Intelligent Edge-Aided Network Slicing for 5G and Beyond Networks

Jianhang Tang[1], Jiangtian Nie[2], Wei Yang Bryan Lim[2,3], Yang Zhang[4,5], Zehui Xiong[5],
Dusit Niyato[2], and Mohsen Guizani[6]
[1]School of Information Science and Engineering, Yanshan University, China
[2]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[3]Alibaba-NTU Joint Research Institute, Nanyang Technological University, Singapore
[4]College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
[5]Pillar of Information Systems Technology and Design, Singapore University of Technology and Design, Singapore
[6]Machine Learning Department, Mohamed Bin Zayed University of Artificial Intelligence, UAE

ICC 2022 - IEEE International Conference on Communications | 978-1-5386-8347-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICC45855.2022.9882270

***Abstract*—Network slicing at the edge is becoming a new enabler for 5G and beyond networks to support diverse and differential services, with efficient employment of virtualized resources provided by edge and cloud layers. However, as the arrival patterns of user requests are complicated and unpredictable in practical network scenarios, conventional system architectures and scheduling algorithms for core network slicing cannot handle the uncertain edge user task arrivals properly. Meanwhile, dynamic and intelligent allocation of multi-dimensional resources for edge-aided network slices is also a crucial issue. In this work, we develop a novel intelligent edge-aided network slicing scheme to reduce the system response time. Firstly, a deep belief network (DBN)-based task classification scheme is proposed. Due to the multi-layer structure of DBNs, the nonlinear features of user requests can be extracted efficiently with several individual restricted Boltzmann machines (RBMs). Compared with the classical models for network service classification, the DBNs can avoid over-fitting by the unsupervised pre-training process. Based on the classification results, a resource orchestration (S-RO) algorithm for edge-aided network slicing is investigated to reduce the system response time. Finally, the experiments to evaluate the proposed scheme are conducted with real-world datasets. The experimental results show that the S-RO algorithm is able to improve system throughput for providing network services.**

I. Introduction

The 5th generation (5G) communication technology works as an important role to provide communication and network functionalities, which can accommodate three typical and distinct services, which are *Enhanced Mobile Broadband* (eMBB), *Ultra-Reliable and Low Latency Communication* (uRLLC), and *Massive Machine Type of Communication* (mMTC). Network slicing is a key component and solution for 5G networks to provide dynamic, on-demand and customized resources for these typical services. With network slicing, multiple logically isolated networks of different service types are created over a common physical network infrastructure, which improves the resource utilization and efficiency for the networks to provide diverse 5G applications.

However, the proliferation of the new emerging application scenarios at the edge, e.g. unmanned aerial vehicles (UAVs) and Internet of Everything (IoE), is bringing new challenges for 5G network slicing schemes [1]. Network slicing at the edge requires intensive coordination of resources in both core networks as well as edge networks to provide fast service response, high data rates, elastic computing resources, and broad network coverage. It is urgent to push the current network slicing approaches from the core network to the edge in future 5G and beyond networks [2]. For 5G and beyond networks, it is expected that system resources are provided and coordinated both at the edge side as well as the core system side. Edge computing is able to offer low-latency services for user requests and applications, while cloud computing can supply supplementary resources for edge devices that have only limited resources. The edge-cloud integrated systems consist of extensive heterogeneous hardware facilities including several servers, routers, switches, access points, and base stations. It is a key problem to integrate these components which are controlled within the frameworks of different protocols and interfaces in current 5G network slicing architectures.

By exploiting network functions virtualization (NFV) and software-defined networking (SDN) technologies, edge computing resources are strategically integrated together with cloud resources as sub-slices in future 5G and beyond networks [3]. SDN provides centralized controllers to make edge and cloud networks programmable and improve the efficiency of resource management. As a result, underlying network facilities of SDN are controlled and managed by software regardless of their vendor variations. With centralized SDN controllers, resource blocks are created by virtualizing resources provided by both edge and cloud servers in edge-cloud integrated computing systems, such as storage and computational resources. Such edge-aided network slices are formulated by integrating multiple resource blocks into current network slices at the core network infrastructures, which can offer isolated and adequate resources for the emerging edge applications. With edge-cloud integrated resources, conventional typical services in 5G systems can evolve continuously to meet the dynamic requests of the ever-changing 5G and beyond networks, where the system is able to learn uncertain user behaviors and varying system environments to self-organize resources [4]. How to efficiently and dynamically allocate multi-dimensional resources for edge-aided network slices in such a hierarchical system is also a crucial problem.

To solve the aforementioned resource issues, we propose an intelligent edge-aided network slicing scheme for 5G and beyond networks, where the resources provided by edge and cloud servers are integrated into the current 5G network slices. The basic idea of the proposed slicing scheme is to make intelligent decisions for orchestrating the network slices and edge-cloud resources with the usage of artificial intelligence (AI) technologies to handle the limited knowledge and environment uncertainties. The main contributions are depicted as follows:

- A deep belief network (DBN)-based task classification scheme is proposed, where the inputs of DBNs are the task requirements and the outputs are the task types.
- Based on the classification results, a resource orchestration (S-RO) algorithm for edge-aided network slicing is developed to reduce the system response time, where each edge-aided network slice provides sufficient resources to the tasks with similar resource requirements.
- We conduct experiments with real-world datasets. The experimental results demonstrate that the proposed S-RO algorithm can increase the system throughput significantly.

This work is organized as follows. Section II reviews the related works. Section III proposes the intelligent edge-aided network slicing scheme. Section IV validates the proposed algorithm experimentally. Finally, the paper is concluded in Section V.

II. Related Work

Network slicing has been studied thoroughly in 5G and beyond networks [5]–[8]. *Togou et al.* [5] proposed a distributed network slicing architecture based on blockchain technology. The proposed framework used a blockchain-based bidding scheme to improve resource utilization. *Shu et al.* [6] developed a QoS framework for network slicing, where 5G networks were divided into three types of network slices, including radio access network slices, transport network slices, and core network slices. Different resource management schemes were proposed to create various network slices. *Zanzi et al.* [7] proposed a radio slicing orchestration scheme in a multi-tenancy environment. With the proposed scheme, the guarantees of both delay and throughput were provided without prior knowledge of system status statistics. *Messaoud et al.* [8] proposed a novel network slicing algorithm based on deep Q-learning and federated learning to improve the QoS reward. The deep Q-learning was used to adjust the transmission power, while federated learning was applied to learn the system status in a multi-agent environment.

AI enabled technologies were widely used for resource management in 5G and beyond networks [9]–[12]. *Jia et al.* [9] proposed an intelligent resource management scheme to improve the spectrum efficiency in 5G and beyond networks. A support vector machine is used to improve the detection accuracy of spectrum status and a convolutional neural network is applied to predict spectrum status. *Zhang et al.* [10] developed a deep Q-learning-based resource management algorithm for NFV. The proposed algorithm can guarantee communication quality and improve bandwidth utilization simultaneously. *Li et al.* [11] studied the security problems for edge intelligence in 5G and beyond networks. The authors used blockchain technologies to improve both the security and privacy of the system. *Alsenwi et al.* [12] studied a deep reinforcement learning-based resource management algorithm for network slicing. The proposed algorithm can improve resource utilization for eMBB and uRLLC applications.

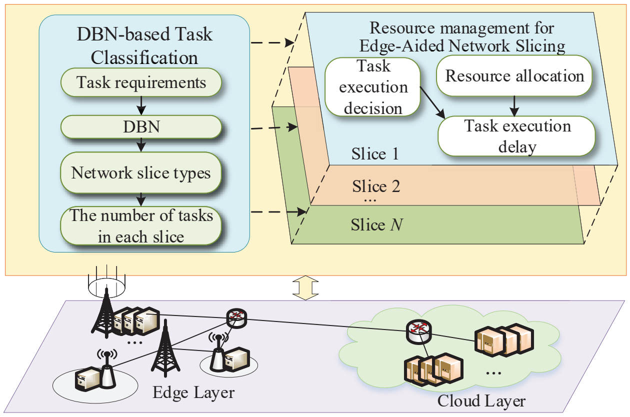


Fig. 1: System diagram of intelligent edge-aided network slicing.

In contrast to the aforementioned literature investigating the resource management for conventional network slicing, this work proposes a novel intelligent edge-aided network slicing scheme where multi-dimensional resources provided by edge and cloud computing are integrated into network slices.

III. Intelligent Edge-Aided Network Slicing

The system diagram of an intelligent edge-aided network slicing scheme is illustrated in Figure 1, where the underlying facilities are composed of edge and cloud layers. The intelligent edge-aided network slicing scheme operates as follows. Firstly, tasks submitted by end devices are classified based on their features, including latency, uplink and downlink transmission rates, and movement speeds. With the classification results, the network slice types and the number of tasks in each slice are determined. Then, the slice resources are orchestrated to process the submitted tasks.

A. DBN-based Task Classification

To identify the new service types and configure the system resources for these services, AI-enabled methods are employed to extract sufficient information from system operation data. In this work, deep belief networks (DBNs) are used to classify the tasks submitted by end devices. Compared with the classical approaches for network service classification, DBNs with the multi-layer structure can avoid over-fitting by the unsupervised pre-training process, which can improve the classification accuracy dramatically. A DBN consists of several Restricted Boltzmann Machines (RBMs) and a classifier [13]. With the stacked RBMs, DBNs abstract and capture deeper features from the user behaviors and system parameters. The network slices for emerging services can be created properly according to the service features. The tasks will be dispatched to appropriate network slices based on the classification results.

Let Re_t denote the requirements of task t, and let N_s represent the total number of network slices. Softmax function is employed as the classifier in the DBN. The output of the Softmax function is shown as follows:

$$f_\theta\left(Y_t\right)=\begin{bmatrix} P(Z_t=1|Y_t,\theta) \\ P(Z_t=2|Y_t,\theta) \\ \vdots \\ P(Z_t=N_s|Y_t,\theta) \end{bmatrix}=\frac{1}{\sum_{\eta=1}^{N_s} e^{\theta_\eta^T Re_t}}\begin{bmatrix} e^{\theta_1^T Re_t} \\ e^{\theta_2^T Re_t} \\ \vdots \\ e^{\theta_{N_s}^T Re_t} \end{bmatrix}, \tag{1}$$

where Y_t is the output of the last RBM with input Re_t, Z_t denotes the type of task t, and θ indicates the model parameters. For training DBNs, RBMs are pre-trained greedily first, and fine-tuning is then conducted to adjust all the parameters. The contrastive divergence algorithm is used for the former step, while the Limited memory Broyden-Fletcher-Goldbard-Shanno (L-BFGS) algorithm is applied for the latter step. The cross-entropy loss is used as a loss function for training DBNs [14]. The inputs of the DBNs are the task resource requirements, and the outputs are the task types.

B. Resource Management for Edge-Aided Network Slicing

After classifying the tasks submitted by end devices, the resources provided by edge and cloud servers will be allocated to tasks. Since network slice can logically isolate the resources, these slices will not affect each other when the system is in operation. $\mathcal{S}=\{1,2,\ldots,N_s\}$ denotes the resource slice set. In network slice s_n, the number of tasks, which is denoted by N_n, is determined by the classification results, where $n\in\mathcal{S}$. Let $t_{i,n}$ denote the i^{th} task in slice s_n. Task $t_{i,n}$ is defined by input data size $\lambda_{i,n}$ and deadline $DT_{i,n}$. A binary variable $d_{i,n}$ is introduced to indicate whether task $t_{i,n}$ is executed or not, as follows:

$$d_{i,n}=\begin{cases} 1, & \text{task } t_{i,n} \text{ is executed,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Let $R_{e,j}$ and $R_{c,j}$ be the amount of jth-type resources provided by the edge and cloud servers, respectively, where $j=1$ and $j=2$ indicate that network and computation resources are allowed. The network and computation resources are defined as network bandwidths and CPU-cycle frequencies respectively. Let $\pi_{e,j}^{i,n}$ and $\pi_{c,j}^{i,n}$ denote the amount of j^{th} type resources provided by edge and cloud servers. Then, the delay of processing task $t_{i,n}$ can be calculated as:

$$T_{i,n}=\frac{\lambda_{i,n}}{\pi_{e,1}^{i,n}+\pi_{c,1}^{i,n}}+\frac{\lambda_{i,n}\cdot X_{i,n}}{\pi_{e,2}^{i,n}+\pi_{c,2}^{i,n}}, \tag{3}$$

where $X_{i,n}$ denotes the number of CPU cycles required by task $t_{i,n}$ to process one-bit input. If task $t_{i,n}$ is dropped, we have $T_{i,n}=Lar$, where Lar is a large number to indicate the penalty of dropping tasks. Thus, we formulate the following optimization problem for slice resource management.

$$\textbf{P1}\text{: } \min \sum_{n=1}^{N_n}\sum_{i=1}^{N_n} T_{i,n} \tag{4a}$$

$$\text{s.t. } d_{i,n}\cdot T_{i,n}\le d_{i,n}\cdot DT_{i,n}, \forall n\in\mathcal{S}, i\in\{1,2,\ldots.N_n\}, \tag{4b}$$

$$\sum_{n=1}^{N_s}\sum_{i=1}^{N_n}\pi_{e,j}^{i,n}\le R_{e,j}, \sum_{n=1}^{N_s}\sum_{i=1}^{N_n}\pi_{c,j}^{i,n}\le R_{c,j}, \forall j\in\{1,2\}, \tag{4c}$$

$$\frac{\sum_{j=1}^{2}\left(\pi_{e,j}^{i,n}+\pi_{c,j}^{i,n}\right)}{\sum_{j=1}^{2}\left(R_{e,j}+R_{c,j}\right)}\le d_{i,n}\le\phi\cdot\frac{\sum_{j=1}^{2}\left(\pi_{e,j}^{i,n}+\pi_{c,j}^{i,n}\right)}{\sum_{j=1}^{2}\left(R_{e,j}+R_{c,j}\right)}, \tag{4d}$$

$$d_{i,n}\in\{0,1\}, \forall n\in\mathcal{S}, i\in\{1,2,\ldots.N_n\}, \tag{4e}$$

$$\pi_{e,j}^{i,n}\ge 0, \pi_{c,j}^{i,n}\ge 0, \forall j\in\{1,2\}, n\in\mathcal{S}, i\in\{1,2,\ldots.N_n\}, \tag{4f}$$

where ϕ denotes a large number. In **P1**, constraint (4b) guarantees the QoS of the tasks. Constraints (4c) and (4d) represent the resource limitations. Finally, constraints (4e) and (4f) limit the variable ranges.

The proposed optimization problem **P1** is a mixed-integer nonlinear programming (MINLP) problem, which is an NP-hard problem. Thus, the linear relaxation algorithm and the sequential quadratic programming (SQP) method [15] are used to obtain the optimal resource management solution.

Next, binary variable $d_{i,n}\in\{0,1\}$ is relaxed to $d'_{i,n}\in[0,1]$, where $n\in\mathcal{S}, i\in\{1,2,\ldots.N_n\}$. Due to the fractional objective function and constraints, **P1** can be transformed to a convex optimization problem with respect to $d'_{i,n}$, $\pi_{e,j}^{i,n}$, and $\pi_{c,j}^{i,n}$.

$$\textbf{P2}\text{: } \min \sum_{n=1}^{N_n}\sum_{i=1}^{N_n} T_{i,n} \tag{5a}$$

$$\text{s.t. (4b), (4c), (4d), (4f),} \tag{5b}$$

$$d_{i,n}\in[0,1], \forall n\in\mathcal{S}, i\in\{1,2,\ldots.N_n\}. \tag{5c}$$

We assume that $T(\pi,d')$ denotes the objective function of **P2** and $C(\pi,d')$ is the constraint set. Then, the Lagrangian function of **P2** can be defined as:

$$L(\pi,d')=T(\pi,d')+\beta\cdot Con(\pi,d'), \tag{6}$$

where β is the vector of Lagrange multipliers.

Next, a Taylor polynomial approximation is leveraged to reduce the computational complexity of obtaining the optimal solution of the aforementioned optimization problem with several fractional functions. Next, a two-term Taylor Series of Equation (6) can be obtained as:

$$\begin{aligned} & L\left(\pi,d',\lambda_{(In)}\right) \\ & =L\left(\pi_{(In)},d'_{(In)},\lambda_{(In)}\right) \\ & +\nabla L\left(\pi_{(In)},d'_{(In)},\lambda_{(In)}\right)^T\left[(\pi,d')-\left(\pi_{(In)},d'_{(In)}\right)\right] \\ & +\frac{1}{2}\left[(\pi,d')-\left(\pi_{(In)},d'_{(In)}\right)\right]^T\hat{H}_{L\left(\pi_{(In)},d'_{(In)},\lambda_{(In)}\right)} \\ & \cdot\left[(\pi,d')-\left(\pi_{(In)},d'_{(In)}\right)\right] \end{aligned} \tag{7}$$

where In denotes the iteration index, $\nabla L(\cdot)$ is the gradient of the Lagrangian function, and $\hat{H}(\cdot)$ indicates the Hessian matrix. Then, **P2** can be transformed as:

$$\textbf{P3}\text{: } \min OB \tag{8a}$$

$$\begin{aligned} \text{s.t., } & C\left(\pi_{(In)}, {d'}_{(In)}\right) \\ & + \nabla C\left(\pi_{(In)}, {d'}_{(In)}\right)\left[\left(\pi, d'\right) - \left(\pi_{(In)}, {d'}_{(In)}\right)\right] \leq 0, \end{aligned} \tag{8b}$$

where

$$\begin{aligned} OB = & \nabla L\big(\pi_{(In)}, {d'}_{(In)}, \lambda_{(In)}\big)^T \left[\left(\pi, d'\right) - \left(\pi_{(In)}, {d'}_{(In)}\right)\right] \\ & + \frac{1}{2}\left[\left(\pi, d'\right) - \left(\pi_{(In)}, {d'}_{(In)}\right)\right]^T \hat{H}_{L\left(\pi_{(In)}, {d'}_{(In)}, \lambda_{(In)}\right)} \\ & \cdot \left[\left(\pi, d'\right) - \left(\pi_{(In)}, {d'}_{(In)}\right)\right]. \end{aligned} \tag{9}$$

In the In^{th} iteration, the initial value of the next iteration can be obtained by the following rule:

$$\begin{aligned} \left(\pi_{(In+1)}, {d'}_{(In+1)}\right) = & \left(\pi_{(In)}, {d'}_{(In)}\right) \\ & + F_{(In)}\left[\left(\pi, d'\right) - \left(\pi_{(In)}, {d'}_{(In)}\right)\right], \end{aligned} \tag{10}$$

and

$$\beta_{(In+1)} = \beta_{(In)} + F_{(In)}\left(\beta^* - \beta_{(In)}\right), \tag{11}$$

where $F_{(In)}$ denotes the step length. **P3** is a convex quadratic programming problem, which can be solved by the active set method [16].

The resource orchestration algorithm for edge-aided network slicing (S-RO) is shown in Algorithm 1. First of all, a relaxed problem is obtained by transforming the original problem. Then, we use a two-term Taylor Series to deal with the objective function and constraints. A convex quadratic programming problem is achieved in each iteration, which is solved by the active set method. Finally, the binary variables are recovered according to the relaxed results.

The time complexity of Algorithm 1 is mainly determined by the computation of the Hessian matrix and the binary result recovery. The time complexity of calculating the Hessian matrix is $O(N_n^2)$. Moreover, the time spending on recovering the binary results is $O(N_n)$. Thus, the time spending of Algorithm 1 is $O(N_n^2)$.

Algorithm 1 S-RO Algorithm

```
Input: The amount of available resoureces $R_{e,j}$ and $R_{c,j}$, input data
    size $\lambda_{i,n}$, and the number of required CPU cycles to process one-
    bit input $X_{i,n}$
Output: Resource management decisions $\boldsymbol{d}^*$ and $\boldsymbol{\pi}^*$;
  Obtain the reformulated problem P2;
  Obtain the transformed problem P3 by a two-term Taylor Series;
  while $In \leq In_{\max}$ do
      Use the active set method to solve P3;
      Update constraints based on Equation (10) and Equation (11);
  end while
  Get the optimal solution of P2, $\boldsymbol{d}'^*$ and $\boldsymbol{\pi}'^*$;
  for $n \in \mathcal{S}$ do
      for $i \in \{1, 2, \ldots, N_n\}$ do
          $P[d_{i,n} = 1] = d'_{i,n}$
      end for
  end for
  return $R_{e,j}$ and $R_{c,j}$
```

IV. Performance Evaluation

In this section, we validate the proposed approach by experiments deployed in a real-world system environment with actual datasets.

A. Experimental Setup and Test Dataset

The experiment environment mainly includes edge and cloud layers. The edge layer is composed of 9 edge servers with different configurations. The edge servers are placed in 9 different rooms in a building. The cloud layer consists of 15 cloud servers. These cloud servers are deployed in another building distant away from edge devices. Due to the long distance between cloud and edge servers, the network bandwidth is 100 Mbps. Meanwhile, the network capacity between the edge servers or cloud servers is 200 Mbps. The edge and cloud layers are controlled by an edge SDN controller and a cloud SDN controller, respectively.

The edge servers are empowered with Intel(R) Core(TM) i5-2450M CPU and Intel(R) Xeon(R) E7-4870 CPU cores with 4GB RAM. The cloud servers are empowered with Intel(R) Core(TM) i5-9400F and Intel(R) Xeon(R) i5-4210M cores with 4GB RAM and 8GB RAM, respectively. The edge and cloud SDN controllers are installed on two servers with Intel(R) Core(TM) i7-9700F and 8GB RAM.

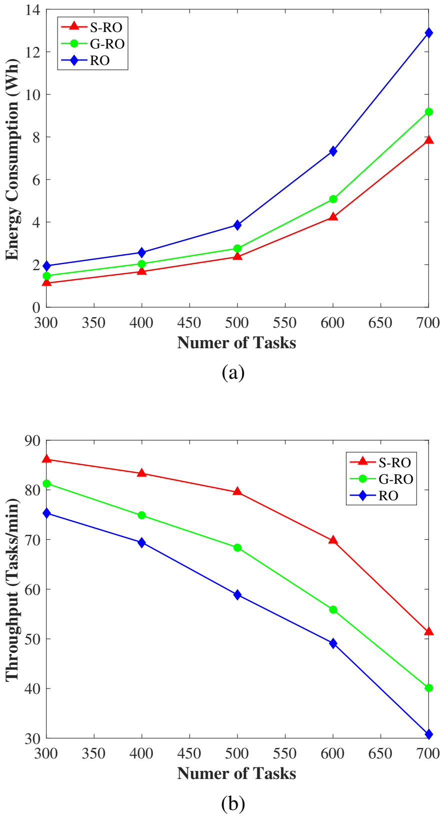
5G Trace Dataset [17] is used to train the DBNs. Scale-insensitive convolutional neural networks (SINets) and long short-term memory (LSTM) networks are considered as the benchmark tasks. As autonomous driving and virtual reality are two typical applications in 5G and beyond networks, the KITTI dataset and 360° video viewing dataset are adopted as the inputs of SINets and LSTM networks [18]. Based on the proposed S-RO algorithm, the benchmark tasks with different input sizes and deadlines are classified and dispatched to the corresponding network slices.

B. Benchmark Schemes

The proposed S-RO algorithm is compared with a resource orchestration algorithm without network slicing (RO) and a greedy resource allocation algorithm with network slicing (G-RO). We employ the following two benchmark schemes.

- RO scheme: Various tasks share the resources according to their deadlines without network slicing, where more resources are allocated to tasks with shorter deadlines.
- G-RO scheme: In each network slice, the tasks with shorter deadlines can obtain more resources to guarantee their QoS.

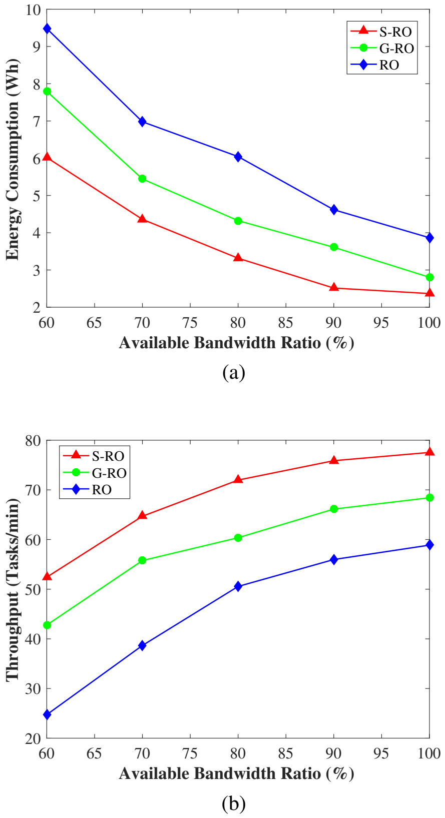
The system throughput and energy cost are selected as evaluation metrics. The system throughput is defined as the number of tasks completed by the system per minute. The energy cost is calculated from the total amount of energy to execute the tasks, which is measured by the energy consumption for both CPU and disk operations.



(a)

(b)

Fig. 2: Impacts of different numbers of tasks on (a) energy cost, and (b) throughput.



(a)

(b)

Fig. 3: Impacts of available bandwidth ratios on (a) energy cost, and (b) throughput.

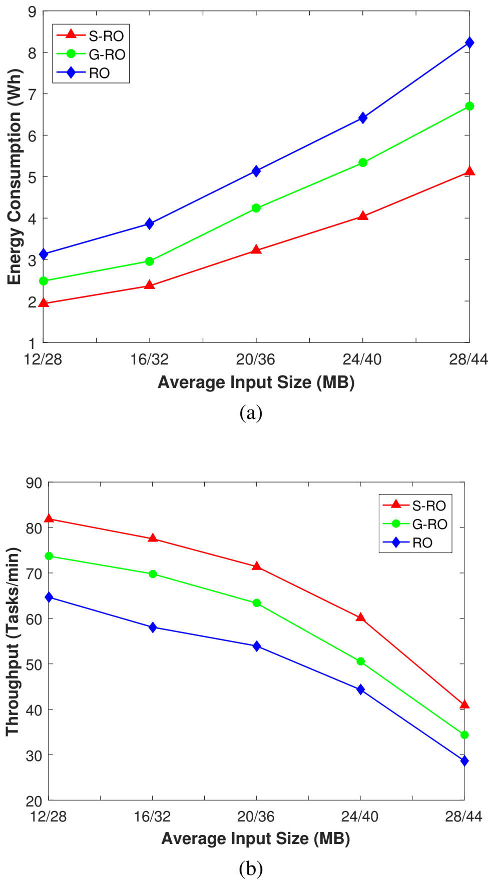
C. *Experimental Results*

In this work, the experiments are conducted in a real-world system environment with the aforementioned setups. First, we randomly select subsets of the data from datasets as the inputs of SINets and LSTM networks. The input sizes follow Poisson distributions with the means of 16MB and 32MB respectively. The total number of network slices created in the system is set to 4. The network slice lifecycle follows an exponential distribution with 300 seconds on average.

Figure 2 evaluates the system performance by setting different numbers of tasks. Figure 2a demonstrates that the energy costs act as increasing functions of the number of tasks. The result can be explained as, the resource requirements increase when more tasks are submitted, which needs more execution energy. The proposed algorithm can reduce the energy consumption by 17.54% and 39.44% on average by comparing it with G-RO and RO schemes. The reason is that the S-RO algorithm can use the minimum amount of resources to complete tasks to meet the deadline, which reduces energy consumption. Figure 2b illustrates the decreasing trend of system throughput by increasing the number of tasks. It is because the edge-cloud computing system which provides limited multi-dimensional resources cannot process more tasks. When the number of tasks is 500, the proposed S-RO algorithm can increase the throughput by 35.1% by comparing it with the RO algorithm. This is since resource slices can provide customized resources for tasks with diverse requirements. In this work, the inference tasks of SINets require more computation resources, while the inference tasks of LSTM networks need more network resources.

Figure 3 shows that the S-RO algorithm outperforms other baseline algorithms by setting different available bandwidths, where the available bandwidth ratio is defined as the ratio of the available bandwidths to the total bandwidths. As shown in Figure 3a, the energy consumption decreases dramatically when the available bandwidth increases. The reason is that more available network resources in each slice can result in higher utilization of storage and computation resources and lower task execution energy consumption. Figure 3b shows the impacts of average input sizes on the throughput. The S-RO algorithm can increase the system throughput by 13.34% and 31.72% with the maximum available bandwidth by comparing it with G-RO and RO algorithms. The reason is that the proposed algorithm can improve the utilization of multi-dimensional resources to improve system throughput.

Finally, the impacts of average input sizes on energy consumption and throughput are plotted in Figure 4. For all resource management schemes, the energy costs increase quickly with the increase in the average input size due to the fact that more multi-dimensional resources are required for each task, resulting in higher energy consumption for transmission, storage, and execution. Similarly, the system



(a)

(b)

Fig. 4: Impacts of average input sizes on (a) energy cost, and (b) throughput.

throughputs decrease quickly when the input sizes increase. With limited resources, the edge-cloud computing system can provide services for limited tasks, which increases the task response time. Moreover, the higher system performance of the proposed S-RO algorithm compared with the RO algorithm indicates that the network slicing can provide customized services for tasks with distinct requirements.

V. Conclusion

In this work, a novel intelligent edge-aided network slicing for 5G and beyond networks has been investigated, where the current network slices at the core network side are integrated with edge and cloud computing. To deal with the dynamic and uncertain user requesting behaviors, a DBN-based task classification scheme has been proposed, where the inputs of DBNs are the task requirements and the outputs are the task types. According to the classification results, the number of tasks in each slice has been determined. A resource orchestration (S-RO) algorithm for edge-aided network slicing has been developed to reduce the task execution latency. Finally, the experiments have been conducted with real-world datasets. The experimental results have shown that the proposed S-RO algorithm can enhance the system performance significantly in terms of increased throughput.

VI. Acknowledgment

This work is partly supported by National Natural Science Foundation of China (Grant No. 62071343), SUTD SRG-ISTD-2021-165, SUTD-ZJU IDEA Grant (SUTD-ZJU (VP) 202102), SUTD-ZJU IDEA Seed Grant (SUTD-ZJU (SD) 202101).

References

[1] M. Javad-Kalbasi and S. Valaee, "Re-configuration of UAV relays in 6G networks," in *2021 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2021, pp. 1–6.

[2] S. Zhang, J. Liu, H. Guo, M. Qi, and N. Kato, "Envisioning device-to-device communications in 6G," *IEEE Netw.*, vol. 34, no. 3, pp. 86–91, 2020.

[3] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6G wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 28–41, 2019.

[4] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5g: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, 2017.

[5] M. A. Togou, T. Bi, K. Dev, K. McDonnell, A. Milenovic, H. Tewari, and G.-M. Muntean, "DBNS: A distributed blockchain-enabled network slicing framework for 5G networks," *IEEE Commun. Mag.*, vol. 58, no. 11, pp. 90–96, 2020.

[6] Z. Shu and T. Taleb, "A novel QoS framework for network slicing in 5G and beyond networks based on SDN and NFV," *IEEE Netw.*, vol. 34, no. 3, pp. 256–263, 2020.

[7] L. Zanzi, V. Sciancalepore, A. Garcia-Saavedra, H. D. Schotten, and X. Costa-Pérez, "LACO: A latency-driven network slicing orchestration in beyond-5G networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 667–682, 2020.

[8] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, "Deep federated Q-learning-based network slicing for industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5572–5582, 2020.

[9] M. Jia, X. Zhang, J. Sun, X. Gu, and Q. Guo, "Intelligent resource management for satellite and terrestrial spectrum shared networking toward B5G," *IEEE Wireless Commun.*, vol. 27, no. 1, pp. 54–61, 2020.

[10] C. Zhang, M. Dong, and K. Ota, "Fine-grained management in 5G: DQL based intelligent resource allocation for network function virtualization in C-RAN," *IEEE Trans. on Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 428–435, 2020.

[11] Y. Li, Y. Yu, W. Susilo, Z. Hong, and M. Guizani, "Security and privacy for edge intelligence in 5G and beyond networks: Challenges and solutions," *IEEE Wireless Commun.*, vol. 28, no. 2, pp. 63–69, 2021.

[12] M. Alsenwi, N. H. Tran, M. Bennis, S. R. Pandey, A. K. Bairagi, and C. S. Hong, "Intelligent resource slicing for eMBB and URLLC coexistence in 5G and beyond: A deep reinforcement learning based approach," *IEEE Trans. Wireless Commun.*, vol. 20, no. 7, pp. 4585–4600, 2021.

[13] L. Tang, X. He, P. Zhao, G. Zhao, Y. Zhou, and Q. Chen, "Virtual network function migration based on dynamic resource requirements prediction," *IEEE Access*, vol. 7, pp. 112 348–112 362, 2019.

[14] X. Li, L. Yu, D. Chang, Z. Ma, and J. Cao, "Dual cross-entropy loss for small-sample fine-grained vehicle classification," *IEEE Trans. Veh. Technol.*, vol. 68, no. 5, pp. 4204–4212, 2019.

[15] C. D. Vilor and H. Jafarkhani, "Optimal 3D-UAV trajectory and resource allocation of DL UAV-GE links with directional antennas," in *2020 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2020, pp. 1–6.

[16] G. Cimini and A. Bemporad, "Exact complexity certification of active-set methods for quadratic programming," *IEEE Trans. Autom. Control*, vol. 62, no. 12, pp. 6094–6109, 2017.

[17] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5G dataset with channel and context metrics," in *2020 ACM Multimedia Systems Conference (MMSys)*. ACM, 2020, pp. 303–308.

[18] J. Tang, J. Nie, Z. Xiong, J. Zhao, Y. Zhang, and D. Niyato, "Slicing-based reliable resource orchestration for secure software defined edge-cloud computing systems," *IEEE Internet Things J.*, 2021.