a means to address the shortcomings identified in literature, we derive a mathematical model for 5G F-RAN architectures and propose a technique for autonomous and dynamic resource management in 5G F-RANs in the subsequent sections.

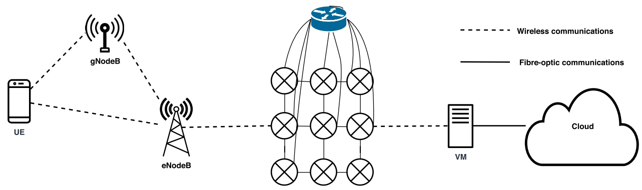

FIGURE 1. System architecture of F-RAN.

III. SYSTEM MODEL

The system model considers a hierarchical architecture, which is composed of UEs, 5G base stations, fog nodes and SDN-based remote cloud servers. UEs connect to the fog nodes through wireless communication links using 5G base stations, while fog nodes access servers in the remote cloud data centre through fibre-optic communication as illustrated in Fig. 1. This work is focused on underserved communities, which are often characterised by intermittent or no Internet connectivity. Therefore, most data processing is completed in the fog network, while the cloud is used for historical storage and batch analytics.

The model is based on dual radio connectivity, incorporating the architecture of 4G LTE and 5G New Radio (NR). The LTE eNB is deployed as the master eNB, while the NR eNB/gNB is the secondary eNB balancing the load and enhancing user throughput.

The network functions of the system are softwarised and run on isolated virtual machines (VMs) through the Network Function Virtualisation (NFV) technique. These VMs, also referred to as fog nodes, connect to each other using Software Defined Networking (SDN), while also monitoring and managing network traffic among them.

The specification of virtual network (VN) resource requirements is represented by a weighted undirected graph $G = (N; L)$, where N and L represent the sets of fog nodes and links respectively. Each virtual link $l_{ij} \subseteq L$ or fog node $i \subseteq N$ is characterised by requirements such as maximum delay, CPU, memory, bandwidth etc. In the network, the set of UEs is denoted by $K = \{1, 2, \ldots, |K|\}$. A set of UEs of a specific node n is denoted as K_n, while k_n denotes a single UE of the node. Various kinds of UEs send their task data to a certain fog node n, and the service arrival rate to the node is γ_{k_n} packets per second. For simplicity, it is assumed that each task packet has the same size of M bits.

A. DELAY MODEL

The transmission delay between a UE and a fog node is given by:

$$T_{k_n}^{tran} = \frac{M_{k_n}}{R_{k_n}} \tag{1}$$

where M_{k_n} denotes the data packet size, R_{k_n} denotes the achievable transmission rates in bits/s.

The signal-to-noise ratio (SNR) is used to denote the maximum transmission rate, through Shannon's rule on the upper bound limit for the achievable transmission rates on the capacity of a communications channel. The limit can be defined in dB, in relation to changes in available bandwidth B in Hz and SNR, as:

$$R_{k_n} = B\log_2(1 + SNR) \tag{2}$$

The transmission delay between the n^{th} fog node and the j^{th} node is given as:

$$T^{tran}_{n,j} = \frac{M_{n,j}}{R_{n,j}} \tag{3}$$

where $M_{n,j}$ is the fraction of data packet size being transmitted from the n^{th} node and the j^{th} node, $R_{n,j}$ denotes the achievable transmission rate between the n^{th} node and the j^{th} node and $n \subseteq N$.

On the fronthaul link, the transmission delay is defined as:

$$T^{tran}_{n,c} = \frac{M_{n,c}}{R_{n,c}} \tag{4}$$

where $M_{n,c}$ is the data packet size being transmitted from the node to the cloud and $R_{n,c}$ is the fronthaul capacity between the node and the cloud.

Finally, the total transmission delay for the UE becomes:

$$T^{tran}_{k} = T^{tran}_{k_n} + T^{tran}_{n,j} + T^{tran}_{n,c} \tag{5}$$

It is assumed that data processing only begins once all the packet data has been received by the fog node and that the task arrival rate follows a Poisson distribution. Considering M/D/1 queuing system, the computational delay incurred is given by:

$$T^{comp}_{k} = \frac{\mu_{k_n}}{2\mu_{k_n}(\mu_{k_n} - \gamma_{k_n})} + \frac{D_{k_n}}{f_{k_n}} \tag{6}$$

where D_{k_n} represents the required CPU cycles modelled as $D_{k_n} = M_{k_n}a_{k_n}$, a_{k_n} is the minimum processing density requirement (CPU cycles/bit) and f_{k_n} is the fraction of CPU clock frequency allocated. The first term in the equation represents the waiting delay in the computation queue and the second term is the execution delay.

B. THROUGHPUT AND UTILISATION MODEL

Using the assumption that all fog nodes in the system follow exponential service rates, the aggregate arrival rate can be defined as the sum of individual rates. If γ_n defines the individual arrival rate, then the aggregate rate can be denoted as:

$$\lambda_n = \sum_{k=0}^{K} \gamma_{k_n} \tag{7}$$

Each node's utilisation can be defined as:

$$U_n = \frac{\lambda_n}{\mu_n} \tag{8}$$

where μ_n denotes the average service rate of a fog node and C_n is the node's throughput, which is defined by:

$$C_n = \sum_{j=1}^{N} C_j P_{nj} + \gamma_n \tag{9}$$

in a system with N queues and associated fog nodes, given that packets leave the fog node n with the probabilities defined as $P_{n0} = 1 - P_{nj}$. The underlying notion behind the definition is that the possibility of a task leaving the system is equal to the complement of the probability that the task will be left in the system. Since the throughput arriving from the IoT network is known, C_0 can be set to λ and solve for the outstanding C terms.

C. PROBLEM FORMULATION

Given the abovementioned system model, the problem of resource allocation for a fog-enabled 5G communication system is formulated. Since the goal of the system is to minimise the total end-to-end latency experienced by users through computation resource allocation F while enforcing the maximum tolerable latency requirement constraint, the optimisation problem is defined as:

$$\min_{F} T = \sum_{k \subseteq K} T_k \tag{10}$$

subject to

$$\sum_{k \subseteq K} f_{k_n} \le f^{max}_{n}, \quad \forall k \subseteq K \tag{11}$$

$$T_k \le T^{max}_{k} \forall k \subseteq K \tag{12}$$

$$f_{k_n} \ge 0, \quad \forall n \subseteq N,\ k \subseteq K \tag{13}$$

where f^{max}_{n} is the total computation resources in the n^{th} node and T^{max}_{k} denotes the maximum tolerable latency of the k^{th} user. The constraint in (11) ensures that the computation resources on individual nodes are not allocated in excess, (12) guarantees that the service latency experienced by individual users do not exceed their maximum tolerable latency and (13) is the non-negative constraint on computation resource allocation. In this work, only CPU is considered as a computation resource.

IV. AUTONOMOUS RESOURCE MANAGEMENT MODEL

A. PROPOSED REINFORCEMENT LEARNING SYSTEM PARAMETERS

The need to deal with the ongoing rapid increase in IoT applications and services, especially in complexity, scale and connectivity, has created a need for the development of computer systems with an aptitude for self-management [25]. Conventional legacy approaches to system management involve algorithms that are programmed to only respond to pre-defined logic based on the specified cases, and inevitably respond poorly when there are unpredictable changes in the workload. Therefore, the proposed approach to the resource allocation problem leans towards an autonomous method in order to dynamically manage the heterogeneity of 5G applications.

In general, machine learning is aimed at constructing models and algorithms that are capable of decision making without adhering to predetermined rules [26]. Instead, the models and algorithms learn to apply data directly. In the case of reinforcement learning, as illustrated in Fig. 2, learning is

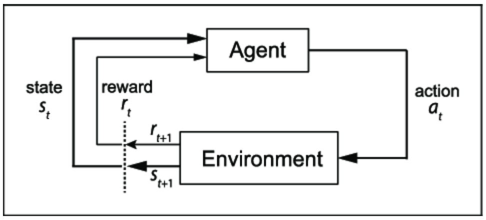

FIGURE 2. **Reinforcement learning model.**

facilitated by the interaction between an agent and the environment, which can also be referred to as the system. The agent gains rewards from interacting with the environment, and with every action taken, the agent learns the optimum policy of actions so as to maximise the expected cumulative reward and achieve the objective [27].

Since the decision making often results in randomness and uncertainty, reinforcement learning problems are generally modelled as Markov Decision Processes (MDP), which outlines that the effects of an action taken in the current state of the environment depend solely on that state and not on prior history. The MDP model contains:

- A set of possible system states S
- A set of probabilistic transition from current state s_t to the next state s_t', on the action a_t
- A set of possible actions A
- A reward function R for the action a_t on state s_t

Reinforcement learning can be characterised by a method of trial and error in which an agent observes the current state s_t of the environment at time t in order to take an action a_t and transition into a new state. The end of each interaction is marked by the agent receiving a reward r_t, which represents a numerical value that the agent aims to maximise by optimising its decision making in the long run.

The key MDP parameters for the F-RAN system are defined as follows:

- **State:** The current system state s_t is determined by the data processing requirements of users and the state of resource availability in the fog network. The system state at time slot t is defined as $s_t = [v, e, t, R_i, R_a, R_x] \subseteq S$, where v denotes the sum of user requests, e represents the sum of request arrival rate, R_a is the percentage of resource allocation, R_x is the percentage of allocated resources currently unused, R_i is the sum of minimum allocation requirement, and t defines the sum of maximum delay requirement
- **Action:** The action at time instant t is defined as $a_t = \{upscape, downscale, no\ operation\}$
- **Reward:** The reward is determined by the link delay D_{ij}, packet drop ratio P_i and resource allocation ratio R_a and resource utilization R_u. r_t is then defined by:

$$r_t = \begin{cases} -100 & \text{if } R_a \leq R_{min}; \\ \alpha R_u - (\beta D_{ij} + \theta P_j) & \text{if } R_a > R_{min}. \end{cases} \tag{14}$$

where α, β and θ are constants that adjust the influence of R_u, D_{ij} and P_i on the overall reward, and R_{min} is the threshold for minimum resource allocation. The objective of the reward function is to encourage high virtual resource utilisation while penalising nodes for dropping packets and links for having a high delay. A punitive reward of -100 to R_a below R_{min} has also been assigned to ensure that this is the minimum allocation to a virtual resource and therefore avoid adverse effects to QoS in cases of fast changes from very low to high virtual network (VN) loading.

- **Next state:** A node's change of state to a new state will be dictated by the predicted number of expected requests at the next time slot $t + 1$. In order to predict the expected request arrival rate (δ_n), linear regression statistical modelling is employed. The general form of the linear regression model is given by the equation:

$$Y_{t+1} = aX_t + b \tag{15}$$

where X_t denotes the sample service request, Y_{t+1} is the number of expected service requests, and t is the time value when the request was taken. The values of a and b can be calculated by solving the linear regression equation as given by (16) and (17) below:

$$a = \frac{\sum {X_t}^2 \sum Y_t - \sum X_t \sum X_t Y_t}{n \sum {X_t}^2 - (\sum X_t)^2} \tag{16}$$

$$b = \frac{n \sum X_t \sum Y_t - \sum X_t \sum Y_t}{n \sum {X_t}^2 - (\sum X_t)^2} \tag{17}$$

where n is the total number of service requests received.

B. PROPOSED RESOURCE ALLOCATION MECHANISM

To describe the procedure of the proposed method, the RL agent evaluates the number of requests that arrived at a time t and the amount of free resources in fog resource pool in order to determine the state of the current system. The state can be classified as under-utilised or over-utilised, with the former indicating that there is a surplus of resources that are available and thus must be released back to the resource pool by means of a downscaling operation. Alternatively, a classification of the current state as over-utilised dictates that an upscaling operation should be initiated. In neither of the two cases, the system resumes its normal operation.

The learning agent then determines whether or not the next state of the system will be under-utilised or over-utilised by taking into account the predicted number of expected service requests at time $t + 1$. Based on the calculated future availability and the requirement of resources, the agent then makes a decision on the appropriate action to take, whether to perform an upscale, downscale, or no scaling operation.

The described state-action mapping for the scaling decision is shown in Table 1.

If the entire environment model is known, a MDP problem can be easily resolved by some dynamic programming methods, such as value iteration and policy iteration. However,

TABLE 1. State-action mapping.

Time	State	Action
t: $v > f$ $t+1$: $\delta > e$	Over-utilisation	Upscale
t: $v = f$ $t+1$: $\delta = e$	Normal operation	No scaling operation
t: $v < f$ $t+1$: $\delta < e$	Under-utilisation	Downscale

in most cases, the state transition probability function and reward function are not known in advance. Q-learning [28] is a model-free reinforcement learning algorithm which can be used to find optimal policies by learning from previous decision-making experiences. The term model free refers to an algorithmic technique that does not need a prior trained model to take dynamic decisions. It does not rely on complete *a priori* knowledge of the environment. Following the basic idea of reinforcement learning, agents constantly perform actions in different states and then observe state transitions and relevant rewards.

Among several RL techniques, Q-learning requires low computational resources for its implementation and does not require the knowledge of the model of the environment, thus being a suitable learning technique for the resource-constrained fog nodes [29]. Furthermore, Q-learning has been used extensively to address resource allocation problems [30], thus being a suitable learning technique for the problem.

The Q-learning algorithm is expressed by the Q-function $Q(s, a)$ where at time t an action a_t is taken on the current state s_t which will lead to the next state s_{t+1}, and $\gamma \subseteq [0, 1]$ is the discount factor which describes how much future reward affects current decision. It is used to finitely evaluate the overall expected reward for an infinite sequence of decisions. The Q-function is then updated by the Bellman equation (18):

$$Q(s_t, a_t) = r(s_t, a_t) + \gamma \max a_{t+1} Q(s_{t+1}, a_{t+1}) \tag{18}$$

The pseudocode illustrating procedures of the proposed autonomous RL-based resource allocation algorithm is presented in Algorithm 1. The algorithm was implemented using the Open AI Gym toolkit by constructing an environment to record and store the Q-table values as a matrix.

Algorithm 1 Autonomous Resource Allocation

Initialise number of fog nodes
for every time step t **do**
  for each fog node **do**
    Read total number of requests
    for each request **do**
      Collect processing requirements
      Read resource availability
      Collect the request arrival rate
      Calculate expected number of service
      requests at time t + 1 using (Eq.: 15)
    end for
    Initialise Q-values table of pairs (s, a) by zero
    Observe the current state s_t
    Choose an action a_t from the set of actions
    defined for that state s_t in the Q-table
    Perform the action a_t, receive the feedback
    reward r_{t+1} to reach the next state s_{t+1}
    Update the Q-value table using (Eq.: 18)
    $s_t = s_{t+1}$
  end for
  t = t + 1
end for

V. SIMULATION SETUP

A. PARAMETER DESCRIPTION

The simulation environment was created using 5G K network simulator, 5G K-SimNet [31]. 5G K-SimNet is an open source ns3-based network simulator for evaluating end-to-end performance of 5G systems. Its key elements for 5G include support for 5G New Radio based on mmWave, 5G core, multi-connectivity, SDN, and NFV modules.

As part of modelling a 5G network, a smart farm use case is considered. In smart farming, various sensing technologies are deployed across the field for the provision of data to be processed and implemented as need be in order to enable farmers to monitor and optimize crop yield while adapting to changing environmental factors [32]. The idea behind smart farming is to leverage real-time connectivity to enable machine-to-machine communication between farm equipment and other machines on the field. Thus, making 5G and fog computing technologies suitable enablers for the use case.

With respect to 5G NR, the performance objectives for smart farm sensing and monitoring, which is classified as a massive machine-type communications (mMTC) application, are identified below [30], [33]:

1) Ultra-low complexity and low-cost IoT devices and networks.
2) Latency of 10 seconds or less on the uplink to deliver a 20-byte application layer packet measured at 164 dB Maximum Coupling Loss (MCL) or 21 dB of gain.
3) Connection density of 1 million devices per square km in an urban environment.

In the simulation, UEs generate data after regular time interval and transmit it to fog nodes for processing. Since the output after processing is usually small, the simulation only considers the uplink communication for the environment. The simulation was executed on an Intel ® Core (TM) i7 CPU at 2.70 GHz, with 8 GB of RAM, a Linux Ubuntu 16.04 operating system, 5G K-SimNet version 1.2, and Open AI Gym version 0.17.1.

A LTE eNB is deployed as the master node and a NR gNB as the secondary node, with UEs using the dual radio interfaces of both LTE and NR for connectivity. 142 UEs are connected to the RAN, and uplink packets are generated

TABLE 2. Simulation parameter settings.

Parameter	Value
Configuration parameters	
LTE bandwidth	20 MHz
LTE link capacity	75 Mbps
LTE carrier frequency	1800 MHz
mmWave bandwidth	2.16 GHz
mmWave carrier frequency	60 GHz
X2 data rate	10 Gb/s
X2 link delay	50 ms
gNB transmission power	46 dBm
eNB transmission power	23 dBm
Application parameters	
Transport layer protocol	TCP
Number of nodes	2
Simulation time	300 s
Number of simulation runs	100
Confidence interval	0.95
SDN parameters	
Inter-switch data rate	10 Mbps
Switch-GW data rate	100 Mbps
Switch-gNB data rate	100 Mbps
Virtualisation parameters	
Scaling ratio	0.25
VM provisioning delay	0 ms

continuously throughout the simulation time, which is set to 300s. The gNB is also connected to the SDN network consisting of an OpenFlow controller and OpenFlow switches.

The parameters used in the simulation are listed in Table 2.

B. PERFORMANCE METRICS

The following metrics were computed for the purpose of performance analysis of the proposed algorithm:

- CPU utilisation (%)
- Virtual link utilisation (%)
- Latency (ms)
- Cost efficiency (%)

VI. RESULTS AND DISCUSSION

The performance evaluation through simulation modelling seeks to determine the efficacy of the proposed reinforcement learning algorithm regarding resource allocation in a 5G F-RAN architecture. This section describes the resource allocation mechanisms used for comparison then presents the performance results by measuring the proposed algorithm with the systems described.

A. DESCRIPTION OF OTHER RESOURCE ALLOCATION TECHNIQUES

The proposed reinforcement learning framework in this paper implementing a proactive auto-scaling algorithm based on Q-learning will be referred to as System I. The work in [34] described several RL-based methods for resource allocation in FRAN architectures. The algorithm based on SARSA will be referred to as System II, while the Monte Carlo mechanism is System III. The authors of [35] described a dynamic resource allocation framework for an NFV-enabled mobile fog cloud. The proposed framework consists of a fast heuristic-based incremental allocation mechanism that dynamically performs resource allocation and a re-optimisation algorithm that periodically adjusts allocation over time. An offline algorithm estimates the desired response time with minimum resources, and the auto-scaling and load-balancing algorithm makes provision for workload variations. When the capacity violation detection algorithm identifies a failure of the auto-scaling mechanism, a network latency constraint greedy algorithm initialises an NFV-enabled edge node to cope with the failure. This system is referred to as System IV.

The summary of the various systems is provided in Table 3.

TABLE 3. Summary of resource management systems.

Name	Description
System I	The proposed system implementing a Q-learning algorithm
System II	An alternative system implementing a SARSA algorithm
System III	An alternative system implementing a Monte Carlo algorithm
System IV	An alternative system implementing a reactive auto-scaling and load-balancing algorithm.

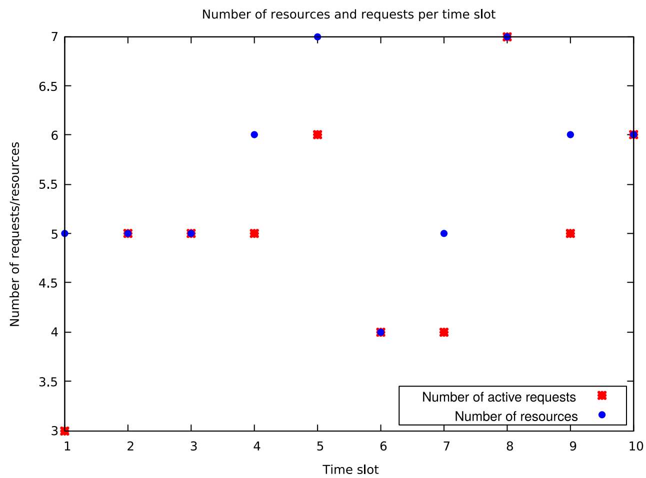

FIGURE 3. Relationship between the number of active requests and resources.

B. RESULTS

The relationship between the number of active requests that approximates demand and capacity represented by the number of compute resources in the system is presented for the proposed Q-learning algorithm in Fig. 3. The data plotted was measured in every 30 second interval for the duration of the simulation period (300 s). In every time slot, the number of resources is either equal to or slightly higher than the number of active requests. This demonstrates the effectiveness of the proposed Q-learning algorithm because every increase or decrease in the number of active requests is accompanied by a corresponding increase or decrease in the number of resources.

Fig. 4 shows the average amount of data transferred by varying the number of connected users in the proposed architecture, compared with the cloud-only model. It is observed

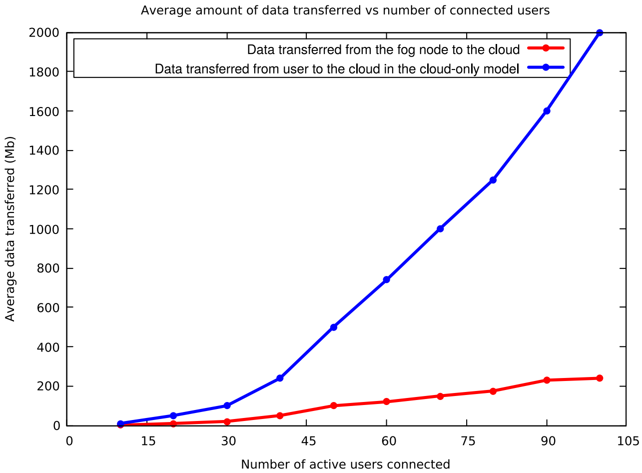

FIGURE 4. **Average data transmitted vs number of connected users.**

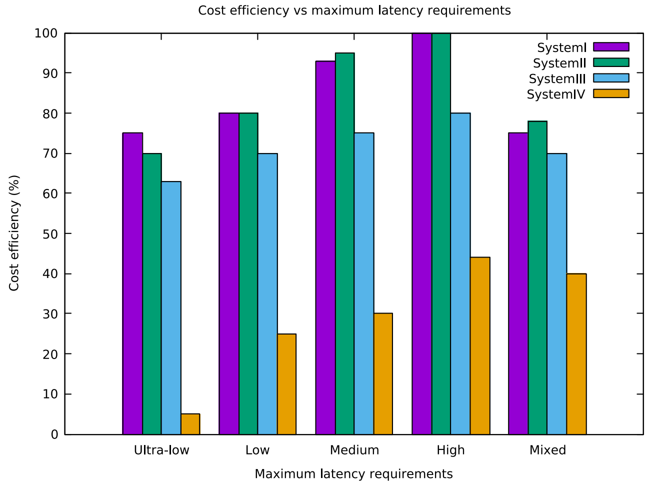

FIGURE 5. **Impact of latency requirements on cost efficiency.**

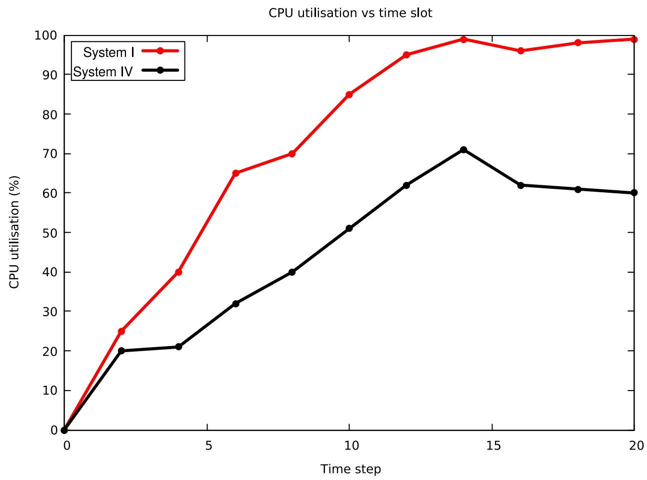

FIGURE 6. **CPU utilisation comparison vs time slot.**

that using the proposed resource management architecture, a significant reduction in the amount of data transferred between users and the cloud can be achieved. On average, the amount of data transferred between the fog node and the cloud server is reduced by up to 90%, which is encouraging particularly for mMTC applications which are characterised by a large volume of data generated by connected sensors. The proposed reinforcement learning model facilitates data processing closer to the users at the fog nodes such that very little traffic is transmitted beyond the local network. This is promising for applications in underserved communities.

In order to take into account the resource utilisation of the fog nodes, cost efficiency is measured, which quantifies the percentage of users who receive their services within the services' latency requirements. The maximum tolerable latency requirements are categorised as ultra-low (<1 ms), low (10 ms), medium (100 ms), high (150 ms), and mixed (randomly selected between 0 and 150 ms). Fig. 5 illustrates the impact of maximum tolerable latency requirement on cost efficiency. The graph shows that extremely strict network latency requirements are less cost effective than more tolerant latencies. Systems I, II and III achieves a cost efficiency above 50% for all latency requirements, while System IV achieves a maximum of 45% cost efficiency for a workload with lenient latency requirements. Therefore, it can be concluded that dynamic auto- scaling is more efficient than the fixed threshold counterpart. System I appears to utilise resources more efficiently and outperforms all three systems for ultra-low latency requirements, however System II is comparable for low latency requirements. For flexible latency requirements, both System I and System II achieve an optimal operational cost where the efficiency is equal to 100 percent.

Resource utilisation of the proposed Q-learning algorithm (System I) is measured against the reactive auto-scaling approach where VMs provisioning is performed based on a fixed scaling threshold (System IV). The CPU utilisation of fog nodes is measured with every time slot, as illustrated in Fig. 6. The proposed reinforcement learning-based algorithm far surpasses the reactive approach in terms of CPU usage. This is because in the former, resources are dynamically allocated based on the actual traffic demand, with unused resources being released back into the resource pool to be reused by other VMs. The inferior performance of the dynamic Q-learning approach can be attributed to the initial learning period of the agent. In the early stages of the simulation when the agent is still learning, fog nodes are allocated less resources than their demand. However, the algorithm progressively learns the Q-function, updating it only for the visited states if and only when visited.

As shown in Fig. 7, the proposed Q-learning mechanism performed better than the reactive auto-scaling approach in terms of link utilisation. Therefore, the proactive approach achieves better link efficiency because the virtual links in the proactive Q-learning utilise more bandwidth than the links in their reactive counterparts.

The measured CPU utilisation for the RL-based systems is shown in Fig. 8. In all three systems, resources are dynamically allocated based on the actual traffic demand, with

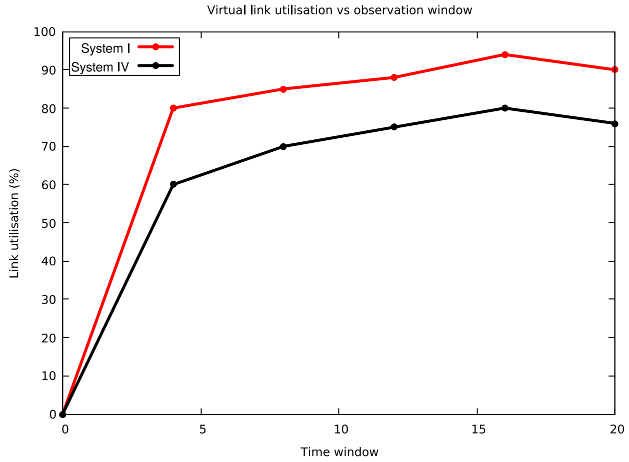

FIGURE 7. Virtual link utilisation comparison.

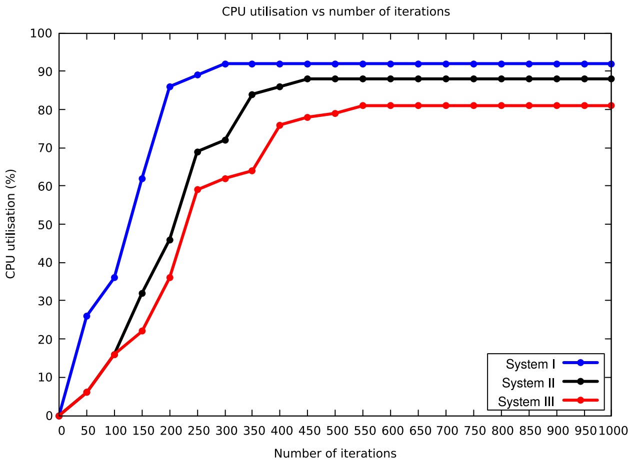

FIGURE 8. CPU utilisation comparison vs number of iterations.

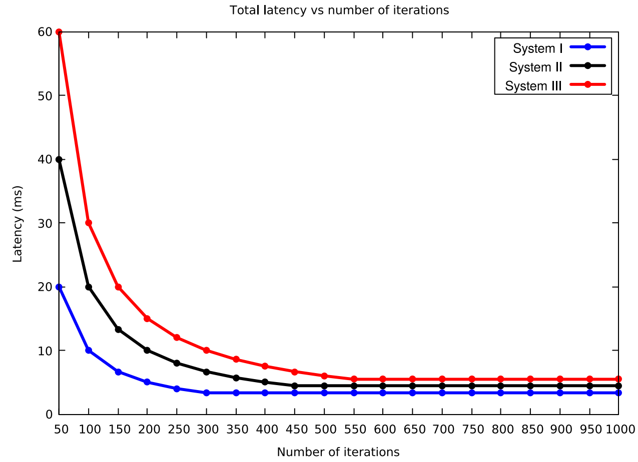

FIGURE 9. Sum of latency comparison of RL systems.

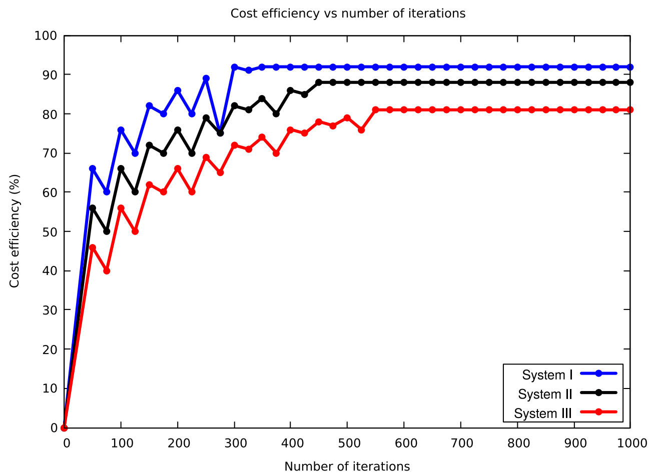

FIGURE 10. Cost efficiency comparison of RL systems.

unused resources being released back into the resource pool to be reused by other VMs. CPU utilisation increases with the rise in the number of iterations, with System I achieving the highest CPU utilisation. System I obtains the highest maximum CPU utilisation earlier than the other systems.

The graph in Fig. 9 shows the sum of latency experienced by users in the RL-based systems. The general pattern is marked by an exponential decrease in latency in the initial stages of training until an equilibrium is reached. As illustrated, System I converges to the minimum total latency of 3 ms after 300 iterations, while System II reaches 4,4 ms after 450 iterations and System III requires 550 iterations to obtain a minimum latency of 5,4 ms. The maximum latencies observed for Systems I, II and III are 20 ms, 40 ms and 60 ms, respectively.

The percentage of users who receive their services within the application's latency requirements, referred to as cost efficiency, was also measured for a system in which the latency requirements are random integers between zero and 150 ms. The cost efficiency observed, illustrated in Fig. 10, exhibits poor performance in the initial stages and converges to optimal values. The beginning of the training period marks the agent's initial learning period, thus the curve of cost efficiency is at a state of constant fluctuation. System I achieves the highest maximum efficiency of 92% after 300 iterations, while System II and System III achieve a maximum efficiency of 88% and 81% after 450 and 550 iterations, respectively.

C. DISCUSSION

The goal of the proposed Q-learning algorithm is to minimise the total end- to-end latency experienced by users through computation resource allocation while enforcing the maximum tolerable latency requirement constraint. It has been demonstrated that the proposed framework always manages to allocate sufficient resources in time to guarantee continuous satisfaction of applications' low latency requirements under dynamic workloads. The amount of data transferred is an important metric to gauge the communication frequency between the fog nodes and the remote cloud server. Reducing the amount of data transferred between fog nodes and the cloud also decreases the transmission delay, which has an impact on end-to-end latency. Furthermore, reduced frequency of communication between fog nodes and cloud servers reduces the propagation delay, since the distance between users and fog nodes is much shorter than the distance to the remote cloud and therefore there is a fewer number of hops for packets to travel. Given that round-trip latency proportionately declines when the number of hops and packet size drops [36], one can deduce that the proposed algorithm leads to lower latency. This is supported by the measurements of latency against varying traffic loads, which demonstrated that the proposed algorithm exhibits the minimum latency. The proposed Q-learning algorithm achieves minimum latency through computation resource allocation while ensuring maximum CPU utilisation and maximum link utilisation, compared with the reactive auto-scaling counterpart. This serves as a demonstration of the potential of machine learning capabilities in 5G F-RAN architectures for resource management. In comparison with other reinforcement learning systems, namely SARSA and Monte Carlo, the proposed Q-learning algorithm achieved a higher percentage of users who receive their services within the application's latency requirements. Furthermore, the proposed Q-learning algorithm converges faster than the other systems.

VII. CONCLUSION

Conventional legacy approaches to system management involve algorithms that are programmed to work according to a predetermined case-based reasoning method and gradually fail when there are unpredictable changes in the workload. In the case of resource constrained, heterogeneous and dynamic resources, traditional approaches are insufficient. This paper proposed reinforcement learning as a solution and devised an algorithm for dynamic and autonomous resource allocation in 5G F-RAN architectures based on Q-learning, as a means to ensure minimum total end-to-end latency experienced by users through computation resource allocation while enforcing the maximum tolerable latency requirement constraint. The results showed significant improvements in performance, including reduced end-to-end delay for applications with ultra-low latency requirements. The improvement in latency is particularly significant because many applications in which fog computing is considered are time sensitive. The relative difference can improve proportionally as the network scales.

This paper focused on the integration of fog computing, machine learning and 5G technologies as a means to aid 5G deployment in underserved communities. To this end, the benefits of this work are divided into three parts. Firstly, by exploiting the capabilities of the fog computing architecture to configure a 5G network with reduced cost, this paper contributes to advancing the limited body of knowledge about making efforts to deploy 5G in underserved regions of developing countries as a means to bridge the digital divide. Secondly, in comparison to URLLC and eMBB, mMTC applications in 5G are an area that is lesser explored. There is a lack of studies in the domain of utilising enhanced next-generation network features such as 5G New Radio to support deployment scenarios for mMTC services and applications. Therefore, by modelling an mMTC application to measure the performance of the proposed methods, this work makes an effort to validate the envisioned requirements of IoT applications in 5G networks. Finally, there is a limited number of studies that discuss the integration of fog computing, machine learning and 5G. By using machine learning techniques to address the resource allocation problem in 5G F-RAN architectures, this paper has contributed to the area of machine learning applications in fog computing and 5G.

The future work needs to investigate the scalability issue in detail to adequately gauge the impact the proposed system for mMTC applications. Furthermore, it would be interesting to see how the proposed solution performs for eMBB and URLLC applications, like in a network slicing architecture.

REFERENCES

[1] M. Chen, U. Challita, W. Saad, C. Yin, and M. Debbah, "Artificial neural networks-based machine learning for wireless networks: A tutorial," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3039–3071, 4th Quart., 2019, doi: 10.1109/COMST.2019.2926625.

[2] E. K. Markakis, K. Karras, A. Sideris, G. Alexiou, and E. Pallis, "Computing, caching, and communication at the edge: The cornerstone for building a versatile 5G ecosystem," *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 152–157, Nov. 2017.

[3] S. Kitanov, E. Monteiro, and T. Janevski, "5G and the fog—Survey of related technologies and research directions," in *Proc. 18th Medit. Electrotech. Conf. (MELECON)*, Apr. 2016, pp. 1–6.

[4] (Oct. 2017). *IEEE 5G and Beyond Technology Roadmap White Paper*. [Online]. Available: https://futurenetworks.ieee.org/roadmap/roadmap-white-paper.

[5] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for mobile networks—A technology overview," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015, doi: 10.1109/COMST.2014.2355255.

[6] P. Chanclou, A. Pizzinat, Y. Denis, and S. Randazzo, "C-RAN architecture and fronthaul challenges," in *Proc. RAN World*, Dusseldorf, Germany, Jan. 2015.

[7] Z. Zhu, P. Gupta, Q. Wang, S. Kalyanaraman, Y. Lin, and H. Franke, "Virtual base station pool: Towards a wireless network cloud for radio access networks," in *Proc. 8th ACM Int. Conf. Comput. Frontiers CF*, 2011, p. 1, doi: 10.1145/2016604.2016646.

[8] Y.-J. Ku, D.-Y. Lin, C.-F. Lee, P.-J. Hsieh, H.-Y. Wei, C.-T. Chou, and A.-C. Pang, "5G radio access network design with the fog paradigm: Confluence of communications and computing," *IEEE Commun. Mag.*, vol. 55, no. 4, pp. 46–52, Apr. 2017.

[9] Y.-Y. Shih, W.-H. Chung, A.-C. Pang, T.-C. Chiu, and H.-Y. Wei, "Enabling low-latency applications in fog-radio access networks," *IEEE Netw.*, vol. 31, no. 1, pp. 52–58, Jan. 2017.

[10] D. Pouillot, "The dynamics of broadband markets in Europe: Realizing the 2020 digital agenda," *Commun. Strateg.*, vol. 99, p. 183, Oct. 2015.

[11] S. Lavanya, N. M. S. Kumar, S. Thilagam, and S. Sinduja, "Fog computing based radio access network in 5G wireless communications," in *Proc. Int. Conf. Wireless Commun., Signal Process. Netw. (WiSPNET)*, Mar. 2017, pp. 559–563, doi: 10.1109/WiSPNET.2017.8299819.

[12] G. Li, Y. Liu, J. Wu, D. Lin, and S. Zhao, "Methods of resource scheduling based on optimized fuzzy clustering in fog computing," *Sensors*, vol. 19, no. 9, p. 2122, May 2019, doi: 10.3390/s19092122.

[13] Y. Wang, K. Wang, H. Huang, T. Miyazaki, and S. Guo, "Traffic and computation co-offloading with reinforcement learning in fog computing for industrial applications," *IEEE Trans. Ind. Informat.*, vol. 15, no. 2, pp. 976–986, Feb. 2019, doi: 10.1109/TII.2018.2883991.

[14] L. Huang, X. Feng, C. Zhang, L. Qian, and Y. Wu, "Deep reinforcement learning-based joint task offloading and bandwidth allocation for multi-user mobile edge computing," *Digit. Commun. Netw.*, vol. 5, no. 1, pp. 10–17, Feb. 2019, doi: 10.1016/j.dcan.2018.10.003.

[15] Y. Wei, F. R. Yu, M. Song, and Z. Han, "Joint optimization of caching, computing, and radio resources for fog-enabled IoT using natural actor–critic deep reinforcement learning," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2061–2073, Apr. 2019, doi: 10.1109/JIOT.2018.2878435.

[16] Q. D. La, M. V. Ngo, T. Q. Dinh, T. Q. S. Quek, and H. Shin, "Enabling intelligence in fog computing to achieve energy and latency reduction," *Digit. Commun. Netw.*, vol. 5, no. 1, pp. 3–9, Feb. 2019, doi: 10.1016/j.dcan.2018.10.008.

[17] Y. Zhou, M. Peng, S. Yan, and Y. Sun, "Deep reinforcement learning based coded caching scheme in fog radio access networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC Workshops)*, Aug. 2018, pp. 309–313, doi: 10.1109/ICCChinaW.2018.8674478.

[18] G. M. S. Rahman, M. Peng, K. Zhang, and S. Chen, "Radio resource allocation for achieving ultra-low latency in fog radio access networks," *IEEE Access*, vol. 6, pp. 17442–17454, 2018, doi: 10.1109/ACCESS.2018.2805303.

[19] G. M. S. Rahman, M. Peng, S. Yan, and T. Dang, "Learning based joint cache and power allocation in fog radio access networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4401–4411, Apr. 2020, doi: 10.1109/TVT.2020.2975849.

[20] Z. Zhao, S. Bu, T. Zhao, Z. Yin, M. Peng, Z. Ding, and T. Q. S. Quek, "On the design of computation offloading in fog radio access networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 7, pp. 7136–7149, Jul. 2019, doi: 10.1109/TVT.2019.2919915.

[21] K. Liang, L. Zhao, X. Zhao, Y. Wang, and S. Ou, "Joint resource allocation and coordinated computation offloading for fog radio access networks," *China Commun.*, vol. 13, no. 2, pp. 131–139, 2016, doi: 10.1109/CC.2016.7833467.

[22] L. Ruan, Z. Liu, X. Qiu, Z. Wang, S. Guo, and F. Qi, "Resource allocation and distributed uplink offloading mechanism in fog environment," *J. Commun. Netw.*, vol. 20, no. 3, pp. 247–256, Jun. 2018, doi: 10.1109/JCN.2018.000037.

[23] A. Nassar and Y. Yilmaz, "Resource allocation in fog RAN for heterogeneous IoT environments based on reinforcement learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6, doi: 10.1109/ICC.2019.8761626.

[24] M. Mukherjee, Y. Liu, J. Lloret, L. Guo, R. Matam, and M. Aazam, "Transmission and latency-aware load balancing for fog radio access networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6, doi: 10.1109/GLOCOM.2018.8647580.

[25] E. Casalicchio, D. A. Menascé, and A. Aldhalaan, "Autonomic resource provisioning in cloud systems with availability goals," in *Proc. ACM Cloud Autonomic Comput. Conf. - CAC*, 2013, doi: 10.1145/2494621.2494623.

[26] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine learning for networking: Workflow, advances and opportunities," *IEEE Netw.*, vol. 32, no. 2, pp. 92–99, Mar. 2018.

[27] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3072–3108, 4th Quart., 2019, doi: 10.1109/COMST.2019.2924243.

[28] T. E. Bogale, X. Wang, and L. B. Le, "Machine intelligence techniques for next-generation context-aware wireless networks," in *Proc. ITU Special Issue, Impact Artif. Intell. (AI) Commun. Netw. Services*, vol. 1, 2018. [Online]. Available: https://arXiv:1801.04223

[29] M. Moh and R. Raju, "Machine learning techniques for security of Internet of Things (IoT) and fog computing systems," in *Proc. Int. Conf. High Perform. Comput. Simul. (HPCS)*, Jul. 2018, pp. 709–715, doi: 10.1109/HPCS.2018.00116.

[30] S. K. Sharma and X. Wang, "Toward massive machine type communications in ultra-dense cellular IoT networks: Current issues and machine learning-assisted solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 426–471, 1st Quart., 2020, doi: 10.1109/COMST.2019.2916177.

[31] S. Choi, J. Song, J. Kim, S. Lim, S. Choi, T. T. Kwon, and S. Bahk, "5G K-SimNet: End-to-end performance evaluation of 5G cellular systems," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6, doi: 10.1109/CCNC.2019.8651686.

[32] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, 2015.

[33] R. Ratasuk, N. Mangalvedhe, D. Bhatoolaul, and A. Ghosh, "LTE-M evolution towards 5G massive MTC," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2017, pp. 1–6, doi: 10.1109/GLOCOMW.2017.8269112.

[34] A. T. Nassar and Y. Yilmaz, "Reinforcement learning-based resource allocation in fog RAN for IoT with heterogeneous latency requirements," 2018, *arXiv:1806.04582*. [Online]. Available: http://arxiv.org/abs/1806.04582

[35] B. Yang, W. K. Chai, Z. Xu, K. V. Katsaros, and G. Pavlou, "Cost-efficient NFV-enabled mobile edge-cloud for low latency mobile applications," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 475–488, Mar. 2018, doi: 10.1109/TNSM.2018.2790081.

[36] A. U. Qureshi, "Light weight mobile cloud computing environment for mobile applications," Tech. Rep., 2020.

NOSIPHO N. KHUMALO received the B.Sc. and B.Sc. (Hons) degrees in computer science from the University of Cape Town, South Africa. She is currently pursuing the M.Sc. degree in electrical engineering with the University of the Witwatersrand, South Africa. Since 2018, she has been a Software Engineer with the Spectrum Access and Management Innovation Research Group, Council for Scientific and Industrial Research (CSIR). Her research interests include information and communications technologies for development (ICT4D), the Internet of Things (IoT), and software engineering. She was a recipient of the IEEE Vehicular Technology Society Best Paper Award, in 2019.

OLUTAYO O. OYERINDE (Senior Member, IEEE) received the Ph.D. degree in electronic engineering from the University of Kwazulu–Natal, Durban, South Africa, in 2011. Since 2013, he has been with the School of Electrical and Information Engineering, University of the Witwatersrand at Johannesburg, South Africa, where he is currently an Associate Professor. He is also a National Research Foundation (NRF) Rated Scientist, a Registered Professional Engineer (Pr.Eng.) with the Engineering Council of South Africa (ECSA), a Registered Engineer (R.Eng) with COREN, and a Corporate Member of NSE among others. He is an Associate Editor of the IEEE ACCESS journal and an Editorial Board Member of the *International Journal of Sensors, Wireless Communications and Control*. His current research interests include wireless communications with specific interests in 5G and beyond 5G technologies, cognitive radio networks, and signal processing techniques for wireless communication systems.

LUZANGO MFUPE (Senior Member, IEEE) received the Ph.D. degree in telecommunications. He is currently a Principal Researcher and an Innovator with the Council for Scientific and Industrial Research (CSIR), South Africa, and an Adjunct Professor with the Nelson Mandela African Institution of Science and Technology (NM-AIST), Tanzania. He is a Technical Leader of the CSIR's TVWS Geo-location spectrum database (GLSD) initiatives. His main research interests include futuristic fixed, airborne, and mobile wireless network technologies such as TVWS, 5G and beyond, and innovative flexible radio frequency (RF) spectrum management policy schemes such as machine learning and dynamic spectrum access techniques. He is the Vice-Chairman of IEEE DySPAN-SC 1900.6 Standardisation Work Group. He is an Organizing Chair of the IEEE Africon 2021 Flagship Conference to be held in Arusha, Tanzania.