Device Association for RAN Slicing based on Hybrid Federated Deep Reinforcement Learning

Yi-Jing Liu, Gang Feng, *Senior Member, IEEE*, Yao Sun, Shuang Qin, *Member, IEEE*, and Ying-Chang Liang, *Fellow, IEEE*

***Abstract*—Network slicing (NS) has been widely identified as a key architectural technology for 5G-and-beyond systems by supporting divergent requirements in a sustainable way. In radio access network (RAN) slicing, due to the device-base station (BS)-NS three layer association relationship, device association (including access control and handoff management) becomes an essential yet challenging issue. With the increasing concerns on stringent data security and device privacy, exploiting local resources to solve device association problem while enforcing data security and device privacy becomes attractive. Fortunately, recently emerging federated learning (FL), a distributed learning paradigm with data protection, provides an effective tool to address this type of issues in mobile networks. In this paper, we propose an efficient device association scheme for RAN slicing by exploiting a hybrid FL reinforcement learning (HDRL) framework, with the aim to improve network throughput while reducing handoff cost. In our proposed framework, individual smart devices train a local machine learning model based on local data and then send the model features to the serving BS/encrypted party for aggregation, so as to efficiently reduce bandwidth consumption for learning while enforcing data privacy. Specifically, we use deep reinforcement learning to train the local model on smart devices under a hybrid FL framework, where horizontal FL is employed for parameter aggregation on BS, while vertical FL is employed for NS/BS pair selection aggregation on the encrypted party. Numerical results show that the proposed HDRL scheme can achieve significant performance gain in terms of network throughput and communication efficiency in comparison with some state-of-the-art solutions.**

***Index Terms*—RAN Slicing, Device Association, Federated Learning, Deep Reinforcement Learning**

I. Introduction

It is widely acknowledged that network slicing (NS) is one of the most vital architectural technologies for 5G-and-beyond systems. In order to support various applications with diverse quality of service (QoS) requirements in different communication scenarios, *e.g.*, enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low-latency communications (URLLC), multiple isolated network slices (NSs) can be designed, deployed, customized, and optimized on a common physical network infrastructure [1]–[3]. The NS based networks (virtualized networks) can provide tailored services efficiently and flexibly to meet the specific needs of applications and corresponding Service Level Agreement. However, driven by the rapidly growing wireless applications with diversified service requirements, how to identify and classify service flows for accommodation by appropriate application-specific NS (*i.e.*, device association including access control and handoff management) is still a challenging issue, especially in radio access network (RAN) domain.

In RAN slicing, device association and relevant resource allocation are fundamentally distinct from that in conventional mobile networks because of the introduction of NS [4], [5]. On one hand, NSs are logically virtualized and isolated over shared physical networks [4], [5]. Thus, both physical and virtual resource, *e.g.* computing, network, storage, radio, access hardware, and virtual network functions, should be considered to form a function chain for a specific service [4]. On the other hand, to meet the service requirements, a device needs to select an appropriate NS which may cover multiple access points (APs), i.e., base stations (BSs) [5]. Therefore, in virtualized networks, device association inherently includes NS selection, BS association, and associated resource allocation issues, which should be addressed jointly to improve resource utilization while guaranteeing service quality. Moreover, due to the dynamic nature of network environments, the computational complexity incurred by searching the optimal solution could be too high and the environmental changes may not be accurately described in some complex and dynamic scenarios. Fortunately, recently emerging reinforcement learning (RL) can be exploited to solve such sequential decision problems under complex network environments. By using RL, devices can continuously interact with the environments and thus obtain an optimal solution by using a trial and error learning process. Although RL works well in decision-making scenarios, the effectiveness of RL diminishes as the size of the state-action space becomes large [6], [7]. Then, deep reinforcement learning (DRL) emerges as a good alternative to solve the decision-making problem in the wireless system with a large size of data [6].

With the dramatic growth of the heterogeneous data from geographically distributed devices, traditional centralized DRL algorithms may not be feasible in practice since they require the data to be transferred and processed in a central entity, which definitely causes large latency in uploading a huge

Y. Liu, G. Feng, Y. Sun, S. Qin, and Y.C. Liang are with the National Key Lab on Communications, University of Electronic Science and Technology of China, Chengdu 611731, and also with the Center for Intelligent Networking and Communications (CINC), University of Electronic Science and Technology of China, Chengdu 611731, China. G. Feng is the corresponding author (e-mail: fenggang@uestc.edu.cn; phone: +862861830292; fax: +862861830292).

This work was supported by Development Program in Key Areas of Guangdong Province under Grant 2018B010114001, and the National Science Foundation of China under Grant 62071091.

amount of raw data and consumes certain precious network bandwidth [8], [9]. As a result, decentralized DRL algorithms that exploit local data are much more appealing. Furthermore, in light of the increasingly stringent data security and device privacy concerns, an emerging decentralized approach, federated learning (FL) [10], is introduced. FL trains non-independently identically distribution and unbalanced data locally at individual devices and exploits the collaboration of the devices. Specifically, FL is classified into horizontally FL (hFL), vertically FL (vFL), and federated transfer learning based on how data is distributed among various devices in the feature and sample space [10], [11]. Most of the existing related work focuses on hFL to share the sample space or vFL to share the feature space, such as [9], [12], [13]. Indeed, in order to reduce the amount of required training samples and/or make more precise decisions, combing hFL and vFL, called hybrid FL, is intuitively advantageous [10].

In this paper, we propose an intelligent device association scheme for RAN slicing, called hybrid federated deep reinforcement learning (HDRL) scheme, with the aim to improve network throughput while reducing handoff cost. Considering the large state-action space and the diversity of services, HDRL is designed to consist of two layer model aggregations: 1) Horizontal aggregation: for the same type of services (*e.g.*, eMBB services), we aggregate the parameters of local DRL model on BSs to share the similar samples; 2) Vertical aggregation: for the services of different types (*e.g.*, eMBB and URLLC services), we aggregate the access features of local DRL model on the third encrypted party in vFL [10], where we use Shapley value [11] to evaluate aggregated access features. Numerical results show that in the typical scenarios, our proposed HDRL scheme for device association significantly outperforms the traditional solutions in terms of network throughput and communication efficiency.

The main contributions of this work can be summarized as follows:

(1) We combine DRL and hFL to train distributed data over smart devices, with the aim to retain the privacy of local data.
(2) We calculate Shapley values to evaluate the importance of different global access features [10], [11] and promote collaboration between devices.
(3) We propose to exploit two levels of aggregation for device association problem. Specifically, one is for the same type of services to aggregate the local parameter models to share the similar samples. Another one is for the different types of services to aggregate access features to make a global optimal decision on NS and BS selection.

In the rest of this paper, we begin with an overview of related work in Section II. Then we present the system model and problem formulation in Section III and Section IV respectively. In Section V, HDRL is presented to solve the device association problem of RAN slicing. Finally, we present the numerical results in Section VI and conclude the paper in Section VII.

II. Related Work

In recent years, there has been a large body of research work on resource management in the virtualized core network (CN), such as [14]–[18], etc. However, considering that end-to-end slices span both CN and RAN, in order to improve the efficiency and resource utilization of the virtualized networks, RAN slicing should also be considered to provide specific services for smart devices through end-to-end NSs. The authors of [19]–[23] pointed out that device association is one of the key issues in virtualized networks since that device association determining whether a device is associated with a certain NS via a specific BS, plays a crucial role for load balancing, radio spectrum efficiency, and network efficiency [20]. Moreover, device association in sliced mobile networks is fundamentally distinct from that in conventional mobile networks because of the device-BS-NS three layer association relationship. Thus existing access/handoff control schemes for traditional mobile network cannot be applicable to virtualized mobile networks [20], [21]. Specifically, a joint optimization of NS and BS selection for a device with specific QoS requirements should be addressed [19]–[23]. In addition, the handoff under device-BS-NS three layer relationship is different from traditional reference signal received power (RSRP)-based handoff mechanisms. Both the handoff types (*i.e.*, switching NS only, switching BS only, switching NS and BS) and the RSRP of BS should be taken into account to guarantee the service quality [20], [21], [23].

Indeed, there are existing some investigations focusing on access control or slice association in RAN slicing, such as [3], [21], [23]–[25]. In [3], the authors proposed a framework to investigate access control, with the aim of minimizing wireless bandwidth consumption while guaranteeing QoS of users. In [21], the authors proposed a unified framework for RAN slicing (including user admissibility, slice association, and bandwidth allocation) with the aim of maximizing resource utilization. The authors of [23] resorted to a genetic algorithm to investigate NS selection, with the aim to improve network resource utilization. In [24], joint access control and power allocation were addressed in an Open-RAN system, where the problem of wireless link scheduling was formulated as maximizing the energy efficiency and minimizing power consumption and the cost of physical resources. The authors of [25] proposed an integrated slice allocation and admission control scheme, with the aim to improve network throughput of the whole system. However, these existing approaches in [3], [21], [23]–[25] which tackled similar problems under the device-BS-NS three layer association relationship did not consider data security and device privacy. In addition, the authors of [3], [23], [24] did not consider the handoff management and the authors of [25] only considered the inter-slice handoff management. Furthermore, the authors of [3], [21], [24], [25] applied the static optimization algorithms and the authors of [23] applied the static heuristic algorithm (*i.e.*, genetic algorithm). Both the static optimization algorithms and heuristic algorithms may be inappropriate for device association in complex and dynamic network scenarios as the computational complexity could be prohibitively high to

retain the optimality by constantly performing the optimization algorithm in dynamic network scenarios.

Considering the uncertainty of access conditions and user mobility, some researchers proposed to optimize the long-term network performance by using conventional RL algorithms, such as actor-critic (A3C) and DRL. In [26], the authors designed an on-line scheme based on DRL to accomplish the optimal resource orchestration in the virtualized network. The authors of [27] exploited a collaborative A3C learning framework to manage the resources in RAN slicing. The algorithms in [26] and [27] consume certain network bandwidth resources to transmit training data. Moreover, both of them did not consider the data security and device privacy, which is highly emphasized and concerned in 5G-and-beyond systems [2], [22], [28].

Recently, in order to enforce data security and device privacy, a novel and safe distributed machine learning framework, FL, has been introduced into wireless networks [9], [12], [13]. Specifically, the authors of [9] tried to bridge the trade-off gaps by formulating FL over wireless network as an optimization problem. The authors of [12] formulated the joint wireless resource allocation and user selection as an optimization problem with the aim to minimize an FL loss function that captures the optimal transmit power. In [13], the authors employed FL scheme to transfer the control and responsibility from the centralized controller to individual user devices. However, the authors of [9], [12], [13] only focused on hFL to share the sample space and did not consider the diversity of services. Moreover, the authors of [11] used Shapley values in vFL to calculate the importance of features, opening the door for investigating hybrid FL and crediting allocation in the context of FL in terms of diversified service types. To the best of our knowledge, device association over RAN slices based on FL is still not considered in current researches.

III. System Model

A. Network Model

We consider a scenario where the virtualized network is built upon a Software Define Network/Network Function Virtualization -enabled 5G network infrastructure, which is composed of CN and RAN. As shown in Fig. 1, the access and mobility management function (AMF) is responsible for the connectivity and mobility management for associating devices with slices [29]. The selection of network slice instances for a device is triggered by the first contacted AMF. When the location of a device changes, the initially selected AMF entity may be changed to receive services, to enable mobility tracking and enable reachability. Specifically, if the AMF entity can serve the single network slice selection assistance information (S-NSSAI), the AMF entity remains the serving AMF for the device. Otherwise, the network slice selection function (NSSF) which is responsible for selecting the set of network slice instances (NSIs) and AMF set (or candidate AMF) to serve devices [29], will select the NSIs and determine the target AMF set to serve the devices. More details about network functionalities can be found in [26]. In addition, some network functions can be shared among multiple slices, while others are slice-specific. For example, in CN domain, AMF and NSSF can be shared among multiple slices, while UPF, NEF, and UDM are slice-specific [29], [30]. In cloud-RAN domain, DU and RU could be shared if the functions (*e.g.*, radio functions, baseband processing functions) are implemented by physical devices [31]. In general, CU could be slice-specific because it realizes the "packet processing functions" as virtualized network functions [31].

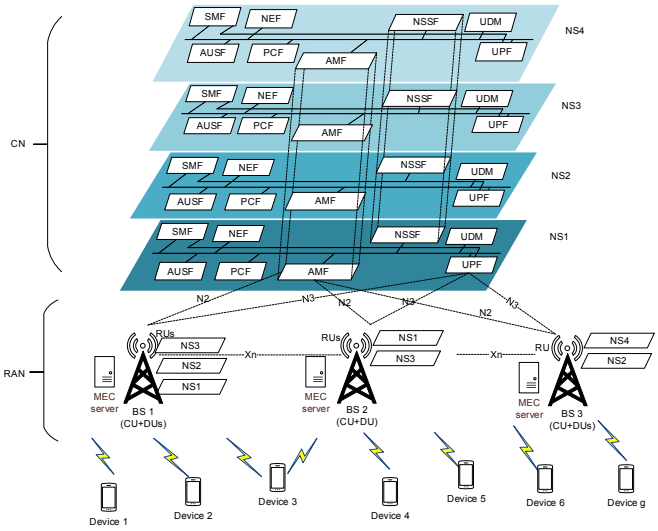


Fig. 1. The NS based mobile network model.

B. RAN slicing

We consider a multi-NSs and multi-BSs RAN slicing scenario, as shown as Fig.1, where an operator deploys a slice on multiple BSs (not all BSs). In other words, a NS may expand multiple BSs and a BS may be covered by multiple NSs. When a device accesses the mobile network or experiences a handoff, both BS and NS may need to be selected/reselected for provision seamless service while meeting the QoS requirements of the device. Specifically, for serving mobile devices, the change of device association is only the change of serving BS. For the case that a device moves out of the coverage of a specific slice, two methods can be used to guarantee the connections. One is to expand the coverage of the current serving slice by deploying it on more BSs. Another one is to change the device association to an exiting slice which can provide the similar service thus to fulfill the QoS requirements. In addition, if the operator knows that a service provided by some slices, it must cover a specific region. If the intended NS is not deployed on a specific BS or the QoS of a device cannot be guaranteed by a specific BS, the device can access the slice via other BSs with the slice deployed in this region.

Let $\mathcal{B}$, $\mathcal{N}$, and $\mathcal{D}$ denote the set of BSs, NSs, and devices, respectively. For a specific BS k, we use $\mathcal{N}_k = \{j, ..., g\}$ to represent the set of NSs which are supported by it. For a specific NS j, we use a four-tuple $\left(R_j, T_j, \Omega_j, \vec{W}_j\right)$ to represent the state where R_j and T_j denote the minimal transmission rate and the maximal latency which are provided by NS j to serve devices. Moreover, Ω_j represents the bandwidth allocated to NS j in CN (including transport network), and $\vec{W}_j$

is a vector, which represents the bandwidth allocation of NS j from all BSs. We assume that the kth element in $\vec{W}_j$ is denoted by $b_{j,k}$, which represents the bandwidth resource allocated to NS j by BS k, where $b_{j,k} = 0$ means BS k is not covered by NS j. For convenience, the frequently used notations are summarized in Table I.

TABLE I
FREQUENTLY USED NOTATIONS

Notation	Definition
$\mathcal{B}$	the set of BSs
$\mathcal{N}$	the set of NSs
$\mathcal{D}$	the set of devices
d_i	the ith device
T	total number of time slots
T_j	the maximal latency provided by NS j
$\mathcal{N}_k$	the set of NSs supported by BS k at t
R_j	the minimal transmission rate provided by NS j
Ω_j	the bandwidth allocated to NS j in CN
$\vec{W}_j$	bandwidth allocation of NS j from all BSs
$b_{j,k}$	the bandwidth allocated to NS j by BS k
$b^t_{j,k}$	the available bandwidth allocated to NS j by BS k during time slot t
u	the total number of smart devices
u_x	the number of devices with service of type x
$\hat{r}^t_i$	the minimum transmission rate of d_i
$\hat{d}^t_i$	the maximum tolerated latency of d_i
$r^{j,k}_{i,t}$	the transmission rate of d_i served by NS j via BS k
$w^{j,k}_{i,t}$	the wireless bandwidth that BS k allocates to d_i served by NS j during time slot t
$\hat{T}^{j,k}_{i,t}$	the transmission delay in RAN of d_i served by NS j via BS k during time slot t
$\hat{T}^{j,k}_{i,t} + T_j$	end-to-end delay
q_i	the volume of flow data generated by d_i
α^{HO}	handoff cost

C. Service Requirements

Since the services required by devices may vary with time, we assume the time is slotted, where the services remain fixed for the duration of one time slot and change from one slot to the next. Slotted time can be regarded as a sampled version of continuous-time which consists of T time slots (fixed time intervals) [32], [33]. During time slot $t \in [1, T]$, we assume that a device connects only one BS and remains connected to the same NS and BS. Let u be the number of devices in the network. For a specific device $d_i \in \mathcal{D}$, its service quality can be described by two metrics: the minimum transmission rate $\hat{r}^t_i$ and the maximum tolerated latency $\hat{d}^t_i$. Therefore, NS j can accommodate d_i only if $R_j \geq \hat{r}^t_i$ and $T_j \leq \hat{d}^t_i$.

Let $r^{j,k}_{i,t}$ be the transmission rate of d_i which is served by NS j via BS k during time slot t, and $w^{j,k}_{i,t}$ be the wireless bandwidth that BS k allocates to d_i which is served by NS j during time slot t (Here $w^{j,k}_{i,t}$ also called consumed radio resources of d_i during time slot t). In this work, the models may affect the absolute value of communication efficiency, but do not invalidate the relative performance enhancement of our proposed policies. Hence, more sophisticated and precise models can be applied here, and then use the proposed algorithm to solve the device association problem. As we focus the device association in the RAN slicing, we assume the delay in CN (T_j) is a constant. The similar assumption is widely used in related studies, such as [3], [21], [34]. Therefore, the end-to-end delay can be calculated as $\hat{T}^{j,k}_{i,t} + T_j$, where $\hat{T}^{j,k}_{i,t} = q_i / r^{j,k}_{i,t}$ is the delay in RAN of d_i served by NS j via BS k and q_i is the volume of flow data generated by d_i. Moreover, we use Shannon theory to define the transmission rate (*i.e.*, $r^{j,k}_{i,t} = w^{j,k}_{i,t} log_2\left(1 + SINR^k_{i,t}\right)$), where $SINR^k_{i,t}$ is the signal-to-interference-plus-noise-ratio (SINR) between d_i and BS k during time slot t. Moreover, $SINR^k_{i,t} = \frac{p^k_{i,t} \cdot G^k_{i,t}}{\sum_{k \in \mathcal{B}, k' \neq k} p^{k'}_{i,t} G^{k'}_{i,t} + \zeta^2}, t \in T$, where $p^k_{i,t}$ represents the transmission power allocated to d_i at BS k, $G^k_{i,t}$ is the channel gain between d_i and BS k, and ζ^2 is the noise power level.

D. Handoff Cost

When the location of a device changes or the service quality of a device cannot be satisfied, a handoff may occur to improve the experience of the user. Once a handoff happens, the device needs to re-select appropriate BS and NS. It is obvious that traditional reference signal received power (RSRP)-based handoff mechanisms [35] are no longer applicable to RAN slicing. Specifically, a device accesses to a NS via a specific BS, forming a three-layer associate relationship device-BS-NS. Therefore, both the service type of NSs and the RSRP of BSs should be taken into account to guarantee the service quality when a handoff occurs. Therefore, unlike the handoff in traditional mobile networks, there are three types of handoff we need to consider: switching NS only, switching BS only, and switching both NS and BS [36]. The amount of signaling data needed for a handoff is different for the three types. For example, switching NS only needs to exchange signaling in the same BS, while switching both NS and BS needs to exchange signaling between different BSs and NSs. Therefore, based on the idea of [36], we define the amount of signaling data for three types of handoff as: 1) q_{NS}, the amount of signaling data needed for switching NS only; 2) q_{BS}, the amount of signaling data needed for switching BS only; 3) q_{N-B}, the amount of signaling data needed for switching both NS and BS; with the relationship $q_{NS} < q_{BS} < q_{N-B}$ [36]. Intuitively, the amount of signaling data needed incurs corresponding signaling overhead in terms of bandwidth consumption for signaling exchange.

Furthermore, due to the bandwidth consumed by service flows and the bandwidth consumed by handoff may not be in the same order of magnitude, we define the handoff cost as

follows [37],

$$\alpha^{\text{HO}} = \begin{cases} \frac{q_{NS}}{w_{NS}}, & \text{if switching NSs only,} \\ \frac{q_{BS}}{w_{BS}}, & \text{if switching BSs only,} \\ \frac{q_{N-B}}{w_{N-B}}, & \text{if switching both NSs and BSs,} \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

where w_{NS} represents the bandwidth consumed by the first type of handoff switching NS only, w_{BS} represents the bandwidth consumed by switching BS only, and w_{N-B} states the bandwidth consumed by switching both BS and NS.

IV. Problem Formulation

A. Problem Statement

Given a set of devices which may require services of different types, we investigate the device association problem under network resource constraints. We define a binary variable $x_{i,t}^{j,k}$ to indicate whether the device d_i is served by NS j via BS k during time slot t or not: $x_{i,t}^{j,k} = 1$ yes and 0 otherwise. Therefore, multiplying the two variables $x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$ in adjacent time slots indicates the handoff decision of d_i from time slot $t-1$ to t, which can be summarized in Table II. Note that if $x_{i,t}^{j,k} x_{i,t-1}^{j',k'} = 0$, we know that device d_i is not served by NS j' via BS k' during time slot $t-1$ or/and device d_i is not served by NS j via BS k during time slot t, and we cannot judge whether a handoff happens. However, it is much easier to judge if a handoff happens when $x_{i,t}^{j,k} x_{i,t-1}^{j',k'} = 1$. As shown in Table II, when $x_{i,t}^{j,k} x_{i,t-1}^{j',k'} = 1$, we can judge whether a handoff happens and derive the handoff types and corresponding handoff cost (*i.e.*, α^{HO}) from following four aspects: 1) $j \neq j'$, $k \neq k'$, switching both BS and NS; 2) $j = j'$, $k \neq k'$, switching BS only; 3) $j \neq j'$, $k = k'$, switching NS only; 4) $j = j'$, $k = k'$, device d_i is served by NS j'/j via BS k'/k during both time slot $t-1$ and t. Thus no handoff happens.

TABLE II
The relationship between handoff and $x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$

$x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$	NSs	BSs	Switching	α^{HO}
1	$j \neq j'$	$k \neq k'$	both BS and NS	$\frac{q_{N-B}}{w_{N-B}}$
1	$j = j'$	$k \neq k'$	BS only	$\frac{q_{BS}}{w_{BS}}$
1	$j \neq j'$	$k = k'$	NS only	$\frac{q_{NS}}{w_{NS}}$
1	$j = j'$	$k = k'$	no handoff	0

Therefore, in order to improve network throughput while reducing handoff cost, we define the communication efficiency of the network during time slot t as follows,

$$e_t = \sum_{i \in \mathcal{D}} (\alpha_{i,t}^{\text{flow}} x_{i,t}^{j,k} - \alpha^{\text{HO}} x_{i,t}^{j,k} x_{i,t-1}^{j',k'}), \forall t \in [0, T]. \tag{2}$$

In equation (2), communication efficiency e_t is a bandwidth metric value representing the bandwidth efficiency minus signaling overhead (which is indeed the "handoff cost"). The bandwidth efficiency $\alpha_{i,t}^{\text{flow}} x_{i,t}^{j,k}$ represents the amount of service data transmitted in unit bandwidth during a time slot. The signaling overhead $\alpha^{\text{HO}} x_{i,t}^{j,k} x_{i,t-1}^{j',k'}$ denotes the amount of signaling data transmitted in unit bandwidth. Moreover, $\alpha_{i,t}^{\text{flow}} = \frac{q_i}{w_{i,t}^{j,k}}$ [37], where q_i represents the service flow data volume of d_i, and $w_{i,t}^{j,k}$ represents the wireless bandwidth that BS k allocates to d_i served by NS j.

In our model, $x_{i,t}^{j,k}$ is a decision variable, which represents the decision on NS and BS selection of d_i. As the device association is indeed a sequential decision problem, we use the long-term communication efficiency of the network as the optimization objective in (3), with the aim to improve network throughput while reducing handoff cost. Therefore, we formulate the device association problem as follows.

$$\max \quad \lim_{T \to +\infty} E[\frac{1}{T} \sum_{t=1}^{T} e_t] \tag{3}$$

$$\text{s.t.} \sum_{k \in \mathcal{B}} \sum_{i \in \mathcal{D}} x_{i,t}^{j,k} r_{i,t}^{j,k} \leq \Omega_j, \forall j \in \mathcal{N}, t \in [0, T] \tag{3.1}$$

$$\sum_{i \in \mathcal{D}} x_{i,t}^{j,k} w_{i,t}^{j,k} \leq b_{j,k}, \forall j \in \mathcal{N}, \forall k \in \mathcal{B}, t \in [0, T] \tag{3.2}$$

$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{B}} x_{i,t}^{j,k} r_{i,t}^{j,k} \geq \hat{r}_i^t, \forall i \in \mathcal{D}, t \in [0, T] \tag{3.3}$$

$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{B}} x_{i,t}^{j,k} R_j \geq \hat{r}_i^t, \forall i \in \mathcal{D}, t \in [0, T] \tag{3.4}$$

$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{B}} x_{i,t}^{j,k} (\hat{T}_{i,t}^{j,k} + T_j) \leq \hat{d}_i^t, \forall i \in \mathcal{D}, t \in [0, T] \tag{3.5}$$

$$\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{B}} x_{i,t}^{j,k} = 1, \forall i \in \mathcal{D}, t \in [0, T] \tag{3.6}$$

$$x_{i,t}^{j,k} \in \{0, 1\}, \forall i \in \mathcal{D}, \forall j \in \mathcal{N}, \forall k \in \mathcal{B}, t \in [0, T] \tag{3.7}$$

In problem (3), constraint (3.1) represents the limitation of wired link resource, where the total transmission rate offered by NS cannot exceed the link resource budget during any time slot t. Constraint (3.2) states the wireless bandwidth limitation, which means that the total wireless bandwidth allocated to devices by NS j via BS k cannot exceed the total bandwidth of NS j allocated from BS k during any time slot t. Constrains (3.3) - (3.5) state that the service quality of devices should be satisfied by its serving BS and NS during any time slot t even the selected NS/BS pair and network environment change. Specifically, constraints (3.3) and (3.4) guarantee the transmission rate, and constraint (3.5) guarantees the end-to-end delay. Moreover, constraint (3.6) represents the access limitation, which means that a device can access only one NS via one BS during time slot t. The binary constraint on the decision variable is shown in (3.7).

Theorem 1. *Problem (3) with constraints (3.1)-(3.7) is NP-hard.*

Proof: A special case with fixed $w_{i,t}^{j,k}$ and $r_{i,t}^{j,k}$ in problem (3), can be mapped into a Multiple Choice Multidimensional Knapsack problem (MMKP) [38] which is NP-hard [39]. When $w_{i,t}^{j,k}$ and $r_{i,t}^{j,k}$ change with time, problem (3) with constraints (3.1)-(3.7), is a dynamic MMKP (DMMKP). If DMMKP has solution in polynomial time, its corresponding MMKP should also have solution in polynomial time. Thus, DMMKP can reduce to MMKP. Therefore, problem (3) with constraints (3.1)-(3.7), is NP-hard. ■

B. Markov Decision Process Modeling for Device Association

As Problem (3) is NP-hard, there is no polynomial-time algorithm for solving it. Meanwhile, in view of the dynamic nature of access conditions, the change of relevant parameters (including consumed bandwidth, transmission rate, and service delay) in the device association scheme over time, we formulate the device association problem as a markov decision process (MDP) model. An MDP consists of four-tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R)$, where $\mathcal{S}$ represents the state space, $\mathcal{A}$ represents the action space, P is the transition probability between states, and R represents the reward function. As shown in Fig. 2, for a specific device, the device needs to make a decision to select an appropriate combination of BS and NS (action) to access at the beginning of each time slot. This may change the state of access conditions, causing the network state to transit to another state. Through this action, the device can obtain a certain reward. The state, action, transition probability, and reward are respectively defined as follows.

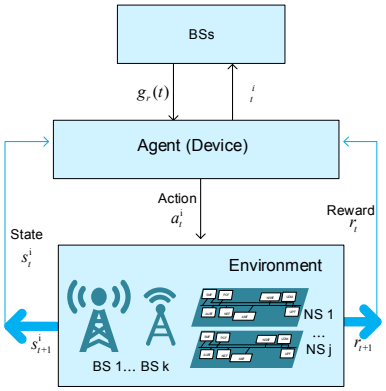


Fig. 2. The Markov transition diagram.

State: The current access conditions are used to describe the system state. We assume $\mathcal{S}$ is the set of all network states for all devices, and the number of NSs and BSs are $|\mathcal{N}|$ and $|\mathcal{B}|$ respectively. For a specific device $d_i \in \mathcal{D}$, the state can be represented by $s_t^{\mathrm{i}} = \{I_{\mathrm{i}}, b_{1,1}^t, ..., b_{j,k}^t, ..., b_{|\mathcal{B}|,|\mathcal{N}|}^t\}$, where $s_t^{\mathrm{i}} \in \mathcal{S}$, $I_{\mathrm{i}} = (j,k)$ states the current selected NS/BS pair of d_i, and $b_{j,k}^t$ represents the available wireless bandwidth allocated to NS j from BS k at time slot t with constraint $b_{j,k}^t \leq b_{j,k}$. Moreover, we filter out $b_{j,k}^t$ if NS j or BS k is not engaged for d_i.

Action: We remove the infeasible actions which do not satisfy either the network resource constraints (3.1)-(3.2), the service quality of devices (3.3)-(3.5), or the access constraints (3.6)-(3.7). Moreover, we assume $\mathcal{A}$ is the set of actions for all devices. For a specific device $d_i \in \mathcal{D}$, let $a_t^{\mathrm{i}} = (j, k, w_{i,t}^{j,k})$ be the action, which means d_i will consume $w_{i,t}^{j,k}$ MHz wireless bandwidth if it accesses to NS j via BS k at time slot t. Here $w_{i,t}^{j,k}$ is selected randomly from $[\hat{r}_i^t / log_2(1 + SINR_{i,t}^k), b_{j,k}^t]$, where $\hat{r}_i^t$ is the minimal transmission rate of d_i at time slot t.

Transition Probability: Let the transition probability of d_i be $P = \left\{ p_{s_t^{\mathrm{i}} s_{t+1}^{\mathrm{i}}}^{a_t^{\mathrm{i}}} | a_t^{\mathrm{i}} \in \mathcal{A}, s_t^{\mathrm{i}}, s_{t+1}^{\mathrm{i}} \in \mathcal{S} \right\}$, which represents the probability that network state of d_i transits from s_t^{i} to s_{t+1}^{i} through action a_t^{i}.

Reward: In order to maximize the communication efficiency while considering the incurred communication cost in FL, we define the reward as $r_t = e_t - u \cdot x_t \cdot \alpha_{i,k}^{\mathrm{c}}$, where u is the number of devices, x_t is the number of communication rounds in FL from the first time slot to the tth time slot, and $\alpha_{i,k}^{\mathrm{c}}$ is the communication cost of each round in FL between d_i and BS k. More details about communication round and FL are shown in next section.

In summary, the information used for training local DRL model includes the consumed radio resources (*i.e.*, $w_{i,t}^{j,k}$), the current selected NS/BS pair (*i.e.*, $I_i = j,k$), the communication efficiency (*i.e.*, e_t), the handoff cost (*i.e.*, α^{HO}), the bandwidth allocated to NS j from BS k (*i.e.*, $b_{j,k}$), and the available bandwidth allocated to NS j from BS k at time slot t.

In the MDP for device association, a smart device can obtain an optimal long-term reward by continuously interacting with the network environment. But the effectiveness fades as the size of the state-action space becomes large (*i.e.*, the state space of MDP for device association is a discrete space with $|\mathcal{B}| \cdot |\mathcal{N}| + 1$ dimensions, the action space is a discrete space with $|\mathcal{B}| \cdot |\mathcal{N}| \cdot b_{j,k}$ dimensions). To address the aforementioned difficulty, we employ DRL to solve the decision-making problem of a large size of state-action space. Meanwhile, in the paradigm of distributed machine learning, federated learning can be exploited to efficiently promote the collaboration between devices and save the network bandwidth consumption for transmitting training data while retaining the privacy of local data.

V. Hybrid Federated Deep Reinforcement Learning for Device Association

A. Framework of HDRL

By incorporating the DRL into the FL framework, we propose a collaborative hybrid federated deep reinforcement learning scheme, called HDRL. Fig. 3 shows the architecture of HDRL, which consists of DRL running on individual devices, and two levels of model aggregation based on DRL: horizontal weights aggregation (called hDRL) and vertical access feature aggregation (called vDRL). Specifically, in hDRL, we exploit hFL for the same type services to aggregate the parameters (*i.e.*, θ_t^{i}) to share the similar data samples, where smart devices and BSs can be enabled to train a global model (*i.e.*, $g_r(t)$) together without raw data transfer. As the RAN needs to support multiple service types, the selected NS/BS pairs derived from hDRL may be not optimal. Therefore, in vDRL, we exploit vFL to aggregate the access features to form a larger feature space for different types of services (*e.g.*, in the scenario of Fig. 3, there are two service types). In our proposed HDRL framework, devices such as cell phones are involved in local model training. Although a certain amount of energy is consumed, HDRL is effective for the following reasons: 1) The power consumption of the training process on a smartphone is smaller than that of some typical smartphone applications (*e.g.*, video play, large-scale game) [40]; 2) Many cell phone vendors, *e.g.*, Apple, Huawei, have introduced smartphones with dedicated powerful AI chips

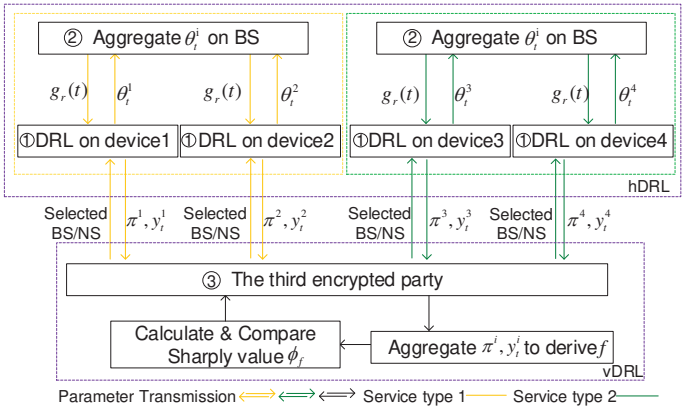


Fig. 3. The hybrid federated deep reinforcement learning (HDRL) framework for device association.

that can perform up to 5 trillion operations per second [41] and thus the battery power consumed by training neural network on the phone is acceptable [40]; 3) Our proposed collaborative HDRL enables independent devices to jointly train the global model together, where the number of training samples and the number of trainings on each device can be reduced and thus the energy consumption for local model training is limited. In the following, we elaborate the detailed mechanisms at smart devices and two aggregation levels.

DRL on smart devices: As FL can inherently support privacy protection on private data, the training data should be kept where it is generated. In other words, devices need to train their own data independently. Furthermore, on one hand, modern smart devices (*e.g.*, smartphones) have fast processors (including GPUs) and AI chips to accelerate training and reduce energy consumption [42]. On the other hand, the state space of MDP for device association is a discrete space with $|\mathcal{B}| \cdot |\mathcal{N}| + 1$ dimensions, and the action space is a discrete space with $|\mathcal{B}| \cdot |\mathcal{N}| \cdot b_{j,k}$ dimensions. Therefore, we employ the discrete-action DRL algorithm, double deep Q-Network (DDQN), to train the local model on individual smart devices. DDQN can address MDP with large state-action space by introducing the experience pool, improve the stability of the training results by introducing the target network, and decouple the selection from the evaluation to reduce the correlation between data [43].

Horizontal model aggregation (hDRL) level: Since different smart devices may generate local data with different patterns based on the usage of the devices, no device has a representative sample of the popular distribution in general. However, for the same type of service, the flow data from different devices is strongly correlated because the data flows not only have similar features (*e.g.*, the service type mark), but also compete for the radio and computing resources in similar slices. Therefore, for the different services of the same type, we propose horizontal aggregation to integrate the similar data samples to train a global access model by adopting an iterative approach that requires a number of model update iterations, where each model update iteration is called a *communication round* [7] [10]. Fig. 4 shows the process of a communication round, which consists of five steps: initialization of DDQN parameters, local model training, local model transmission, global model update, and global model transmission. In each communication round, we aim to update model through the cooperation between BSs and smart devices (*i.e.*, aggregating training samples for updating the global model and using DRL for updating the local model). As a result, an individual device can share the updates of parameters with other devices.

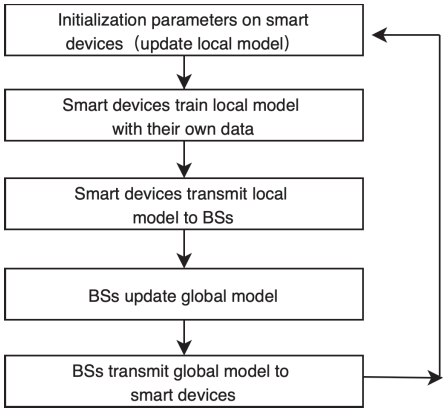


Fig. 4. The process of a communication round.

hDRL is performed in two steps: 1) DRL for training and updating local model; 2) Horizontal weights aggregation for aggregating training samples with some similar data features. Specifically, in the first step, based on the received global model, all devices use their own data to update the local model at the beginning of each communication round, and then continue to train local model through using DRL with the aim to approach optimal parameters that minimize the loss function. In the second step, individual devices send their local model to the corresponding BSs at the end of each communication round. Upon receiving all local models of the trained devices, BSs will update the global model and then send back the updated global model to individual devices.

Vertical model aggregation (vDRL) level: As the RAN needs to support multiple types of services, horizontal aggregation for aggregating the similar data samples may be not optimal. Therefore, in order to further promote the collaboration between devices, we aggregate the local access features to form a global access feature, where the data from different flows is strongly correlated because data flows compete for radio resources with each other. Furthermore, as shown in the Table. 1 and Table. 2 in [10], it brings much more communication cost to directly aggregate the data of services of different types compared with aggregating the data of the same type services, because the parameters are more frequently transmitted and updated in each communication round. Therefore, in order to reduce communication cost in vDRL, based on the aggregated global access feature, we introduce Shapley values [11] in (4) which represents the average marginal contribution of a specific feature across all possible feature combinations, to compare the importance of global access feature,

$$\phi_f = \frac{1}{M} \sum_{m=1}^{M} \left(f(x_{+i}^m) - f(x_{-i}^m) \right), \tag{4}$$

where M is the number of iterations. $f(x_{+i}^m)$ is the prediction

for instance x. x_{-i}^m is identical to x_{+i}^m, except that x_{+i}^m is different. Thus, we can derive the global optimal decision on NS and BS selection by selecting the maximal Shapley value. Here the access feature refers to the selected NS/BS pair and its corresponding estimated value (*i.e.*, the target value for local selected NS/BS pair and the Shapley value for global decision on NS and BS selection).

In vDRL, both the local and global selected NS/BS pairs can be represented by a 0-1 matrix (*i.e.*, local 0-1 matrix, global 0-1 matrix). Specifically, the global 0-1 matrix is composed of the row vectors of the local 0-1 matrices, where each row vector of the global 0-1 matrix represents a selected NS/BS pair of a specific device. Moreover, a local 0-1 matrix consists of possible local selected NS/BS pairs, where each row vector of a local 0-1 matrix represents a specific selected NS/BS pair of this device. Therefore, we can update the local and global decisions on NS and BS selection by changing the row vectors of the global 0-1 matrix. Furthermore, three steps are executed in vDRL: 1) Aggregate access features. All devices will send their own local access features (0-1 matrix and the corresponding target values) to the trusted third encrypted party every several communication rounds (*i.e.*, v). Thus, different global decisions on NS/BS selection (*i.e.*, global 0-1 matrix) can be obtained by selecting different row vectors of the local 0-1 matrices; 2) Calculate and compare Shapley value. Based on the formed global 0-1 matrices, we can calculate the Shapley values and obtain a global optimal 0-1 matrix (global optimal decision on NS and BS selection) with the maximal Shapley value by comparing these Shapley values; 3) Store and update the global decision on NS and BS selection. The third encrypted party will store the global optimal decision on NS and BS selection and Shapley value until it is replaced by a better one. Here the third encrypted party is a logical entity used to aggregate the different features without exposing their respective data. Currently, there are no standards or researchers explicitly specify which entity could play the role of the third encrypted party in a mobile network. In our views, a MEC/cloud encrypted server or a secure computing node in CN/RAN could serve as the third encrypted party because the aggregation on this entity needs certain computing resources. Indeed, the instantiation of the third encrypted party does not affect the effectiveness of our proposed algorithm.

B. Algorithm of Horizontal Model Aggregation

DDQN for training local model: At the beginning of each communication round, smart devices will receive the global model from BSs to update their local model (*i.e.*, weight θ) if communication round $r \neq 1$. Otherwise the devices will update their local model directly with initial weights (which are set as zero). We assume that each communication round consists of τ time slots. During each time slot, each device performs local training once. Therefore, after completing the local model update, the devices continue to train their local model independently with DDQN during τ time slots. DDQN evaluates the greedy policy according to the Q-network with weight θ and estimates state-action value $Q(\cdot)$ according to the target network $\hat{Q}$ with weight $\hat{\theta}$ [43]. The update in DDQN is the same as that in DQN, but the target is replaced by

$$y_t^{\mathrm{i}} = r_{t+1} + \gamma Q(\mathrm{s}_{t+1}^{\mathrm{i}}, \mathrm{argmax}_{\mathrm{a}_t^{\mathrm{i}}} Q(\mathrm{s}_{t+1}^{\mathrm{i}}, \mathrm{a}_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}); \hat{\theta}_t^{\mathrm{i}}), \quad (5)$$

where $\mathrm{argmax}_{\mathrm{a}_t^{\mathrm{i}}} Q(\mathrm{s}_{t+1}^{\mathrm{i}}, \mathrm{a}_t^{\mathrm{i}}; \theta_t^{\mathrm{i}})$ is an ϵ-greedy policy used to select access or handoff actions, and θ_t^{i} is the weight vector of Q-network for device d_i.

For a specific device d_i, if d_i satisfies the access condition and takes access or handoff action $\mathrm{a}_t^{\mathrm{i}}$ at the beginning of time slot t, we will obtain the corresponding state-action value, which is given by

$$Q(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}) = \mathbb{E}[\sum_{k=t}^{T} \gamma^k r_t | s_t^{\mathrm{i}}, a_t^{\mathrm{i}}], \quad (6)$$

where $\gamma \in [0, 1]$ is the discount factor representing the discounted impact of the future reward. The objective of DDQN is to minimize the gap between the estimated $Q(\cdot)$ and the target value. Therefore, DDQN running on d_i can be trained by minimizing the loss function, which is given by

$$L(\theta_t^{\mathrm{i}}) = \mathbb{E}[(y_t^{\mathrm{i}} - Q(\mathrm{s}_t^{\mathrm{i}}, \mathrm{a}_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}))^2]. \quad (7)$$

Moreover, when DDQN approximates the value function using the neural network, it indeed updates the parameter value θ_t^{i} by using the gradient descent method. Therefore, the update algorithm in DDQN is given by

$$\theta_{t+1}^{\mathrm{i}} = \theta_t^{\mathrm{i}} + \alpha \left[y_t^{\mathrm{i}} - Q\left(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}\right) \right] \nabla Q\left(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}; \theta_t^{\mathrm{i}}\right). \quad (8)$$

After training local data for τ time slots, device d_i will send the local model (*i.e.*, θ_t^{i}) to the BSs to update the global model.

Update models: Once receiving all local models from individual devices, BSs will update the global model as follows,

$$g_r(t) = \frac{\sum_{i=1}^{u_x} K_i \theta_t^{\mathrm{i}}}{K}, \forall 1 \leq t \leq T, \quad (9)$$

where K_i is the amount of training data of d_i, $K = \sum_{i=1}^{u_x} K_i$ is the total amount of training data of the devices with service of type x, u_x is the number of devices which has the same service type x, and r represents the rth communication round of hDRL. After updating the global model in the rth communication round, BSs will transmit the global mode $g_r(t)$ to all devices with the same type services to update the local DDQN models based on (10).

$$\theta_{t+1}^{\mathrm{i}} = g_r(t) - \frac{\lambda}{K_i} \sum_{i=1}^{u} \nabla L(\theta_t^{\mathrm{i}}), \forall i \in \mathcal{D}, 1 \leq t \leq T, \quad (10)$$

where λ is the learning rate, and $L(\theta_t^{\mathrm{i}})$ is the loss function of DDQN in (7). After updating the local model, the devices will continue to train their local model. The horizontal model aggregation algorithm is presented as **Algorithm 1**, where the complexity of horizontally FL framework is $\mathcal{O}\left(R(u + |\mathcal{B}|)\right)$ because each communication round includes the computation of BS aggregation and local model updating, where R, u, and $|\mathcal{B}|$ are the number of communication rounds, smart devices, and BSs.

Algorithm 1 Algorithm of Horizon Model Aggregation

Input: s^{i}, a^{i}, α, γ, C, R, K_i, u_x, x, λ, τ
output: Selected NS/BS pair π^{i}, target value y_t^{i}.

Initialize experience relay pool $D_x^{\mathrm{i}}, \forall i \in \mathcal{D}$;
Initialize the global weights g_0;
for communication round $r = 1, 2, ..., R$ **do**
  if $r == 1$ **then**
    Initialize θ_0^{i};
  else
    ▷ Update local model
    for $i = 1, 2, ..., u_x$ **do**
      $\theta_0^{\mathrm{i}} = g_{r-1}(t) - \frac{\lambda}{K_i} \sum_{i=1}^{u_x} \nabla L(\theta_t^{\mathrm{i}})$.
    end for
  end if
  ▷ Local model training
  Let $\hat{\theta}_0^{\mathrm{i}} = \theta_0^{\mathrm{i}}$, initialize target action-value function $\hat{Q}(\cdot)$ according to the parameter $\hat{\theta}_0^{\mathrm{i}}$;
  for $t = 1$ to τ **do**
    Receive the initial observed state $s_1^1, s_1^2, ..., s_1^{\mathrm{u}_x}$;
    if $t \leq |D_x^{\mathrm{i}}|$ **then**
      Randomly select $a_t^1, a_t^2, ...$;
    else
      Select $a_t^{\mathrm{i}} = \mathrm{argmax}_a Q(\cdot)$ using ϵ-greedy policy;
      Execute action a_t^{i}, obtain r_t^{i} and s_{t+1}^{i};
      Store $(s_t^{\mathrm{i}}, a_t^{\mathrm{i}}, r_t^{\mathrm{i}}, s_{t+1}^{\mathrm{i}})$ into $D_x^{\mathrm{i}}, \forall i \in \mathcal{D}$;
      Randomly select a sample $\left(s_j^{\mathrm{i}}, a_j^{\mathrm{i}}, r_j^{\mathrm{i}}, s_{j+1}^{\mathrm{i}}\right)$ from the experience relay pool $D^{\mathrm{i}}, \forall i \in \mathcal{D}$;
      Calculate y_t^{i} according to equation (4);
      Perform a gradient descent step on

$$\sum_{i=1}^{u} \left(y_j^{\mathrm{i}} - Q\left(s_j^{\mathrm{i}}, a_j^{\mathrm{i}}; \theta_t^{\mathrm{i}}\right)\right)^2$$

      Update the parameter $\theta_t^{\mathrm{i}}, \forall i \in \mathcal{D}$;
      Every C slots reset $\hat{Q} = Q$;
    end if
  end for
  ▷ Update global model
  for $i = 1, 2, ..., u_x$ **do**
    $g_r(t) = \frac{\sum_{i=1}^{u} K_i \theta_\tau^{\mathrm{i}}}{K}$.
  end for
end for
Obtain selected NS/BS pair π^{i}, target value y_t^{i}.

C. Algorithm of Vertical Model Aggregation

The aforementioned horizontal model aggregation is used for the same type services with similar data samples. As multiple types of services are considered in this paper, vertical model aggregation could be exploited for further improving the network performance, by aggregating local access features incurred from different types of services. Due to the data on each device is private and not visible to other devices, we use a 0-1 matrix to represent a local global decision on NS and BS selection, where we can update global access feature by transforming these 0-1 matrices. In this paper, according to [11], the estimated global target value of a global decision on NS and BS selection is given by

$$\varphi_f = \sum_{i=1}^{u} y_t^{\mathrm{i}} - \mathbb{E}\left[\sum_{i=1}^{u} y_t^{\mathrm{i}}\right], \tag{11}$$

where $f \subseteq \mathcal{X}$ is a specific global decision on NS and BS selection (0-1 matrix), each row vector of f represents a local selected NS/BS pair. Moreover, y_t^{i} is the target value in (5). For example, we assume there are two devices (*i.e.*, d_1 and d_2) sending two service requests in the overlapping area of multiple BSs (*i.e.*, BS 1, BS 2). Thus $\mathcal{X}$ can be given by

$$\mathcal{X} = \{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}, \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, \begin{bmatrix}1 & 0\\ 1 & 0\end{bmatrix}, \begin{bmatrix}0 & 1\\ 0 & 1\end{bmatrix}\},$$

where $\mathcal{X}$ is the set of possible global selections. Each element of $\mathcal{X}$ represents a global decision on NS and BS selection, composed of row vectors of matrix $\mathbf{A}$ and $\mathbf{H}$, where $\mathbf{A}$ and $\mathbf{H}$ are given by

$$\mathbf{A} = \mathbf{H} = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}.$$

In this case, $\mathbf{A}$ and $\mathbf{H}$ represent the possible selected NS/BS pairs of device 1 and device 2 respectively. For example, in $\mathbf{A}$, the first row [1 0] represents that device 1 accesses to BS 1 and the second row [0 1] represents that device 1 accesses to BS 2. Moreover, the sum of each row of $\mathbf{A}$ or $\mathbf{H}$ is 1, which means that a device can only access to one BS.

In [11], the authors proposed an Monte-Carlo sampling, where the Shapley value is given by

$$\phi_f = \frac{1}{M} \sum_{m=1}^{M} \left(\varphi_{+f} - \varphi_{-f}\right), \tag{12}$$

where M is the number of access feature updates in vDRL. Moreover, ϕ_f is the Shapley value for a specific global decision on NS and BS selection f, representing the average marginal contribution of f across all possible access feature combinations $\mathcal{X}$. For example, in $\mathcal{X}$ above-mentioned, if $f = \mathcal{X}\{1\}$, we can get the $+f = \mathcal{X}\{1\}$ and $-f$ is randomly chosen in $\{\mathcal{X}\{2\}, \mathcal{X}\{3\}, \mathcal{X}\{4\}\}$. Therefore, we can obtain the Shapley value ϕ_f of the corresponding global decision on NS and BS selection f through (12) and derive the global optimal 0-1 matrix by comparing the Shapley values. Thus, when the devices send their service requests, the third encrypted party will send the i-th row vector of f to devices, where the i-th row vector of f represents the local selected NS/BS pair of d_i.

D. HDRL Algorithm for Device Association

Next we elaborate the model training process of HDRL scheme. Fig. 5 shows an illustrative example of HDRL process, where two types of services are considered. In this process, we assume that each communication round consists of τ time slots, where only the first and last time slots in a communication round are involved in the global parameter aggregation. Between BSs and devices, the parameters (including global model $g_r(t)$ and local model θ_t^{i}) are transmitted and updated for a global model for the same type services. In the

first time slot of each communication round, the global model will be sent to individual devices if communication round $r \neq 1$. Otherwise the devices will update their local model directly with initial weights zero. During each communication round, based on the received global model or initial weights, the devices will update local weights and train their own data with DDQN. At the last time slot of each communication round, according to the service type, the devices will send their local model to BSs to update corresponding global model. It is worth noting that each device and each BS store all models of all the service types.

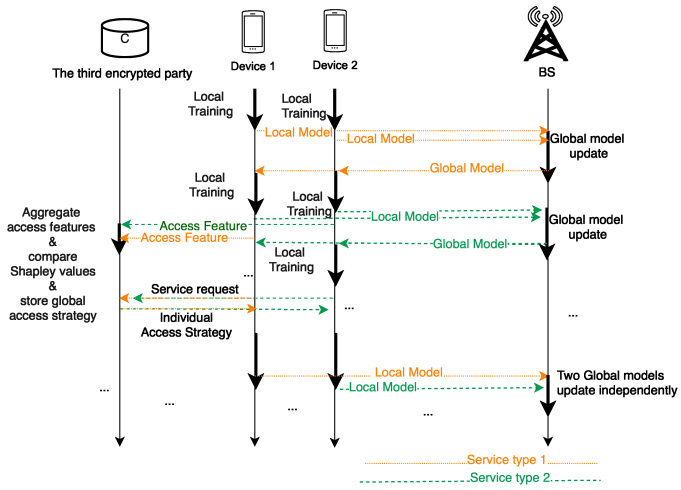


Fig. 5. The process of HDRL.

The global access feature aggregation is performed in the last time slot every several communication rounds (*i.e.*, v). Between the third encrypted party and devices, access features are transmitted and aggregated for making a global decision on NS and BS selection. At the last time slot of every v communication rounds, we aggregate the access feature of individual devices to form a global access feature on the third encrypted party. Based on the aggregated global decision, we calculate the Shapley values and derive a global optimal decision on NS and BS selection with the maximal Shapley value by comparing these Shapley values. Then the third encrypted party will store the global decision on NS and BS selection until it is replaced by a better one. Note that if devices send their service requests simultaneously, the third encrypted party will send the corresponding local selected NS/BS pairs to devices. Otherwise the smart devices can make decisions on NS and BS selection according to their own local model. Based on the train Algorithm 1, the HDRL algorithm is presented as Algorithm 2.

E. Convergence Analysis

HDRL can be regarded as fully distributed DRL if neither horizontal aggregation nor vertical aggregation is performed (*i.e.*, $r = R = v = 0$), and also can be regarded as centralized DRL if we perform global aggregation after every local update (*i.e.*, $\tau = v = 1$) and then the data samples and features are available for the centralized controller, and the communication cost is ignored [44]. Furthermore, we use

Algorithm 2 HDRL

Input: M, individual selected NS/BS pair π_i, and target value y_t^{i} from **Algorithm 1**, iterations.
output: Shapley value ϕ_f, the optimal global decision on NS and BS selection f.

Initialize the maximum Shapley value $\phi_{max} = 0$;
Initialize selected NS/BS pairs for all devices $f_0 = \varnothing$;
for $m = 1, 2, ..., M$ **do**
  Get π_i, θ_{i}, and y_t^{i} of the devices with different service categories in the same overlapping converges of multiple BSs;
  Get $\mathcal{X}$ through π_i;
  Remove unfeasible solution in $\mathcal{X}$;
  for $i = 1, 2, ..., |\mathcal{X}|$ **do**
    $f = \mathcal{X}\{i\}$;
    Initial the sets of $-f$;
    Calculate φ_f;
    for iterations$= 1, 2, ...$ **do**
      Choose $-f$ in $\{\mathcal{X} - \mathcal{X}\{i\}\}$;
      if $-f \subseteq F$ **then**
        Continue.
      else
        Calculate φ_{-f}.
      end if
    end for
    Calculate ϕ_f
    if $\phi_{max} \leq \phi_f$ **then**
      $\phi_{max} = \phi_f$;
      $f_0 = f$.
    end if
  end for
end for
Obtain Shapley value ϕ_f, the optimal global decision on NS and BS selection $f = f_0$.

an auxiliary parameter vector $\mathbf{v}_t^r$, which follows a centralized gradient descent according to

$$\mathbf{v}_{t+1}^r = \mathbf{v}_t^r - \eta \nabla L(\mathbf{v}_t^r), 1 \leq t \leq \tau, \ \forall r \in R. \tag{13}$$

According to [10], [44], the global parameters $g_r(t)$ should be very close to $\mathbf{v}_t^r$ when $\tau = v = 1$. Formally, we have an upper bound on the difference between $L(g_r(t))$ and $L(\mathbf{v}_t^r)$ within $[t - (r-1)\tau, t]$, which is given by

$$|L(g_r(t)) - L(\mathbf{v}_t^r)| \leq h(\tau, r). \tag{14}$$

We have $h(\tau, r) = 0$ if fully distributed DRL or centralized DRL is performed. However, the fully distributed DRL, $r = R = v = 0$, is always the worst solution compared with centralized DRL and HDRL, since fully distributed DRL always only considers independent training and independent decision on NS and BS selection. Moreover, it is always optimal when setting $\tau = 1$ and $v = 1$ if we have unlimited resource budget and ignore the privacy issue, since centralized DRL jointly trains a global model for all services. Theoretically, the performance of model training in HDRL should be between that of the fully distributed DDQN and centralized DDQN. From the

investigations in [44], $h(\tau, r)$ is affected by data distribution, r, and τ. Therefore, when data distribution, τ and r are given, we can obtain the upper bound of the divergence (*i.e.*, $h(\tau, r)$) between HDRL derived loss function and the global loss function. Moreover, since we use a non-linear sigmoid function in the neural network, the loss function in this paper is non-convex. Therefore, we can obtain $h(\tau, r)$ through training and further obtain the convergence bound of HDRL derived loss function $[L(\mathbf{v}_t^r) - h(\tau, r), L(\mathbf{v}_t^r) + h(\tau, r)]$. Intuitively, the frequency of performing global weights aggregation (*i.e.*, τ) should be carefully specified, as the communication cost with a large number of communication rounds cannot be ignored. Numerical results in the subsequent section will illustrate this.

VI. Numerical Results

In this section, we evaluate the performance of our proposed HDRL scheme through simulation experiments. We employ four reference device association (DA) schemes as comparison reference:

(1) Greedy Algorithm for DA (GDA): In this scheme, each device chooses NS/BS to access which can provide the maximal communication efficiency based on instantaneous network conditions, instead of considering long-term optimal communication efficiency.
(2) Centralized DDQN for DA (CDA): In this scheme, all devices transmit data to a controller for centralized training in DDQN. Then the controller makes global decision on NS and BS selection for all devices, where no cost for transferring training data is taken into account.
(3) Distributed DDQN without data aggregation for DA (DDA): In this scheme, individual devices train their own data through DDQN and make decision on NS and BS selection independently, where no data aggregation of FL is used. Moreover, the reward function in CDA and DDA remains the same as that in HDRL except that the cost of communication round is zero.
(4) RSRP-based BS selection for device association (RDA): In this scheme, devices first select the BS with highest RSRP, and then select the slice that can provide the maximum communication efficiency on this BS.

A. Simulation Settings

We consider a network scenario where four BSs are randomly distributed in a square area of 1060×1060 m^2 [23]. The parameters of network scenario are listed in Table III. Specifically, we assume that five end-to-end slices are deployed in the network. The maximal transmit power and the noise power of BSs are set to 47dBm and -174dBm/Hz respectively [23], [38]. The path loss for BSs is modeled as $L(d) = 34 + 40\log(d)$ [23], [38]. Furthermore, the wireless bandwidth of each BS is set to 20 MHz. For a specific BS k, the total wireless bandwidth is randomly allocated to all NSs deployed at BS k. In other words, $b_{j,k}$ is randomly chosen from [0, 20] MHz with the constraint $\sum_{j=1}^{5} b_{j,k} \leq 20MHz$ [23]. Meanwhile, devices are randomly distributed within the simulation area with different transmission rate and delay requirements. In this paper, we assume three types of services (*i.e.*, eMBB, mMTC, and URLLC) are supported. The service type is characterized by transmission rate and delay, where the minimal transmission rate $\hat{r}_i^t$ is randomly generated from [2, 10]Mbps [23], [45] and the delay in CN is randomly generated from [1, 10]ms [45].

TABLE III
Network Parameters

Parameter	Value
The number of BSs	4
The number of NSs	5
Simulation area	1060×1060 m^2
Noise power	-174dBm/Hz
Path loss function	$L(d) = 34 + 40\log(d)$
BS wireless bandwidth	20MHz
The minimal transmission rate	U[2, 10]Mbps
The delay in CN	U[1, 10]ms
The maximal transmit power of BSs	47dBm
The wireless bandwidth that BS k allocates to NS j, $b_{j,k}$	U[0,20]MHZ

TABLE IV
Parameters of Aggregation

Parameter	Value
The number of input neurons	12
The number of hidden neurons	25
Target network update interval step	5
The discount factor	0.99
Learning rate for training	0.001
Learning rate for updating local model	0.001
The batch size	64
Replay memory size	1000
Aggregation frequency of access features	2

Table IV lists the parameters used in two levels of aggregation. For each device, we consider a three-layer fully connected neural network, including input layer, hidden layer, and output layer. Specifically, the input layer consists of 12 neurons, representing the input of access condition and access/handoff action. The hidden layer consists of 25 neurons, where the activation function is set to sigmoid function. The output layer has one neuron, where the activation function is linear. To avoid correlation between action-values and target values, we copy the weights of Q-network θ to the weights of target network $\hat{\theta}$ every 5 training steps [46]. Moreover, the memory size for each service type is set to 1000, the batch size is set to 64, the discount factor is set to 0.99 [46], and the access feature is updated every 2 communication rounds. In addition, both the learning rate for training and the learning rate for updating local model are set to 0.001.

B. Numerical Results and Discussions

First, we examine the relationship between the total long-term reward and the frequency of performing global aggregation. In this experiment, we assume that the communication cost for one communication round, $\alpha_{i,k}^{c}$, is a constant which

is set to 0.05. Fig. 6 illustrates the total long-term reward as a function of the number of communication rounds. From Fig. 6, we can observe that, the total long-term reward increases in the beginning and then decreases with the number of communication rounds. There is an optimal number of rounds, say 15, which leads to the maximal total reward when the number of trainings in a communication round (*i.e.*, τ) is set to 2000. Furthermore, we observe that the number of trainings in a communication round (*i.e.*, τ) affects the optimal number of communication rounds. The reasons are as follows. On one hand, the local model changes with the number of trainings in a communication round (*i.e.*, τ). On the other hand, the communication cost increases with the number of communication rounds.

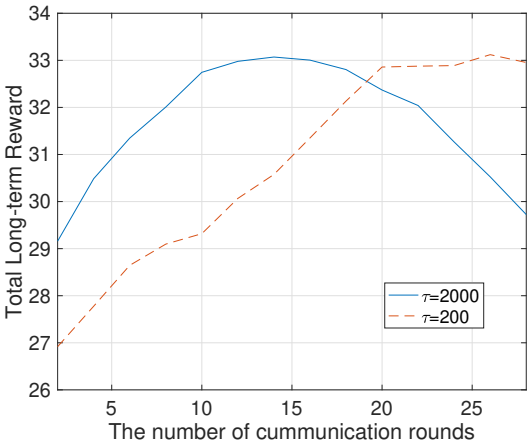


Fig. 6. The relationship between the number of communication rounds and the total long-term reward.

Then, we verify the convergence property of our proposed HDRL by depicting its learning curve (the curve of weights *vs.* the total number of trainings). We set $\tau = 2000$ and randomly select three corresponding local models (*i.e.*, θ^1, θ^2, θ^3) on three different devices. As shown in Fig. 7, HDRL converges with the total number of trainings increasing. Furthermore, Fig. 8 shows the partial convergence curve of Fig. 7 within $[5\tau, 20\tau]$. From Fig. 8, we observe that the three corresponding weights from different smart devices coincide when they tend to be stable, which further illustrates the effectiveness of training a global model with multiple independent smart devices. In the following experiments, we set $\tau = 2000$ and $R = 15$ to evaluate the performance of HDRL.

Next, we explore how the total long-term reward changes with the number of devices in HDRL. Fig. 9 shows the total long-term reward as a function of the number of devices. From Fig. 9, we see that the total long-term reward increases with the number of devices, and further, the increasing speed of the total long-term reward is different. The reasons are as follows. As the number of devices increases, HDRL can exploit more similar data samples for training. However, when the number of devices (*i.e.*, u) is more than 15, the number of duplicate data samples which should be filtered out increases quickly, so that the increasing speed becomes lower than that in the beginning. Moreover, when the number of devices is greater than 35, the devices have sufficient data samples to

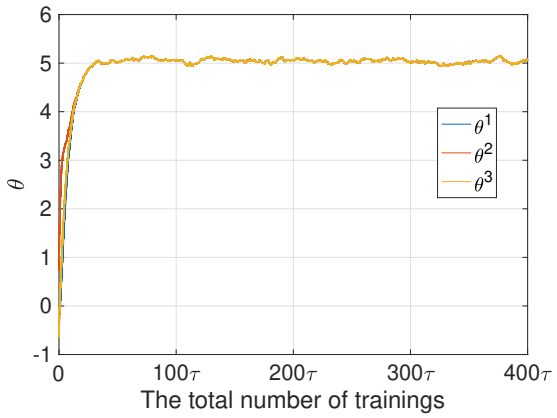


Fig. 7. Convergence of HDRL.

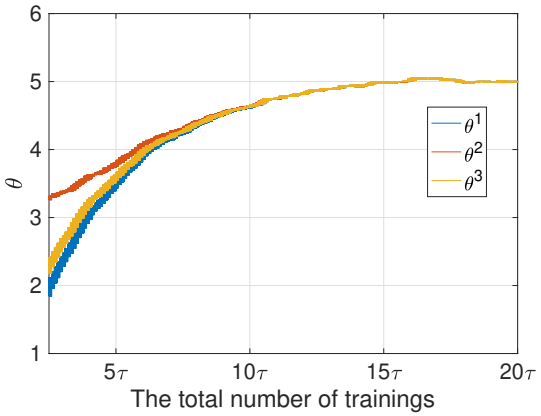


Fig. 8. Partial convergence curve of Fig. 7 within $[5\tau, 20\tau]$.

approximate the value function so that the total long-term reward increases rapidly. Furthermore, when the number of devices continues to increase (*i.e.*, more than 40), due to the constrained network resources, the increasing speed of the total long-term rewards decreases.

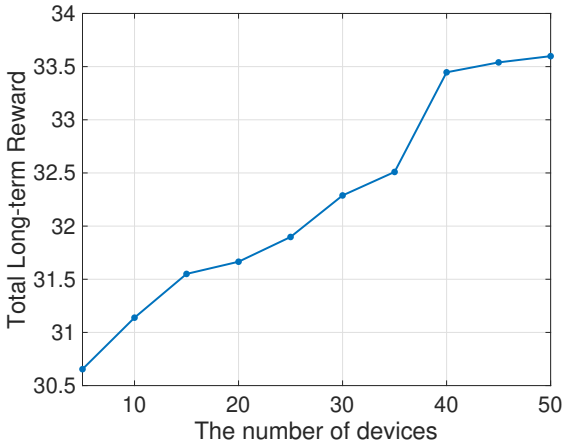


Fig. 9. The relationship between the number of devices and the total long-term reward.

Then, we compare the total long-term reward of five schemes (*i.e.*, HDRL, CDA, DDA, GDA, and RDA) when the number of devices is 35 (*i.e.*, $u = 35$). Fig. 10 shows the

total long-term reward of the five schemes. From Fig. 10, we can see that HDRL and CDA achieve higher long-term reward than other three schemes. This is because HDRL and CDA aim to find the global optimal decision on NS and BS selection, while DDA, GDA, and RDA focus on the local selected NS/BS pair. Moreover, compared with CDA, HDRL aggregates the same type services on BSs, reducing the correlation between the training data from different devices. Therefore, the total reward of HDRL is higher than that of CDA.

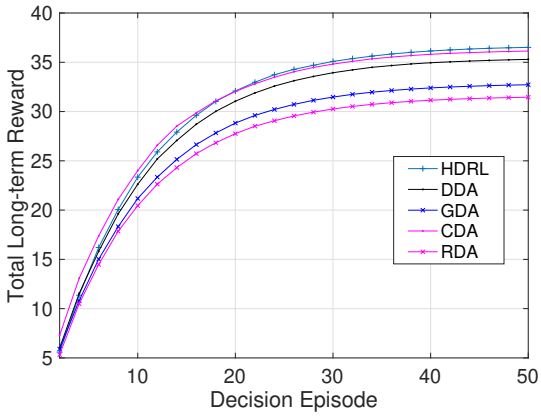


Fig. 10. The performance of the total long-term reward.

Next, we examine the performance of the five schemes in terms of network throughput. Fig. 11 shows the network throughput as a function of the number of devices. We can see that HDRL always outperforms CDA, DDA, GDA, and RDA on network throughput. This is because that HDRL integrates the similar data samples into a global model before aggregating the access features. Moreover, the access feature aggregation in HDRL takes the global optimal decision on NS and BS selection into account. In comparison, the duplicate data samples in CDA for centralized training increase the correlation of data, resulting in overfitting easily. Moreover, DDA, GDA, and RDA focus on individual devices without global perspective. In addition, GDA and RDA makes decisions on NS and BS selection based on instantaneous conditions, instead of considering long-term optimization objectives.

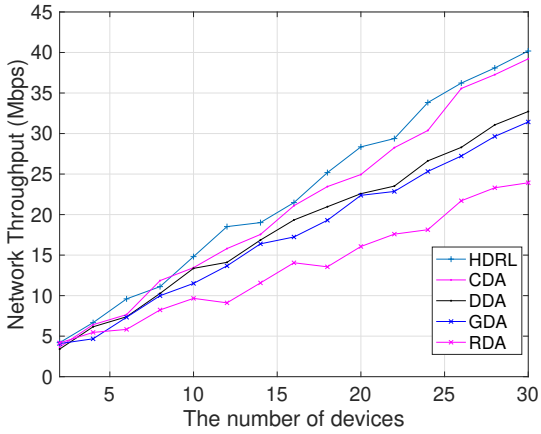


Fig. 11. Comparison of network throughput as a function of the number of devices in five schemes.

Next, we compare handoff cost of the five schemes. Fig. 12 shows the comparison of handoff cost of the five schemes. We can see that HDRL incurs the highest handoff cost. Moreover, the handoff cost of HDRL and DDA is always higher than that of CDA, this is because HDRL and DDA are based on distributed learning, where smart devices train their own data independently. Although HDRL employs two levels of aggregation, training on smart devices independently is not affected. Furthermore, we compare the communication efficiency by combining network throughput and handoff cost to further evaluate the performance of the five schemes.

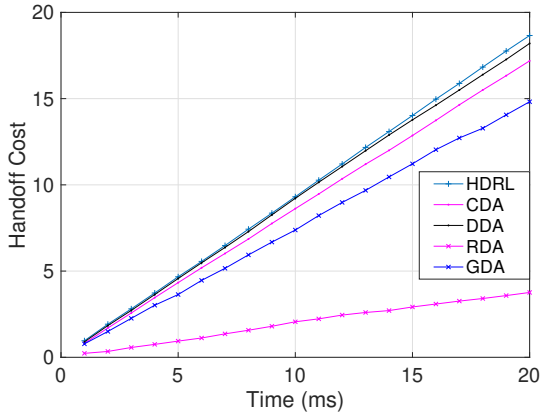


Fig. 12. Comparison of handoff cost of five schemes.

Finally, we compare the performance of communication efficiency of the five schemes in Fig. 13. We see that HDRL always outperforms DDA, CDA, GDA, and RDA in terms of communication efficiency. This is because HDRL not only considers the optimal global decision on NS and BS selection, but also integrates the similar data samples. In particular, numerical results show that HDRL achieves higher communication efficiency by about 14.19%, 20.80%, 26.60%, and 36.36% on average compared with CDA, DDA, GDA, and RDA respectively.

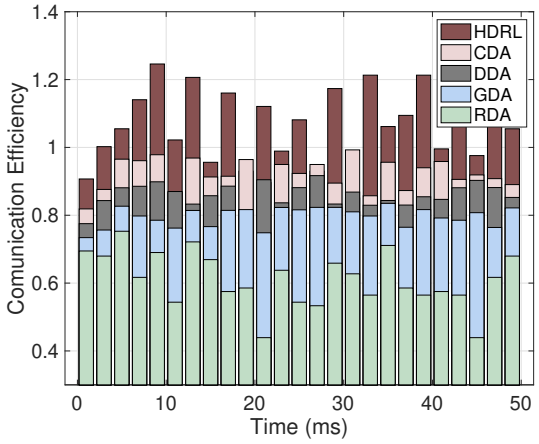


Fig. 13. Comparison of communication efficiency of five schemes.

VII. Conclusion

In this paper, with the aim to improve network throughput while reducing handoff cost, we have modeled the device

association problem for RAN slicing as an MDP model and solved it by developing a novel HDRL scheme that exploits hybrid FL based on DRL. In HDRL, we employ two levels of model aggregation based on DRL to promote the collaboration between smart devices while enforcing the privacy of local data. Numerical results show that our proposed HDRL scheme achieves a significant performance improvement in terms of network throughput and communication efficiency when compared with the state-of-the-art algorithms. In the future, hybrid FL based on DRL is still a challenging issue in Cloud-RAN in terms of privacy, independence, and service diversity, we will continue to explore HDRL schemes in Cloud-RAN in 5G-and-beyond wireless network.

REFERENCES

[1] Ericsson AB, "5G Systems – enabling the transformation of industry and society," *White Paper*, no. January, p. 14, 2017. [Online]. Available: https://www.ericsson.com/res/docs/whitepapers/wp-5g-systems.pdf

[2] S. Zhang, "An overview of network slicing for 5g," *IEEE Wireless Communications*, vol. 26, no. 3, pp. 111–117, 2019.

[3] Y. Sun, G. Feng, L. Zhang, M. Yan, S. Qin, and M. A. Imran, "User access control and bandwidth allocation for slice-based 5g-and-beyond radio access networks," in *proceeding of IEEE International Conference on Communications (ICC)*, 2019, pp. 1–6.

[4] B. Ojaghi, F. Adelantado, E. Kartsakli, A. Antonopoulos, and C. Verikoukis, "Sliced-ran: Joint slicing and functional split in future 5g radio access networks," in *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[5] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: Ran slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proceedings of the 23rd annual international conference on mobile computing and networking*, 2017, pp. 127–140.

[6] L. Zhang, J. Tan, Y. Liang, G. Feng, and D. Niyato, "Deep reinforcement learning-based modulation and coding scheme selection in cognitive heterogeneous networks," *IEEE Transactions on Wireless Communications*, vol. 18, no. 6, pp. 3281–3294, 2019.

[7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[8] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2020.

[9] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *proceeding of IEEE INFOCOM - Conference on Computer Communications*, 2019, pp. 1387–1395.

[10] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, p. 12, 2019.

[11] G. Wang, C. X. Dang, and Z. Zhou, "Measure contribution of participants in federated learning," *arXiv preprint arXiv:1909.08525*, 2019.

[12] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *arXiv preprint arXiv:1909.07972*, 2019.

[13] M. Yan, B. Chen, G. Feng, and S. Qin, "Federated cooperation and augmentation for power allocation in decentralized wireless networks," *IEEE Access*, vol. 8, pp. 48 088–48 100, 2020.

[14] G. Wang, G. Feng, W. Tan, S. Qin, R. Wen, and S. Sun, "Resource allocation for network slices in 5g with network resource pricing," in *proceeding of IEEE Global Communications Conference*, 2017, pp. 1–6.

[15] A. Kammoun, N. Tabbane, G. Diaz, N. Achir, and A. Dandoush, "Dynamic handler framework for network slices management," in *proceeding of International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2019, pp. 1–6.

[16] M. R. Sama, S. Beker, W. Kiess, and S. Thakolsri, "Service-based slice selection function for 5g," in *proceeding of IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.

[17] X. An, C. Zhou, R. Trivisonno, R. Guerzoni, A. Kaloxylos, D. Soldani, and A. Hecker, "On end to end network slicing for 5g communication systems," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3058, 2017.

[18] G. Wang, G. Feng, T. Q. S. Quek, S. Qin, R. Wen, and W. Tan, "Reconfiguration in network slicing—optimizing the profit and performance," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 591–605, June, 2019.

[19] A. Nakao, P. Du, Y. Kiriha, F. Granelli, A. A. Gebremariam, T. Taleb, and M. Bagaa, "End-to-end network slicing for 5g mobile networks," *Journal of Information Processing*, vol. 25, pp. 153–163, 2017.

[20] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5g telecommunication networks: A survey of principles and models," *IEEE Network*, vol. 33, no. 6, pp. 172–179, 2019.

[21] Y. Sun, S. Qin, G. Feng, L. Zhang, and M. A. Imran, "Service provisioning framework for ran slicing: user admissibility, slice association and bandwidth allocation," *IEEE Transactions on Mobile Computing*, 2020.

[22] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2429–2453, 2018.

[23] G. Zhao, S. Qin, G. Feng, and Y. Sun, "Network slice selection in softwarization-based mobile networks," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 1, 2020.

[24] M. K. Motalleb, V. Shah-Mansouri, and S. N. Naghadeh, "Joint power allocation and network slicing in an open ran system," *arXiv preprint arXiv:1911.01904*, 2019.

[25] A. Perveen, M. Patwary, and A. Aneiba, "Dynamically reconfigurable slice allocation and admission control within 5g wireless networks," in *proceeding of 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*. IEEE, 2019, pp. 1–7.

[26] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 10, pp. 2377–2392, 2019.

[27] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y.-C. Liang, "Intelligent resource scheduling for 5g radio access network slicing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7691–7703, 2019.

[28] Y. Sun, L. Zhang, G. Feng, B. Yang, B. Cao, and M. A. Imran, "Blockchain-enabled wireless internet of things: Performance analysis and optimal communication node deployment," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5791–5802, 2019.

[29] 3GPP, "System architecture for the 5G System (5GS); Stage 2," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 23.501, 03 2020, version 16.4.0.

[30] Y.-i. Choi and N. Park, "Slice architecture for 5g core network," in *2017 Ninth international conference on ubiquitous and future networks (ICUFN)*. IEEE, 2017, pp. 571–575.

[31] O. Sunay, S. Ansari, S. Condon, J. Halterman, W. Kim, R. Milkey, G. Parulkar, L. Peterson, A. Rastegarnia, and T. Vachuska, "ONF's Software-Defined RAN Platform Consistent with the O-RAN Architecture," no. February, 2020.

[32] S. Wang, R. Urgaonkar, M. Zafer, T. He, K. Chan, and K. K. Leung, "Dynamic service migration in mobile edge-clouds," in *proceeding of 2015 IFIP Networking Conference (IFIP Networking)*, 2015, pp. 1–9.

[33] Y. Mansouri, A. N. Toosi, and R. Buyya, "Cost optimization for dynamic replication and migration of data in cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 7, no. 3, pp. 705–718, 2019.

[34] W. Cerroni and F. Callegati, "Live migration of virtual network functions in cloud-based edge networks," in *in proceeding of International Conference on Communications (ICC)*, 2014, pp. 2963–2968.

[35] ETSI, "TS 136 331 - V15.3.0 - LTE; Evolved Universal Terrestrial Radio Access (E-UTRA); Radio Resource Control (RRC); Protocol specification." *3GPP TS 36.331 version 13.7.1 Release 13*, vol. 1, pp. 1 – 649, 2018. [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

[36] Y. Sun, G. Feng, L. Zhang, P. V. Klaine, M. A. Iinran, and Y.-C. Liang, "Distributed learning based handoff mechanism for radio access network slicing with data sharing," in *proceeding of IEEE International Conference on Communications (ICC)*. IEEE, 2019, pp. 1–6.

[37] S.-B. Lee, S. Choudhury, A. Khoshnevis, S. Xu, and S. Lu, "Downlink mimo with frequency-domain packet scheduling for 3gpp lte," in *proceeding of IEEE INFOCOM 2009*. IEEE, 2009, pp. 1269–1277.

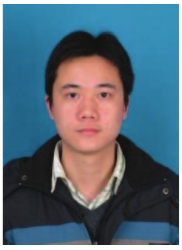
[38] H. Galeana-Zapién and R. Ferrús, "Design and evaluation of a backhaul-aware base station assignment algorithm for ofdma-based cellular networks," *IEEE transactions on wireless communications*, vol. 9, no. 10, pp. 3226–3237, 2010.
[39] A. Sbihi, "A best first search exact algorithm for the multiple-choice multidimensional knapsack problem," *Journal of Combinatorial Optimization*, vol. 13, no. 4, pp. 337–351, 2007.
[40] Z. Xu, L. Li, and W. Zou, "Exploring federated learning on battery-powered devices," in *Proceedings of the ACM Turing Celebration Conference-China*, 2019, pp. 1–6.
[41] A. Ignatov, R. Timofte, W. Chou, K. Wang, M. Wu, T. Hartley, and L. Van Gool, "Ai benchmark: Running deep neural networks on android smartphones," in *Proceedings of the European conference on computer vision (ECCV)*, 2018.
[42] H. B. McMahan, E. Moore, D. Ramage, S. Hampson *et al.*, "Communication-efficient learning of deep networks from decentralized data," *arXiv preprint arXiv:1602.05629*, 2016.
[43] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-Learning," *proceeding of 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 2094–2100, 2016.
[44] S. Wang, T. Tuor, T. Salonidis, K. K. Leung, C. Makaya, T. He, and K. Chan, "Adaptive federated learning in resource constrained edge computing systems," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1205–1221, 2019.
[45] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *arXiv preprint arXiv:1909.11315*, 2019.
[46] Z. Xu, J. Tang, J. Meng, W. Zhang, Y. Wang, C. H. Liu, and D. Yang, "Experience-driven networking: A deep reinforcement learning based approach," in *proceeding of IEEE INFOCOM - Conference on Computer Communications*, 2018, pp. 1871–1879.

Yi-Jing Liu received her B.S. degree in college of communication and information engineering from Chong Qing University of Post and Telecommunications, Chongqing, China, in 2017. She is currently pursing the Ph.D degree at the National Key Laboratory of Science and Technology on Communications, University of Electronic Science and Technology of China, Chengdu, China.

Her current research interests include next generation mobile networks, mobile edge computing, network slicing, and machine learning.

Gang Feng received his BEng and MEng degrees in Electronic Engineering from the University of Electronic Science and Technology of China (UESTC), in 1986 and 1989, respectively, and the Ph.D. degrees in Information Engineering from The Chinese University of Hong Kong in 1998. He joined the School of Electric and Electronic Engineering, Nanyang Technological University in December 2000 as an assistant professor and was promoted as an associate professor in October 2005. At present he is a professor with the National Laboratory of Communications, University of Electronic Science and Technology of China. Dr. Feng has extensive research experience and has published widely in computer networking and wireless networking research. His research interests include resource management in wireless networks, next generation cellular networks, etc. Dr. Feng is a senior member of IEEE.

Yao Sun received the B.S. degree in Mathematical Science, and the Ph.D. degree in Communication and Information System both from University of Electronic Science and Technology of China (UESTC), in 2014 and 2019, respectively. From Nov. 2017 to Nov. 2018, he was an international research visitor at School of Engineering, University of Glasgow. Dr. Sun is currently a research fellow at National Key Laboratory of Science and Technology on Communications, UESTC. Dr. Sun has extensive research experience and has published widely in wireless networking research area. He has won the IEEE Communication Society of TAOS Best Paper Award in 2019 ICC. His research interests include intelligent wireless networking, network slicing, blockchain system, internet of things and resource management in mobile networks.



Shuang Qin received the B.E. degree in Electronic Information Science and Technology, and the Ph.D degree in Communication and Information System from University of Electronic Science and Technology of China (UESTC), in 2006 and 2012, respectively. He is currently an associate professor with National Key Laboratory of Science and Technology on Communications in UESTC. His research interests include cooperative communication in wireless networks, data transmission in opportunistic networks and green communication in heterogeneous networks.

Ying-Chang Liang (F'11) is currently a Professor with the University of Electronic Science and Technology of China, China, where he leads the Center for Intelligent Networking and Communications and serves as the Deputy Director of the Artificial Intelligence Research Institute. He was a Professor with The University of Sydney, Australia, a Principal Scientist and Technical Advisor with the Institute for Infocomm Research, Singapore, and a Visiting Scholar with Stanford University, USA. His research interests include wireless networking and communications, cognitive radio, symbiotic radio, dynamic spectrum access, the Internet-of-Things, artificial intelligence, and machine learning techniques. Dr. Liang has been recognized by Thomson Reuters (now Clarivate Analytics) as a Highly Cited Researcher since 2014. He received the Prestigious Engineering Achievement Award from The Institution of Engineers, Singapore, in 2007, the Outstanding Contribution Appreciation Award from the IEEE Standards Association, in 2011, and the Recognition Award from the IEEE Communications Society Technical Committee on Cognitive Networks, in 2018. He is the recipient of numerous paper awards, including the IEEE Jack Neubauer Memorial Award, in 2014, and the IEEE Communications Society APB Outstanding Paper Award, in 2012. He was elected a Fellow of the IEEE for contributions to cognitive radio communications. He is the Founding Editor-in-Chief of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS: COGNITIVE RADIO SERIES, and the Key Founder and now the Editor-in-Chief of the IEEE TRANSACTIONS ON COGNITIVE COMMUNICATIONS AND NETWORKING. He is also serving as an Associate Editor-in-Chief for China Communications. He served as a Guest/Associate Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS, the IEEE Signal Processing Magazine, the IEEE TRANSACTION ON VEHICULAR TECHNOLOGY, and the IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORK. He was also an Associate Editor-in-Chief of the World Scientific Journal on Random Matrices: Theory and Applications. He was a Distinguished Lecture of the IEEE communications Society and the IEEE Vehicular Technical Committee on Cognitive Networks, and served as the TPC Chair and Executive Co-Chair of the IEEE Globecom'17.