Hindawi
Discrete Dynamics in Nature and Society
Volume 2022, Article ID 4955498, 11 pages
https://doi.org/10.1155/2022/4955498

Research Article

Edge Intelligence-Based RAN Architecture for 6G Internet of Things

Yang Liu, Qingtian Wang, Haitao Liu, Jiaying Zong, and Fengyi Yang

China Telecom Research Institute, Beijing 102209, China

Correspondence should be addressed to Qingtian Wang; wangqt08@chinatelecom.cn

Received 14 July 2022; Accepted 26 September 2022; Published 15 November 2022

Academic Editor: Bo Rong

Edge Intelligence, which blends Artificial Intelligence (AI) with Radio Access Network (RAN) and edge computing, is recommended as a crucial enabling technology for 6G to accommodate intelligent and efficient applications. In this study, we proposed Edge Intelligent Radio Access Network Architecture (EIRA) by introducing new intelligence modules, which include broadband edge platforms that allow policies to interact with virtualized RAN for various applications. We also developed a Markov chain-based RAN Intelligence Control (RIC) scheduling policy for allocating intelligence elements. Experimental results justified that the virtualized RAN delivers on its performance promises in terms of throughput, latency, and resource utilization.

1. Introduction

The Internet of Things (IoT) contains a bunch of Information sensors, radio frequency identifiers, global positioning devices, infrared sensors, and lasers. To realize the pervasive connection between objects and people, a device must first gather the necessary data, which can include sound, light, heat, electricity, mechanics, chemistry, biology, location, and so on, and then complete the process of intelligently perceiving, identifying, and communicating.

The IoT is an information carrier based on the Internet, traditional telecommunication networks, etc. It enables all common physical objects that can be independently addressed to form an interconnected network. The future IoT will have a deep economical, commercial, and social impact on our lives. The IoT integrates billions of smart devices that can communicate with one another with minimal human intervention. However, the crosscutting nature of IoT systems and the multidisciplinary components involved in the deployment of such systems have introduced new security challenges. Implementing security measures, such as encryption, authentication, access control, and network and application security, for IoT devices and their inherent vulnerabilities is ineffective.

From another point of view, IoT also refers to a network of interconnected computing devices and other endpoints that can exchange data with one another and with other devices and systems over the Internet. Using it, any set of individually addressable everyday physical things can be turned into a distributed system. Economically, commercially, and socially, the future IoT will have profound effects on our lives. The IoT connects billions of devices that may exchange data with one another autonomously, requiring just minimal human oversight. Approximately 50 billion devices have been connected to the Internet by the end of 2020, making IoT one of the fastest-growing sectors in the history of computing.

Cellular network-enabled IoT will make a revolution in our future life. It will pave the way for advanced wireless systems and innovative new services [1]. The widespread implementation of the Internet of Things relies on several enabling technologies, such as 5G and 6G. Recently, the fifth-generation (5G) networks have been globally deployed in practice, while it is still predicted that the 5G may not be able to satisfy the demand of an increasing number of future communications [2, 3]. Researchers from both industry and academia are focusing on the sixth-generation (6G) networks. Compared with 5G, 6G networks not only with the features of high synchronization accuracy, near

100% coverage, and higher bandwidth [4] but also with cloud computing and Artificial Intelligence (AI), especially with the interaction of RAN and computing. Edge Intelligence (EI) is being considered an enabling technology for 6G [5], especially in Radio Access Network (RAN) area, because EI pushes the AI algorithms on the devices or network edge [6]. It is a benefit for the application to be processed with low latency and RAN resources scheduled on the edge.

Regarding the advantages of ultra-low latency and high bandwidth offered by edge computing in the 5G era, edge computing had attracted lots of attention; specifically, edge computing handles the applications by migrating the cloud computing ability into the edge. While at the current 5G, edge computing and network functions are still not merged deeply, this is because edge computing is hard to obtain real-time information from RAN [7]. In addition, the network functions are not beneficial from edge computing to improve the intelligence of RAN in network operation and maintenance. Therefore, edge intelligence has received attention from 6G, and AI has been applied as solutions in the network including radio resource management, mobility management, and orchestration [8]. For example, reinforcement learning is applied in the network to decide on network planning, and federated learning is used to protect the privacy of data. AI not only optimizes the communication resource but also configures the network adaptively and responds quickly [9].

For the vision that EI merges AI into edge computing at the network edge, specifically, EI can be considered as one of the evolution routes for AI natives. On the one hand, EI schedules the computing resource for applications; on the other hand, the resources of RAN can also be orchestrated by EI. For 6 G, networks need to support the new service with more stringent and diverse Quality of Service (QoS) requirements and mass connectivity. Hence, it is necessary to construct EI in future 6G networks.

In this paper, we intend to present the Edge Intelligence-based RAN Architecture (EIRA) towards 6G and build a testbed that takes advantage of both intelligence and virtualization. Specifically, we apply micro-services technologies for orchestrating computing and storage resources. Moreover, virtualization technologies are applied to implement the virtualized RAN. In the experiment, we build the testbed and evaluate the performance of functionalities in the testbed, for virtualized RAN, and intelligence use cases. This paper makes the following contributions:

(i) We present an Edge Intelligent RAN Architecture (EIRA) towards 6G

(ii) We propose extensive and custom edge platforms that embedded AI algorithms to process the applications with various latency and computing resource requirements

(iii) We design a RAN Intelligence Control (RIC) resource scheduling policy between extensive edge platform and custom edge platform to improve the request accepted ratio

(iv) We conduct a testbed in practice and evaluate the performance of network functionalities and intelligence use cases

The remainder of this paper is organized as follows. Section 2 reviews the existing work in the related research field, then Section 3 discusses the architecture of our proposed EIRA. In Section 4, we propose a RIC scheduling policy in the EIRA, followed by experimental results illustrated in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Edge Intelligence is still in its developing stage; in this section, we first survey edge intelligence and attempt to give the initial vision of edge intelligence. Then, we outline the artificial intelligence algorithms applied in the RAN to enhance performance, i.e., reinforcement learning and federated learning. Last, we investigate the architecture for edge intelligence.

As the most promising solution for 6G, edge intelligence has attracted significant attention, and an increasing number of studies about edge intelligence have been proposed in recent years [10–12]. In June 2020, the University of Oulu released the 6G white paper on edge intelligence [10]. Peltonen et al. discuss the infrastructure and platforms for edge computing and present the seven levels of edge intelligence. They also comprehensively analyze the key enablers for edge intelligence for 6G. Additionally, they outline the future directions of 6G edge intelligence. Liu et al. [11] provide an overview of Multiaccess Edge Computing (MEC) in 5G and IoT and discuss edge intelligence in 5G and IoT. They propose a use case named proximity detection with edge intelligence in an IoT environment. In order to reduce the communication cost, they apply five kinds of neural networks and observe that the Long Short-Term Memory neural network and Gated Recurrent Unit have the best prediction accuracy. Nguyen et al. [12] review the fundamental 6G technologies for IoT and discuss the roles of 6G in IoT applications.

In order to get the potential performance improvement in 5G or the upcoming 6G, some works focus on applying AI techniques in 6G. The study in [13] proposes an intelligent reflective surface-based 6G wireless network infrastructure for energy-efficient and sustainable 6G development. Li et al. [14] propose a deep reinforcement learning approach to optimize the coverage ratio in 6G-based IoT networks. They first present a genetic algorithm to maximize the data coverage ratio and then apply deep reinforcement learning to optimal route policy. The experiments show that the proposed method can reduce the length of the collection path and cost. She et al. [15] develop an architecture enabling device intelligence, edge intelligence, and cloud intelligence to achieve Ultra-Reliable and Low Latency Communication in 6G networks. The authors apply Deep Neural Networks for training and federated learning for improving learning efficiency. Compared with the two algorithms and optimal performance, the experiments show that the proposed deep learning approach shows better performance than the

compared algorithms and is close to optimal. Prathiba et al. [16] propose federated learning for computation offloading and resource management within heterogeneous systems. To save the resource cost and improve available resource utilization, they present the federated Q-Learning.

We forward our focus on the studies about AI-enabled architecture for 6G. Han et al. [17] propose an AI-enabled RAN architecture, and then they design a series of functions to maximize the potential gain. In addition, they implement the transceiver with AI and verify that the constellations designed by AI provide a better error rate than the conventional Quadrature Amplitude Modulation constellations. Yang et al. [18] present a four-tier AI-enabled architecture for 6G to fulfill smart resource management, automatic network adjustment, etc. They also discuss edge intelligence in edge computing and cloud AI in the central cloud. Xu et al. [19] demonstrate the machine-learning-based cyber twin architecture for the 6G-enabled Industrial Internet of Things. They, then present a deep reinforcement learning approach to evaluate systematic trial and error in the cyber twin world. The experiment results show that the proposed system has better performance in terms of computing delay and communication delay.

From the review above, it can be seen that the existing research works on edge intelligence are in infancy; moreover, the AI algorithms are applied as a solution in resource allocation, computation offloading for RAN, and so on. However, the previous studies do not consider establishing an overall architecture for edge intelligence-based RAN and building a testbed to promote 6G-related research. To this end, in the paper, we propose a testbed architecture for 6G IoT and deploy it in practice.

3. System Architecture

In this section, the proposed Edge Intelligent RAN Architecture (EIRA) towards 6G is illustrated, as shown in Figure 1. In EIRA, virtualization and service-based architecture are applied to orchestrate the network elements statelessly. We then introduce the potential deployment plan for EIRA.

EIRA is a four-layer architecture including the Ubiquitous Connection layer, Cloud-Network Resource Pool layer, Edge Intelligence Cloud Platform (PICP) layer, and Applications layer. In particular, the Edge Intelligence Cloud Platform layer is used to achieve intelligence native to inner circulation within the architecture. Especially, the PICP obtains the information from RAN, and then, the PICP analyzes the information and makes the decision to RAN in an intelligent way.

3.1. Ubiquitous Connection Layer. This layer supports multiple access styles including Terrestrial and Non-terrestrial networks. The User Equipment (UE) accesses this layer via a terrestrial style, such as 6G, optical, Wireless Local Access Network (WLAN), or nonterrestrial style like the satellite.

In order to strengthen transmission efficiency and network performance, Cell-Free massive Multiple-Input Multiple-Output (CF mMIMO) is considered in this layer. CF mMIMO removes the concept of cellular or cell and introduces the user-centric design. It is also beneficial for high network connectivity and coverage, huge spectrum, and energy efficiency.

3.2. Cloud-Network Resource Pool Layer. The layer contains heterogeneous resource pools and resource virtualization. A heterogeneous resource pool consists of general and dedicated resources with centralized management. General resources contain common and standardized hardware (i.e., industrial servers based on X86 or ARM CPU) and diversified hardware chips with scalability, including acceleration and clock resource chips. For RAN, high-speed processing and a large number of dedicated resources are required, such as Field Programmable Gate Array (FPGA) for coding and encoding. The clock resources are applied to fulfill synchronization accuracy among network elements and UEs. Dedicated resources (e.g., ASIC chips) provide specialized services for a small number of facilities with large capacities and ultra-high performance requirements.

The resource virtualization part abstracts these resources into virtual resources, i.e., virtualized Computing (vComputing), virtualized Storage (vStorage), virtualized Networking (vNetworking), and virtualized Accelerating (vAccelerating). This part eliminates the difference between general resources and dedicated resources; furthermore, it has a unified view of the heterogeneous resources.

3.3. Edge Intelligent Cloud Platform. On the Edge Intelligent Cloud Platform, the AI models are trained and deployed. Regarding the requirements (i.e., latency and computing abilities) of 6G scenarios, we divide the Edge Intelligent Cloud Platform into a Custom Edge Platform and an Extensive Edge Platform. Specifically, Near Real-Time RAN Intelligence Control (Near-RT RIC) is applied in custom Edge-Platform, and Non Real-Time RAN Intelligence Control (Non-RT RIC) is in Extensive Edge-Platform.

Custom Edge-Platform consists of Service-Based Architecture RAN (SBA RAN), Lightweight Core Nets, Trained AI Model, and Near-RT RIC. Particularly, those components are deployed in this platform as Network Functions. SBA RAN is considered in this platform to realize flexible and rapid deployment of RAN. Lightweight Core Net holds a key role in realizing the full potential of 6G services. To process the application in the intelligent method, the trained AI models are embedded in the Edge Intelligent Cloud Platform. Once the requirements are from UEs, the Near-RT RIC deployed in the Edge Intelligent Cloud Platform chooses the trained AI model to process the application. In particular, Near-RT RIC processes the application within low latency.

The Extensive Edge Platform contains Security Capability, Unified Application Programming Interface (API), Date collector, AI model, and Non-Real Time RAN Intelligent Controller (Non-RT RIC). Security Capability protects the processed, transmitted, and stored information in the EIRA. Functions and the Extensive Edge Platform

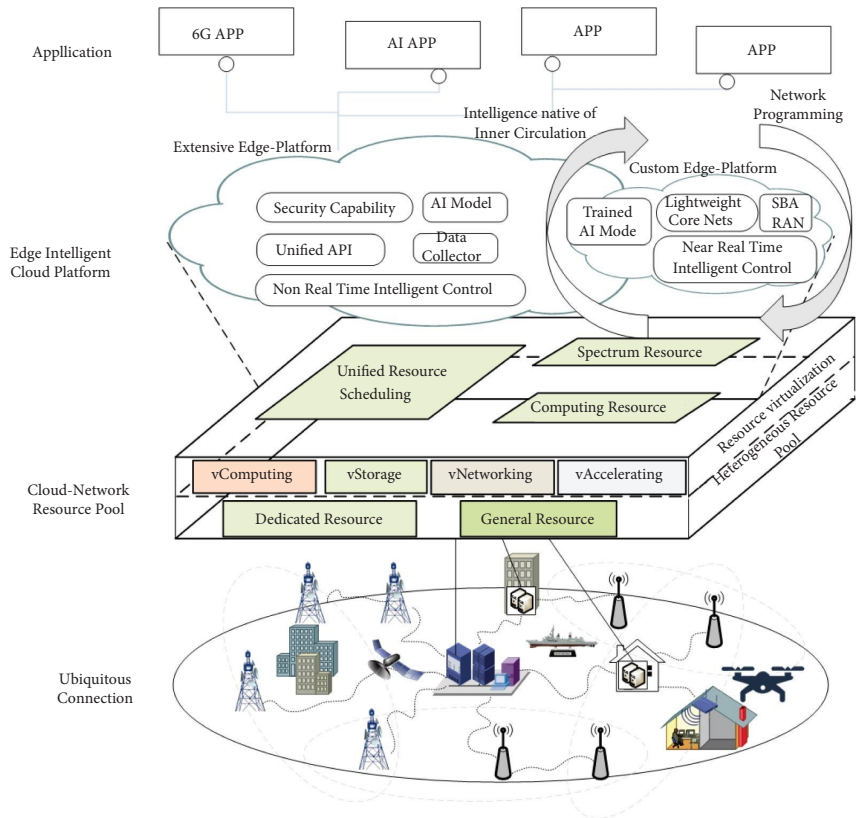

Figure 1: 6G edge intelligent RAN architecture (EIRA).

communicate with each other through a Unified API. Non-RT RIC has features with data collector and model training. The Non-RT RIC collects the network and application data via Unified API, and models are trained in AI models. Moreover, the Non-RT RIC communicates with Near-RT RIC via the Unified API. On one hand, Non-RT RIC sends the trained AI model to the Near-RT RIC; on the other hand, the Near-RT RIC returns feedback to the Non-RT RIC about the updated AI model. Consequently, Near-RT RIC and Non-RT RIC control the RAN with an intelligent method, furthermore making RAN programmable.

3.4. Applications Layer. The future 6G scenarios are deployed in the application layer, and we provide the Unified API for 3rd party to fulfill their application. In Figure 1, we list some applications, such as 6G APP and AI APP.

3.5. Practical Deployment. In order to apply our proposed EIRA into practice, we illustrate the potential deployment plan for EIRA, as shown in Figure 2. As usual, the Extensive Edge Platform could be deployed in the rich computing resource, because it needs to train the AI models. The Custom Edge Platform closes the user to process the application within the requirements of latency. Figure 2 shows the distributed deployment for EIRA, the Extensive Edge Platform in the regional cloud, and Custom Edge Platform in the edge cloud. Especially, an Extensive Edge Platform connects multiple Custom Edge Platforms via wired access style (e.g., optical network). Extensive Edge Platform is able to send the trained AI models to a Custom Edge Platform or multiple Custom Edge Platforms. In addition, Extensive Edge Platform and Custom Edge Platform can also be deployed together at the edge cloud, and it depends on the demand of users.

4. RIC Resource Scheduling Policy

In order to save the storage resource in Custom Edge Platform and improve the acceptance ratio of application requests, in this section, we introduce a scheduling policy-based Markov chain between Non-RT RIC and Near-RT RIC.

4.1. System Model. As shown in Figure 2, we consider the Custom Edge Platform is deployed in the edge cloud and close to the user. The users access the Custom Edge Platform via RAN. The Extensive Edge Platform in the regional cloud connects with multiple Custom Edge Platforms via optical framework. In particular, a Non-RT RIC connects with m Near-RT RICs by a set $\{m = 1, 2, 3, \ldots, M\}$. A Near-RT RIC covers n users denoted as set $\{n = 1, 2, 3, \ldots, N\}$. Let the service requests to a Near-RT RIC be represented by a set $R = \{r_1, r_2, \ldots r_j\}$. Let $r_i = \{\alpha_i, \beta_i, \gamma_i, \theta_i, c\}$ denote a service i request, where α_i is a kind of AI model in xApp that is requested, β_i is the requested process time, γ_i is the access point of the RAN, θ_i is the size of the request, and c_j is the requested computational resource. We assume that only an

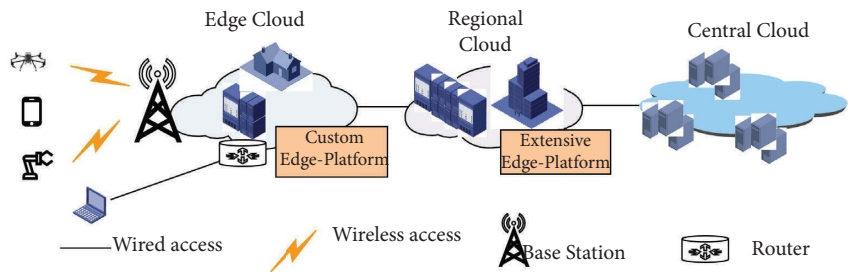

Figure 2: Practical EIRA framework in deployment.

AI model is in an xAPP. In order to maximize the acceptance ratio of the request, we need to predict the AI models of requests in the Near-RT RIC. Therefore, which kinds of xAPP deployed in the Near-RT RIC should be predicted. The acceptance ratio a is calculated as follows:

$$a = \frac{R_a}{R_t}, \tag{1}$$

where R_a is the accepted request and R_t is the total request.

For an accepted request, the processing time should be less than the requested process time β_i. The processing time consists of access time and calculating time.

$$\begin{aligned} &\beta_i \le t_i, 1 < i < j, \\ &t_i = t_{\text{access}} + t_{\text{calculatingtime}}. \end{aligned} \tag{2}$$

The access time is

$$\begin{aligned} t_{\text{access}} &= \frac{\theta_i}{b}, \\ b &= \text{B}\log_2(1+\lambda), \end{aligned} \tag{3}$$

where θ_i is the size of the request, B is the bandwidth of the wireless link and λ is the signal-to-inference noise ratio (SNR) of the wireless link.

The calculating time is

$$\begin{aligned} t_{\text{calculatingtime}} &= \frac{c_p}{c_i}, \\ c_p &= c_{\text{total}} - c_{\text{occupied}}, \end{aligned} \tag{4}$$

where c_p is the provided computational resource, c_{total} is the total computational resource that Custom Edge Platform can provide, and c_{occupied} is the occupied computational resource.

Let us consider a Near-RT RIC, the xAPPs set is denoted as $C = \{c_1, \ldots, c_s\}$. We assume that all the xApps in the Near-RT RIC have the same size. Suppose that a Near-RT RIC contains S xApps and a Non-RT RIC contains L AI models. The M xApps are stored in the Near-RT RIC, because of the limited storage resource, $M < S < L$. Let $P = \{p_1, \ldots, p_s\}$ be the set of xAPP popularity for the users, $0 \le p_k \le 1$, $k \in [1, s]$. We count the total acceptance number of requests within a period and record the acceptance requests for xAPP c_k; thereby, the p_k is shown as follows:

$$P_k = \frac{R_{a,k}}{R_{a,S}}, \tag{5}$$

where the number of the acceptance requests for c_k could be defined by $R_{a,k}$ and the total acceptance requests could be defined by $R_{a,S}$.

The xAPPs replacement strategy is the Least Recently Used (LRU). According to the LRU, the number of xAPPs of the Near-RT RIC is regarded as a stack of length $s+1$, where position 1 is at the top of the stack and position s at the bottom of the stack. Whenever a request is accepted, the least requested xApp at the bottom position is replaced and the object xAPP is placed at the top position.

We assume that t-th xApp is at position l of the stack. When a request is accepted by the Near-RT RIC, the following three changes are then possible:

(i) l will be moved to the top, which means that the t-th xApp is called to process the request

(ii) l will remain at the same position if the xAPP with position b is called, and $b < l$, b ,$l \in [1, s+1]$

l will be moved down by one position if the xAPP with position d is called, and $l < d$, d ,$l \in [1, s+1]$

4.2. LRU Model. Let us consider a Markov chain $\{X_{s+1}\}$ with $s+1$ states, as shown in Figure 3. This chain represents the state transition for xApp, where state 1 means that the object is the most frequently called and state s means that it is the least called. The state $s+1$ means that the called xApp is absent. According to reference [20], the $\{X_{s+1}\}$ is irreducible and aperiodic and $X_1, \ldots X_s$ are independent of each other. The probability of t-th xAPP at the position l is

$$P^l = p_k, \tag{6}$$

where p_k is the popularity of t-th xAPP.

The probability of position l to 1 is

$$\begin{aligned} P^{l \longrightarrow 1} &= p_k, \\ l_t &\in [1, s+1]. \end{aligned} \tag{7}$$

The probability of state l to $l+1$ is equal to the sum of xApp from $l+1$ to s

$$\begin{aligned} P^{l \longrightarrow l+1} &= \sum p_k, \\ l_t &\in [1+1, s]. \end{aligned} \tag{8}$$

Particularly, if $l = s+1$,

$$P^{l \longrightarrow l+1} = 1 - P^l. \tag{9}$$

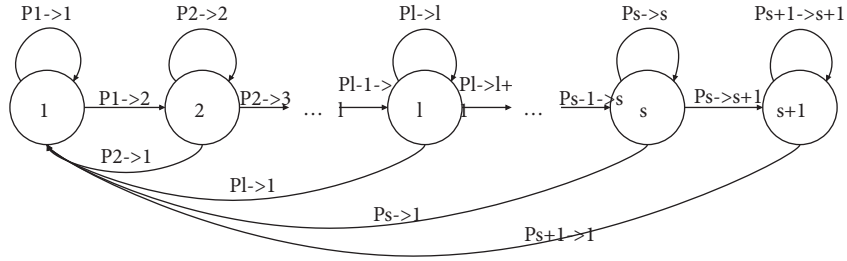

Figure 3: The state transition.

Input: requests, S xApps
Output: M xApps
(1) Calculate the acceptance requests by the subject (2)
(2) Calculate the popularity of S xApps by (5)
(3) Calculate the state transition probability by (8)–(11)
(4) Construct the transition matrix P
(5) Find the stationary distribution π by Chapman-Kolmogorov equations
(6) Rank the element π_i increasingly
(7) Output the first M π_i

Algorithm 1: RIC resource scheduling policy.

The probability of state l to $l-1$ is equal to the sum of xApp from l to l.

$$P^{l\longrightarrow l} = \sum p_k, \quad l_k \in [1, l-1], \quad l \in [2, s]. \tag{10}$$

We construct the transition matrix P, where the row and column of the matrix are $s+1$ and $s+1$, as follows:

$$P = \begin{bmatrix} p_{11} & p_{12} & 0 & 0 & \dots & 0 \\ p_{21} & p_{22} & p_{23} & 0 & \dots & 0 \\ p_{31} & 0 & p_{33} & p_{34} & \dots & 0 \\ & \vdots & & & & \\ p_{l1} & \cdots & p_{ll} & p_{ll+1} & \dots & 0 \\ & \vdots & & & & \\ p_{s+1\,1} & 0 & 0 & \dots & 0 & p_{s+1s+1} \end{bmatrix}. \tag{11}$$

According to Chapman–Kolmogorov, we have

$$\pi_i = \sum_{j}^{s+1} \pi_j P_{j,i} 1 \le i \le s+1, \tag{12}$$

$$\sum_{i=1}^{s+1} \pi_i = 1. \tag{13}$$

Then, the stationary distribution π can solve (12) and (13). For the Near-RT RIC, we calculate the stationary of xApps and then sort the π_i increasingly. At last, we deploy M xApps in the Near-RT RIC with high values of M π_i. The pseudo is given in Algorithm 1.

5. Experimental Results

5.1. Experimental Environment. This section builds an experimental testbed of the proposed Edge Intelligence-based RAN architecture. We evaluate the overhead of network performance with real-time kernel latency and CPU resource usage. Figure 4 shows the testbed hardware devices. Since it is the initial period to discuss the 6G RAN design, the virtualized RAN is established based on 5G NR and 5GC, which can be later updated to 6G RAN. The performance of network latency and throughput is verified within EIRA. In order to evaluate the proposed RIC resource scheduling policy, we consider five application scenarios and set the value of $M = 2$. After this, we analyze the performance of face detection on the platform.

The testbed consists of an all-in-one server, a switch, and an RRU, and the virtualized BBU (vBBU)/lightweight virtualized 5 GC (v5GC) are deployed on the edge intelligence platform in the all-in-one server as virtualized network functions (VNFs). Two same commercial terminals Huawei Mate30 are used. In the test, terminals access the testbed via a 5G NR air interface, and the RRU connects with a switch via optical hybrid cable; moreover, the switch connects with an all-in-one server via optical fiber. The transmission configuration of RAN, the physical configuration of an all-in-one server, and the configuration of the commercial terminal are displayed in Tables 1–3, respectively.

The test is executed in uplink (UL) with the frequency band 3.5 GHz, the bandwidth 100 MHz, and the transmission mode of Time Division Duplex (TDD). Particularly, the frame structure adopted is DSUUU (1D3U) with a 2.5-millisecond single period to achieve the aim of large-capacity transmission for UL, and the number of UL layers is 2. In

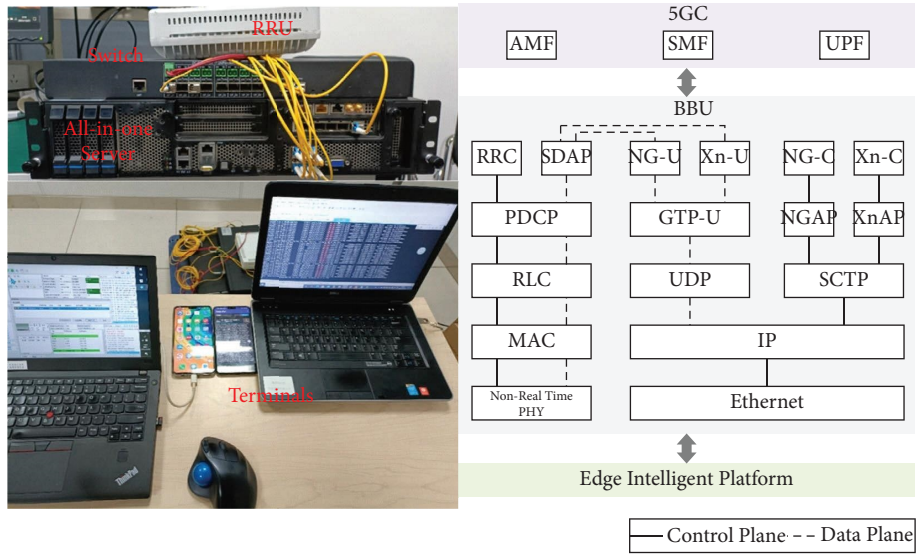

Figure 4: EIRA hardware testbed with a 5G NR protocol stack.

Table 1: Transmission configurations of RAN.

Parameters	Values
Carrier frequency	3.5 GHz
System bandwidth	100 MHz
Transmission mode	TDD
Frame structure	2.5 ms single period DSUUU
Transmission/receiving antennas	2/2
Number of layers for UL	2

Table 2: Physical configurations of the all-in-one server.

Parameters	Configurations
Hardware	Standard 2U server
CPU	2 Xeon(R) silver 4216 CPUs @2.1 GHz 64 RAM
Operating system	CentOS real-time kernel system
Deployed applications	5G vBBU, v5GC (AMF + SMF + UPF), iperf, face detection

addition, the all-in-one server in this experimental environment is a standard 2U server, which has 2 Xeon(R) Silver 4216 CPUs, and two cards of accelerators for channel encoding/decoding and fronthaul processing separately are applied. For the operation system, the CentOS real-time kernel operating system is used to meet RAN real-time requirements. The applications deployed on the all-in-one server with container include iPerf and face detection, with the same deployment modes of vBBU and v5GC and the testbed supports for scaling of the container.

5.2. Evaluation of Network Functionalities. In the experiments, the Key Performance Indicator of real-time kernel latency is used to evaluate the performance of vBBU, which is because real-time kernel latency can reflect the jitter for threaded interrupts, and it can show the processing performance for vBBU. To satisfy the real-time requirement of vBBU, eight CPU threads for parallel computing are used, and the statistics of real-time kernel latency are shown in Figure 5. The duration time for statistics is two minutes, and the statistics results of each thread are basically below 13 us, which satisfies the maximum latency of 20 us in the standard of O-RAN [21].

Table 3: Configurations for the commercial terminal.

Parameters	Configurations
CPU	8-0043ore 2 × Cortex-A76 @ 2.86 GHz + 2 × Cortex-A76 @ 2.09 GHz + 4 × Cortex-A55 @1.86 GHz
GPU	16-Core Mali-G76
NPU (neural network processing unit)	Dual-core NPU
Operating system	EMUI10
Storage	8 GB RAM + 128 GB ROM

The all-in-one server consists of two 16-core CPUs (i.e., 32 physical cores) to deploy an edge intelligence-based platform, vBBU, v5GC, and other components. Hyperthreading technology is applied to generate 64 logical cores. Table 4 lists the usage of CPU when the vBBU is unloaded (i.e., UE access the network and no service is performed) and the vBBU is fully loaded (i.e., UE access the network and uplink service is performed), respectively, and the results are an average of 20 runs. The CPU resource occupied by vBBU at full load is higher than that in the case of no-load because the physical layer is required for packet processing and scheduling at full load. While the resource occupation of v5GC for no-load and full-load is more or less the same, since DPDK is applied for fast packet processing, that is, DPDK continuously occupies the corresponding CPU resources. Due to the fewer number of test users, the resources occupied by v5GC are the same in the two cases. When the

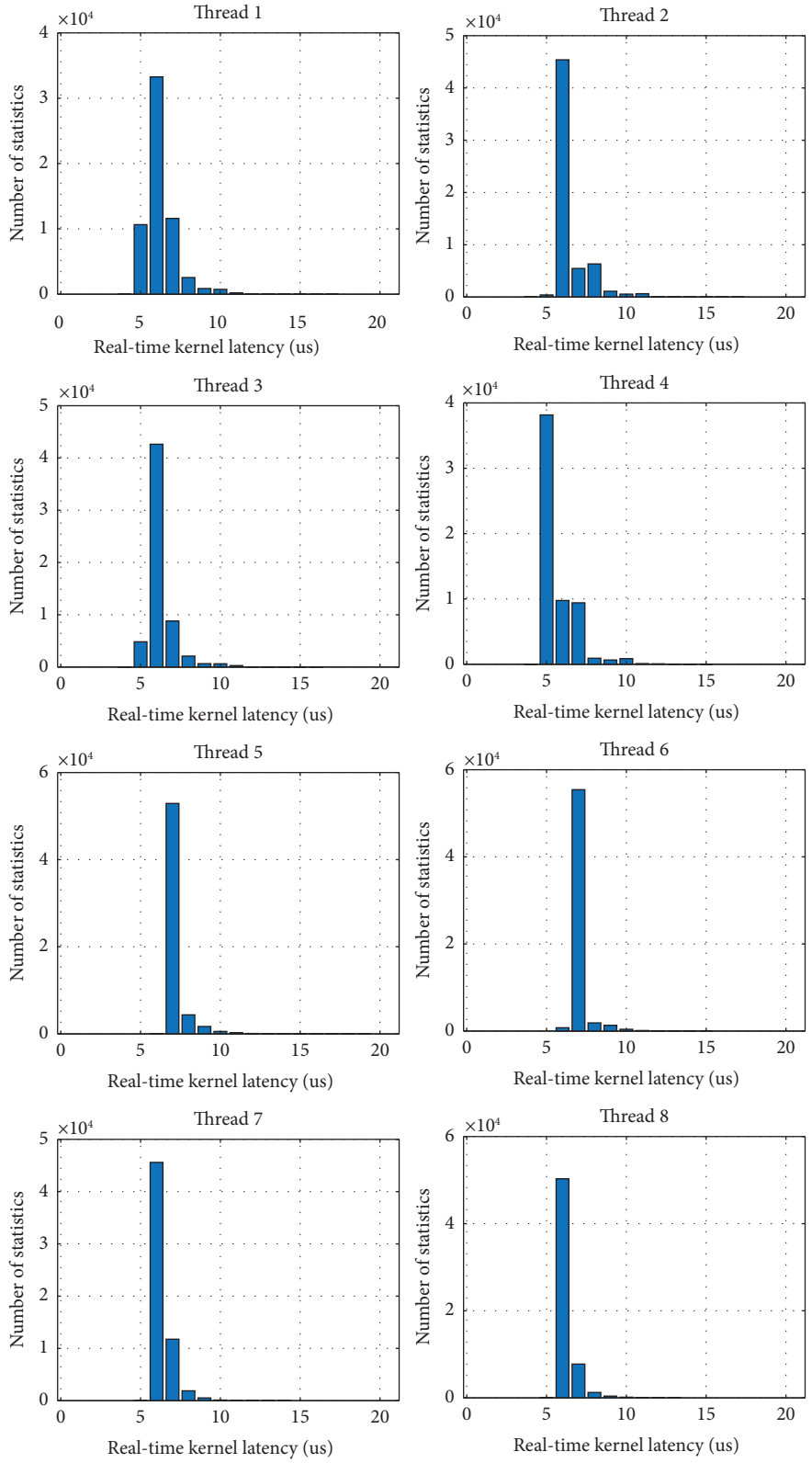

Figure 5: Statistical performance of real-time kernel latency for eight CPU threads.

Table 4: Statistics of CPU resource usage.

Number of CPU cores occupied	vBBU	v5GC	Platform and other components	Total
No-load vBBU	3.90	4.53	2.92	11.35
Full-load vBBU	6.59	4.53	3.47	14.59

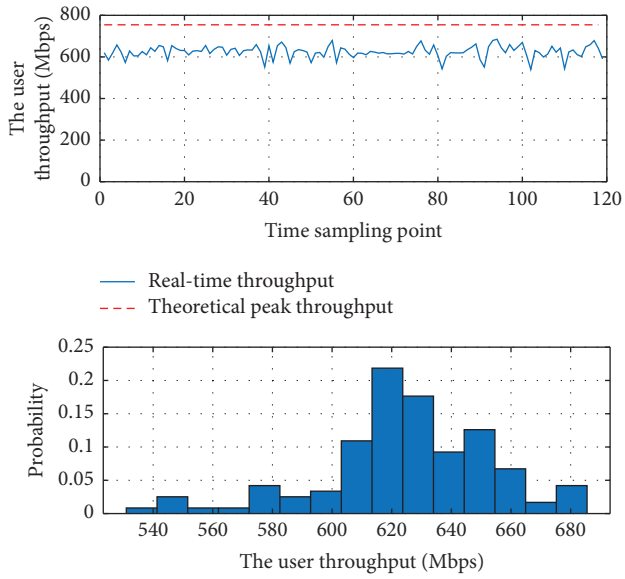

Figure 6: UL rate sampling and probability distribution of single users.

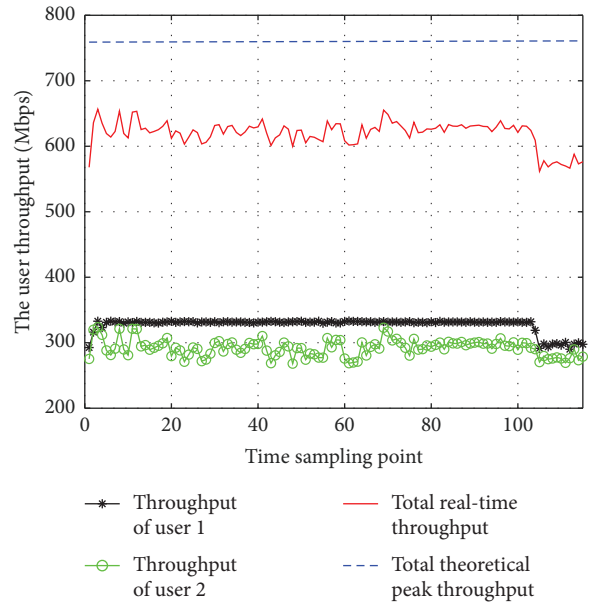

Figure 7: UL rate sampling of two users.

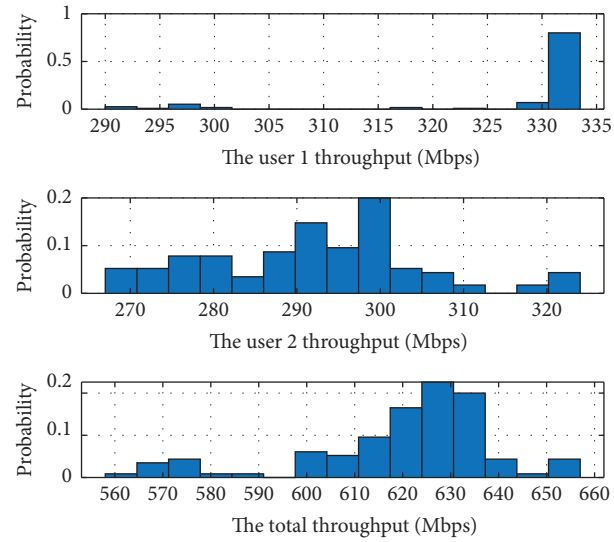

Figure 8: UL rate probability distribution of two users.

vBBU is fully loaded, the CPU resource occupied by the other components is higher than that when the vBBU is not loaded, owing to the increased resource occupied by vBBU containers and control plane components.

Figure 6 shows the sampling curve and probability distribution of the uplink rate of a single user. We can see the values in rate are mainly among 620–660 Mbps, and the average rate is about 624.54 Mbps. In addition, the theoretical peak rate is 760 Mbps based on our configurations, and the actual maximum peak rate during the testing is 684.95 Mbps, which can be 90% of the theoretical peak rate. The theoretical peak value formulation is calculated as follows [22]:

$$\text{data rate} = 10^{-6} * \sum_{j=1}^{J} \left\{ v_{\text{Layers}}^{(j)} * Q_m^{(j)} * R_{\max} * \frac{N_{PRB}^{BW(j),\mu} * 12}{T_s^{\mu}} * \left(1 - OH^{(j)}\right) \right\}. \tag{14}$$

The sampling and probability distribution of the uplink rates with the two users are depicted in Figures 7 and 8, respectively. The peak rate of the first user is 333.19 Mbps, and the average rate is 327.61 Mbps. While the peak rate of the second user is 323.73 Mbps, and the average rate is 292.04 Mbps. The sum of peak rates for the two users is 656.92 Mbps. It can be seen that the peak rate of two users is lower than that of a single user, which is because the overhead of two users is greater than that of a single user. We note the rate of user 2 is lower than that of user 1 and the throughput of user 2 has a fluctuation from time sampling point 0 to 100 because there is more interference in the test environment for user 2.

Table 5 shows the test results of Round Trip Time (RTT) between the terminal and the vBBU. In Radio Resource Control (RRC) connected state, we use the Internet Control Message Protocol (ICMP) with the command PING to test the RTT. The packet size is 32 bytes and 1500 bytes, respectively, and we run 100 times continuously. The results reveal that the average delay of 1500-byte packets is slightly

Table 5: Statistics of user plane RTT.

RTT (ms)	Ping 32 bytes	Ping 1500 bytes
Minimum value	6.677	6.355
Maximum value	10.428	13.276
Average value	7.988	7.993
The success rate of the ping packet	100%	100%

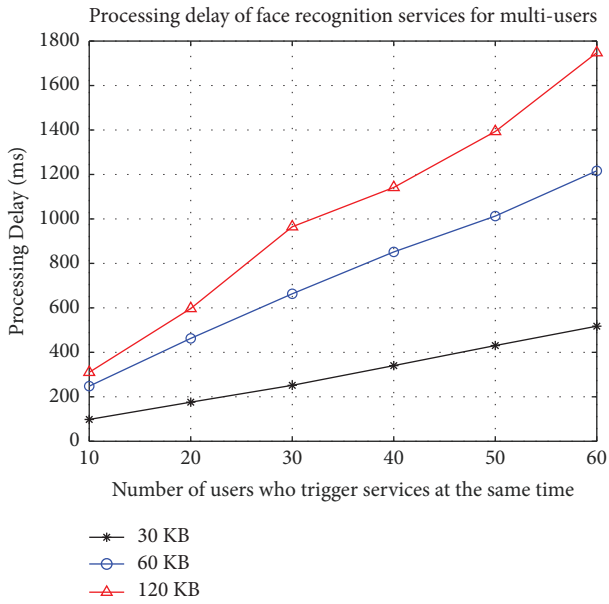

Figure 9: The processing time for intelligent application.

larger than that of 32-byte packets because the testbed processes short packets faster than a long-size packet.

5.3. Evaluation of Intelligence Case. In order to evaluate the performance of an intelligent case, a face detection application is deployed in the testbed in the form of a Pod. It is considered using OpenCV [23], and we use three images with the size of $1024 * 768$ pixels as input data from terminals for face detection. The volumes of the three images are 30 kB, 60 kB, and 120 kB. In Figure 9, the X-axis is the number of images that the testbed received from terminals. The Y-axis is the processing time for the images. The black line is for the processing time of 30 kB, the blue line is for 60 kB, and the red line is for 120 kB. Figure 9 shows that the processing time increases from 300 ms when the number of services is 10 to 1770 ms when the number of services is 60 at the volume of 120 KB. We can see that the trend of processing time is increasing, and the larger images require more processing time.

6. Conclusions

This paper presents the edge intelligence-based RAN architecture towards 6G with a lab environment testbed for EIRA provided to make it reliable, efficient, and smart. In EIRA, virtualized network functions are deployed and the intelligence modules are implanted in extensive Edge Platform and Custom Edge Platform respectively, interacting with virtualized RAN for different applications to build RAN open and programmable. Specifically, we propose an approach for allocating resources between the extensive Edge-Platform and Custom Edge Platform. Our proposed work is unique in the sense that it is one of the initial efforts to build an edge intelligent RAN architecture with a lab-scale testbed supporting all key features.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Key R&D Program of China under Grant 2020YFB1806700. The authors wish to thank Intel for advice on testbed design and Comba Telecom Systems Holdings Limited for experimental support.

References

[1] Z. Wang, R. Liu, Q. Liu, J. S. Thompson, and M. Kadoch, "Energy-efficient data collection and device positioning in UAV-assisted IoT," *IEEE Internet of Things Journal*, vol. 7, no. 2, pp. 1122–1139, 2020.

[2] L. Chen, F. R. Yu, H. Ji, B. Rong, X. Li, and V. C. M. Leung, "Green full-duplex self-backhaul and energy harvesting small cell networks with massive MIMO," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 12, pp. 3709–3724, 2016.

[3] A. Nessa, M. Kadoch, and B. Rong, "Fountain coded cooperative communications for LTE-A connected heterogeneous M2M network," *IEEE Access*, vol. 4, pp. 5280–5292, 2016.

[4] X. H. You, C.-X. Wang, J. Huang et al., "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences*, vol. 64, no. 1, Article ID 110301, 2021.

[5] E. Peltonen, M. Bennis, M. Capobianco et al., " 6G white paper on edge intelligence," *Artificial Intelligence, arXiv preprint arXiv:2004.14850*, 2020.

[6] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: applications, trends, technologies, and open research problems," *IEEE Network*, vol. 34, no. 3, pp. 134–142, 2020.

[7] N. Chen, B. Rong, X. Zhang, and M. Kadoch, "Scalable and flexible massive MIMO precoding for 5G H-cran," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 46–52, 2017.

[8] R. Li, Z. Zhao, X. Zhou et al., "Intelligent 5G: when cellular networks meet artificial intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, 2017.

[9] C. X. Wang, M. D. Renzo, S. Stanczak, S. Wang, and E. G Larsson, "Artificial intelligence enabled wireless networking for 5G and beyond: recent advances and future challenges," *IEEE Wireless Communications*, vol. 27, no. 1, pp. 16–23, 2020.

[10] E. Peltonen, M. Bennis, M. Capobianco et al., "6G white paper on edge intelligence," 2020, https://arxiv.org/abs/2004.14850.

[11] Y. Liu, M. Peng, G. Shou, Y. Chen, and S Chen, "Toward edge intelligence: multiaccess edge computing for 5G and Internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 6722–6747, 2020.

[12] D. C. Nguyen, M. Ding, P. N. Pathirana, S. Aruna, J. Li, and D. Niyato, "6G Internet of Things: a comprehensive survey," *IEEE Internet of Things Journal*, vol. 9, 2021.

[13] Q. Liu, S. Sun, B. Rong, and M. Kadoch, "Intelligent reflective surface based 6G communications for sustainable energy

infrastructure," *IEEE Wireless Communications*, vol. 28, no. 6, pp. 49–55, 2021.

[14] T. Li, W. Liu, Z. Zeng, and N. X. Neal, "DRLR: a deep reinforcement learning based recruitment scheme for massive data collections in 6G-based IoT networks," *IEEE Internet of Things Journal*, vol. 9, 2021.

[15] C. She, R. Dong, Z. Gu et al., "Deep learning for ultra-reliable and low-latency communications in 6G networks," *IEEE Network*, vol. 34, no. 5, pp. 219–225, 2020.

[16] S. B. Prathiba, G. Raja, S. Anbalagan, and K. Dev, "Federated learning empowered computation offloading and resource management in 6G-V2X," *IEEE Transactions on Network Science and Engineering*, vol. 9, 2021.

[17] S. Han, T. Xie, I. Chih-Lin et al., "Artificial-intelligence-enabled air interface for 6G: solutions, challenges, and standardization impacts," *IEEE Communications Magazine*, vol. 58, no. 10, pp. 73–79, 2020.

[18] H. Yang, A. Alphones, Z. Xiong, D. Niyato, J. Zhao, and K. Wu, "Artificial-intelligence-enabled intelligent 6G networks," *IEEE Network*, vol. 34, no. 6, pp. 272–280, 2020.

[19] H. Xu, J. Wu, J. Li, and X Lin, "Deep-reinforcement-learning-based cybertwin architecture for 6G IIoT: an integrated design of control, communication, and computing," *IEEE Internet of Things Journal*, vol. 8, no. 22, Article ID 16337, 2021.

[20] H. Ben-Ammar, Y. Hadjadj-Aoul, G. Rubino, and A.-C. Soraya, "A versatile Markov chain model for the performance analysis of CCN caching systems," in *Proceedings of the 2018 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, Abu Dhabi, United Arab Emirates, December 2018.

[21] O-Ran, "Cloud Architecture and deployment scenarios for O-RAN virtualized," 2021, https://www.o-ran.org/specifications.

[22] 3GPP TS, *User Equipment (UE) Radio Access Capabilities*, https://www.3gpp.org/ftp/Specs/archive/38_series/38.306, 2021.

[23] Willowgarage, "Open-source computer vision library (OpenCV)," 2022, http://https//www.willowgarage.%20com/pages/software/opencv.