

Modified Deep Reinforcement Learning Agent for Dynamic Resource Placement in IoT Network Slicing[☆]

로스세이하¹
Seyha Ros

담프로힘¹
Prohim Tam

김석훈^{1,2*}
Seokhoon Kim

ABSTRACT

Network slicing is a promising paradigm and significant evolution for adjusting the heterogeneous services based on different requirements by placing dynamic virtual network functions (VNF) forwarding graph (VNFFG) and orchestrating service function chaining (SFC) based on criticalities of Quality of Service (QoS) classes. In system architecture, software-defined networks (SDN), network functions virtualization (NFV), and edge computing are used to provide resourceful data view, configurable virtual resources, and control interfaces for developing the modified deep reinforcement learning agent (MDRL-A). In this paper, task requests, tolerable delays, and required resources are differentiated for input state observations to identify the non-critical/critical classes, since each user equipment can execute different QoS application services. We design intelligent slicing for handling the cross-domain resource with MDRL-A in solving network problems and eliminating resource usage. The agent interacts with controllers and orchestrators to manage the flow rule installation and physical resource allocation in NFV infrastructure (NFVI) with the proposed formulation of completion time and criticality criteria. Simulation is conducted in SDN/NFV environment and capturing the QoS performances between conventional and MDRL-A approaches.

☞ keyword : Deep reinforcement learning, network slicing, software-defined networking, network functions virtualization, edge computing

1. Introduction

A new era of the industrial revolution 4.0, the Internet of Things (IoT) is exponential growth, and the ultra-dense networks architecture has increased the connectivity of standard/proprietary devices and brought heterogeneous services of various applications such as the Internet of Vehicle (IoV), Internet of Surveillance, Smart home, and

industrial IoT [1,2]. To gets along with the 5G perspective, the utilization of smart management on applications and various service in complex network management and orchestration is required. In order to manage the network implementation, the exposed function has to be fully instantiated for offering a dynamic approach to allow network adaptations. Network slicing (NS) has operated into the physical-based logical for overcoming the heterogeneous services and complexity connection points from IoT devices and to be certain network traffic flow, and utilized to perform constructing the virtual network resource to a path (determination of network traffic flow in network metric). Additionally, The supportive technology challenges are obtained to all aspects of the 5G core networks (5G-CN) [2,3] which are wildly deployed forwarding the state-of-art that is used for diverse IoT devices. Therefore, the difference in Quality of Service (QoS) needs stringent classification and priority levels (e.g., ultra-reliable low-latency communication (URLLC), massive machine type communication (mMTC),

¹ Department of Software Convergence, Soonchunhyang University, Asan, 31538, Korea

² Department of Computer Software Engineering, Soonchunhyang University, Asan, 31538, Korea

* Corresponding author (seokhoon@sch.ac.kr)

[Received 8 July 2022, Reviewed 26 July 2022(R2 29 September 2022), Accepted 4 October 2022]

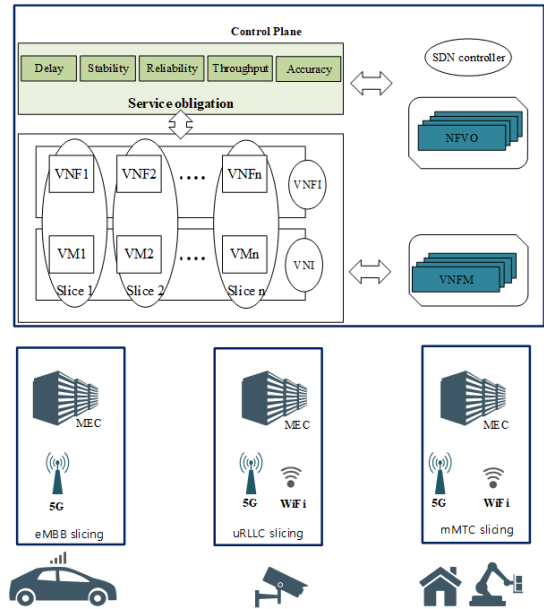
☆ This paper was supported by the Soonchunhyang University. In addition, this research was also supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (NRF-2020R111A3066543).

☆ 본 논문은 2022년도 인터넷정보학회 춘계학술발표대회 우수 논문 추천에 따라 확장 및 수정된 논문임

enhanced mobile broadband (eMBB)), which leads to the necessity of end-to-end intelligent slicing to fulfill the qualified experiences in mobile service operators. The enhancement of orchestrating service functions and achieving multiple QoS requirements are necessitated to consider by optimizing the network usage in flexibility, scalability, and efficiency of communication and computation resources [4]. The multi-service needs to be dynamically customized the functionality services according to gatherable fine-grained features of the data plane. Softwarization and virtualization have emerged from the Software-Defined Networks (SDN) and Network Function Virtualization (NFV) handle the processing of the 5G to investigate the resource management and to ensure high responsibility network resource and performance [5]. Those two paradigms depicted complementary characteristics of responding on extensible and flexible networks. NFV decouples network software from the functionality of physical resources such as (Storage, CPU, and Memory), NFV comes with evolution efficiency for breaking down with traditional networking to dynamically components and layers for supporting to work in any platform at any given time [6]. For instance, a virtualized firewall and load-balancing can be exposed to endure updated by network slicing providers (NSP). In order to implement them later on machines that allow running on top of commodity servers and forwarding elements. The SDN has technically separated the control logic in the centralized controller and break the decentralization of the network controller. Thus, SDN and NFV ensure networks of the programming capability approach and break down to assist service provisioning.

2. Related Works

Deep Reinforcement Learning (DRL) has been used in communications and networking for resource allocation, computation offloading decisions, and end-to-end network slicing perspectives [7]. DRL and the concept of the Markov Decision Process (MDP) activate the studies on action sets, exploration of independent states, and construction of reward approximators for intelligent decisions in the exploitation phase [8, 9]. DRL-based agent aims to observe the underlying features of adjustable communication resources



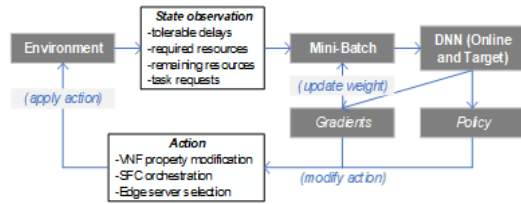
(Figure 1) Network slicing paradigm meets 5G

(e.g., allocated bandwidth between user's equipment and edge node) and computational resources (e.g., virtual machine capacities) for understanding the experienced performances and applying appropriate action at the right state [10]. Multi-agent DRL has been used to strengthen resource management in end-to-end network slicing by proposing a dynamic resource allocation with proximal policy optimization [11]. Traditional by being modified the multi-agent Deep Reinforcement learning (DRL) to enhance massive service which RL for VNF management and orchestration in an online fashion [12]. When a new VNF placement request arrives, a DRL agent decides to do the two conditions which DRL agent make a policy to determine the VNF on an already running server by assigning it a part of the server of remaining resource, separating the server and allocating its resource to the VNF, uploading the NFV to cloud, to minimizing incurred cost [13,14]. Joint slicing and DRL-based network slicing techniques called DRL-NS to perform the system throughput while satisfying all conditions in diverse QoS requirements, DRL-NS which adjusts the resource allocation decision violating various QoS requirements eliminated [15]. Since DL is effective in extracting policy from environments, it can be readily used

for decision-making problems such as resource management and scheduling [16]. DL has been also applied in many network slicing problems to come up with a well-informed slicing decision using available physical resources [17].

Due to the latency requirements of the control and inter-controller, the capacity was considered to meet the traffic load management of the switches. Meet the load balancing played a crucial role in enhancing QoS and getting an experience [17]. However, to meet an avoid network crashes, it is essential to assign and migrate switches between the slave and a master controller to adjust for load-balancing. The controller abstracts all information of the physical resource and traffic incoming to another node.

3. Proposed Approach



(Figure 2) MDRL-A architecture and interactivity

In this section, the proposed approach addresses to classification and recommendation of the network steering to formulated data gathering by implementation in SDN, in which controllability and interface are highly significant to abstract the status of device/resource information from the physical resource to indicator of network monitoring/services. SDN has been provisioning of control plane (CP) and data plane (DP). The DP consisted of connecting nodes from the edge nodes and multi-access computing (MEC) server by joint integration. In CP has been observant machine control entities to compute the required IoT services in network areas. In addition, CP obtains the entire information of the incoming network traffic from massive end-devices to MEC server. Figure 1 demonstrates the proposed network architecture which is contained the level (e.g. non-critical/critical) of the network. The SDN controller caches the status and monitors in updating of traffic incoming based on the conditions of the network status. SDN is inspected the serving capacity parameters in terms of

bandwidth and computing delay of the packets through the server. Moreover, SDN performs in-depth realization in terms of scalability and manageability for the traffic flow classification in different QoS applications has obligated for diverse MEC servers which heterogeneous resource parameters. SDN/NFV-enabled systems will be maintained and configured in forwarding flows and resources pooling for handling massive services accordingly. Figure 2 depicts the procedure flow and interactions between each entity as follows.

State: We defined the state observations which model the instance that comprised the allocation of resource and pending services. The state is designed observation of the informative slicing and global-view network status to policy network in slicing agent. The state provides information including tolerable delays, required resources, remaining resources, and task requests. Each state is non-sequential and independent in mini-batch for inputting into online and target deep neural networks (DNN). In this way, the minimizable loss or errors of DNN will be input into the gradient optimizer for weight modification; the reinforcement learning can be appended and achieved by taking the sequence into time T_s .

Action: A performance to allow the slicing agent to manage and orchestrate the virtual resource to fulfill in different domains of slicing. The Action will be inputted into the policy batch. The actions interact with the management and orchestration (MANO) to modify the VNF properties, orchestrate the forwarding graph, and render the SFC. After the action is applied, the performance of the environment will be updated accordingly.

Reward: The reward r_t formulates the rate in terms of termination ratios, completion time, and slicing prioritization efficiencies to determine whether the applied action a_t is degrading or upgrading the performance of state s_t .

The proposed optimally efficient policy orchestration in resource slicing scheme for modified deep reinforcement learning agent in SDN/NFV-enable system architecture is presented. To outcome with efficiently address classify IoT devices slicing. Algorithm 1 presented the construction flow of the proposed MDRL-A model. A composing to leverage MDRL-A method which is a key to the solution of the

problem the utilization functions of the time. To address this, we consider initializing the setting where an agent interacts with an environment in iteration. At each input decision time T , the agent observes a state S , takes an action a_t is randomly based on its policy, and acquired a reward $R(s_t + a_t)$. The action selection follows epsilon-greedy method to balance between the early and late phases, which are applied as exploration and exploitation, respectively. With defined action, the control SDN/NFV created the NFV orchestrator to obtain the VNF deployment in a forwarding graph descriptor. This VNF forwarding graph (VNFFG) will be appended to create optimal SFC according to the active primary function in each VNF, specifically with another critical class of the network functions services. To serve with virtual machines to enhance efficient execution in corresponding. In addition, the separated network services on each slicing class, due to heterogeneity sync with classified and prioritized based on VNF in a management system, so SDN/NFV interfaces orchestrate the action-based VNF properties and SFC connections. The experienced performance is stored for future batch comparison and enables the automation of direct edge server recommendations based on slicing criticalities.

4. Result and Discussions

In this section, the experimental environment of the proposed algorithm is given in two phases: (1) OpenAI Gym-based DRL agent and (2) RYU-based flow configuration. In the early phase, we model the environment for IoT network slicing, by initializing the primary state features using init function. The state observations are varied from each iteration and associated with the action value adjustment. The agent applies the explored or exploited action using the action step function to calculate the altering value towards the next state. The immediate reward is formulated by evaluating the state-action pair efficiency at that particular network condition. In the experiment, negative rewards output for the non-optimal taken action and append for averaging at that time slot. If an optimal action is taken, the optimal reward is given as 0. The main hyperparameters for MDRL-A are 0.95, 300, and 0.01 for discount factor,

Algorithm 1 Construction flow of the proposed MDRL-A model

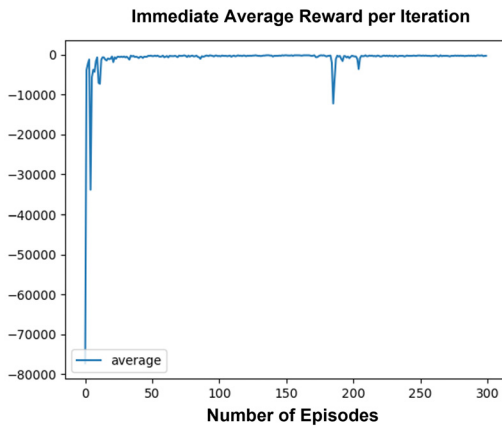
Require: s_t at particular time t

Ensure: Policy orchestration with optimal rewards

- 1: Initialize the structural function approximators (DNNs)
 - 2: **for** each state within step t **do**
 - 3: **if** exploration **do**
 - 4: $a_t \leftarrow \text{random}(A)$
 - 5: **else**
 - 6: Output a_t with function approximator (DNN-Online)
 - 7: **end if**
 - 8: Apply a_t by control SDN/NFV interfaces to orchestrate the action-based VNF properties and SFC connects
 - 9: Formulate $\text{reward } r_t$
 - 10: Append experience batch e_t and adjust mini-batch
 - 11: **Function Approximator (DNN-Online):**
 - 12: Input (s_t, a_t, r_t)
 - 13: Calculate loss values
 - 14: **Function Approximator (DNN-Target)**
 - 15: $q\text{-value: } \text{argmax } Q(a)$
 - 16: Long-term q -value calculation
 - 17: Sync model parameters between DNN-Online and DNN-Target
 - 18: **end for**
-

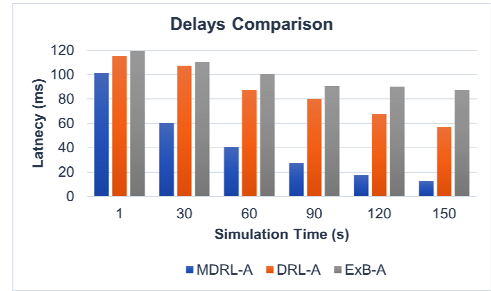
number of episodes, and learning rate, respectively. (Figure 3) shows the immediate average reward for 300 episodes to indicate the efficiency and converging speed of the proposed agent. The output labels are defined into 3 primary classes: (1) the exploration phase, (2) the fluctuation of randomness, congestion, and state reset, and (3) the optimality. In the early episode, negative rewards are high due to the observing procedures of applying random action for forecasting the relativity between each action to each possible state. The fluctuation in some episodes indicates the struggling process to achieve optimal performance in case the network conditions are altered. However, the optimal episodes are outperformed at manifold episodes, which allows the usage of the agent to be efficient. The proposed agent achieves over 250 near-optimal/optimal values out of 300, which leads to a concluded exploitation policy for

RYU-based configuration. RYU 4.32 is used as a primary remote controller for configuring and modifying the rules in the Mininet environment [14, 18]. Post functions are written using Python programming language following the output of exploited MDRL-A policy. The experimental simulation is conducted for 150 seconds. The key performance metrics are delay and packet delivery ratios between 3 conditions: (1) the proposed approach, MDRL-A, (2) a non-modified DRL agent, DRL-A, and (3) an experience-based allocation agent, denoted as ExB-A, following the historical relation features.

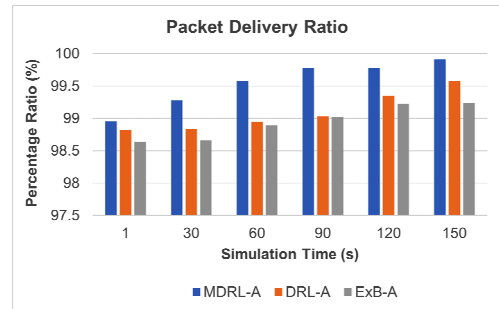


(Figure 3) Immediate average reward of proposed MDRL-A in 300 episodes

The comparison of delay between MDRL-A, DRL-A, and ExB-A is given (see Figure 3). The performance of the proposed agent reached an ending delay of 12.2176 which is 44.9998 ms and 74.993 ms lower than DRL-A and ExB-A, respectively. In the average of 150 seconds of simulation time, the proposed MDRL-A reached 43.0558 ms, while DRL-A and ExB-A reached 85.7244 ms and 99.5572 ms, respectively. These performances indicate an adequate resource adjustment of the exploited policy for serving each service execution in IoT network slicing. After the exploration time of the first 60 seconds, the agent is capable of outputting a reliable metric for mission-critical IoT services. (Figure 5) illustrates the packet delivery ratios in 150 seconds of experimental time. The successful rate of transmission indicates the reliability of the system architecture. In this study, the proposed MDRL-A has an



(Figure 4) Delay comparison



(Figure 5) Comparison of packet delivery ratio

ending reliable rate of 99.9112%, which is 0.337% and 0.677% higher than DRL-A and ExB-A, respectively. The proposed agent ensures each SFC execution in IoT network slicing scenarios.

5. Conclusion

This paper studied the VNF properties placement problem satisfying the dynamic network by enabling SDN/NFV networks. We proposed MDRL-A aims at a maximal cost and requirement based on user requests while the use case requirement for maximizing the resource efficiencies in multi-QoS class network slicing in terms of 5G perspectives. In our scenario, we consider control and orchestration of the network to perform that acquired user's demand and its map. The agent interacts with the controller/orchestrator for modifying the VNF properties, adjusting the optimal by selecting the slice that responds accordingly. Simulation results depict that proposed to be the most convenient network slicing during applying policy scenarios and saving

the experienced performance for edge recommendation. In future works, we aim to consider the multi-agent and reward formulation of end-to-end latency in each slicing to maintain the required QoS.

Reference

- [1] H. Hamzah, D. Le, M. Kim, and H. Choo, "Mobility-Aware Service Migration (MASM) Algorithms for Multi-Access Edge Computing," *Journal of Internet Computing and Services*, Vol. 21, No. 4, pp. 1-8, 2020. <http://dx.doi.org/10.7472/jksii.2020.21.4.1>
- [2] D. Sattar and A. Matrawy, "Optimal Slice Allocation in 5G Core Networks," in *IEEE Networking Letters*, vol. 1, no. 2, pp. 48-51, 2019. <http://dx.doi.org/10.1109/LNET.2019.2908351>
- [3] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen and X. Li, "End-to-End Quality of Service in 5G Networks: Examining the Effectiveness of a Network Slicing Framework," in *IEEE Vehicular Technology Magazine*, vol. 13, no. 2, pp. 65-74, 2018. <http://dx.doi.org/10.1109/MVT.2018.2809473>
- [4] V. S. Mai, R. J. La, T. Zhang and A. Battou, "End-to-End Quality-of-Service Assurance with Autonomous Systems: 5G/6G Case Study," 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), pp. 644-651, 2022. <http://dx.doi.org/10.1109/CCNC49033.2022.9700514>
- [5] A. Moubayed and A. Shami, "Softwarization, Virtualization, and Machine Learning for Intelligent and Effective Vehicle-to-Everything Communications," in *IEEE Intelligent Transportation Systems Magazine*, vol. 14, no. 2, pp. 156-173, 2022. <http://dx.doi.org/10.1109/MITS.2020.3014124>
- [6] A. Filali, Z. Mlika, S. Cherkaoui and A. Kobbane, "Dynamic SDN-Based Radio Access Network Slicing With Deep Reinforcement Learning for URLLC and eMBB Services," in *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2174-2187, 2022. <http://dx.doi.org/10.1109/TNSE.2022.3157274>
- [7] K. Suh, S. Kim, Y. Ahn, S. Kim, H. Ju and B. Shim, "Deep Reinforcement Learning-Based Network Slicing for Beyond 5G," in *IEEE Access*, vol. 10, pp. 7384-7395, 2022. <http://dx.doi.org/10.1109/ACCESS.2022.3141789>
- [8] N. C. Luong et al., "Applications of Deep Reinforcement Learning in Communications and Networking: A Survey," in *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133-3174, 2019. <http://dx.doi.org/10.1109/COMST.2019.2916583>
- [9] T. Li, X. Zhu and X. Liu, "An End-to-End Network Slicing Algorithm Based on Deep Q-Learning for 5G Network," in *IEEE Access*, vol. 8, pp. 122229-122240, 2020. <http://dx.doi.org/10.1109/ACCESS.2020.3006502>
- [10] H. Ko, J. Lee and S. Pack, "Priority-Based Dynamic Resource Allocation Scheme in Network Slicing," 2021 International Conference on Information Networking (ICOIN), pp. 62-64, 2021. <http://dx.doi.org/10.1109/ICOIN50884.2021.9333944>
- [11] S. Nath and J. Wu, "Deep reinforcement learning for dynamic computation offloading and resource allocation in cache-assisted mobile edge computing systems," in *Intelligent and Converged Networks*, vol. 1, no. 2, pp. 181-198, Sept. 2020. <http://dx.doi.org/10.23919/ICN.2020.0014>
- [12] M. Bunyakitanon, X. Vasilakos, R. Nejabati and D. Simeonidou, "End-to-End Performance-Based Autonomous VNF Placement With Adopted Reinforcement Learning," in *IEEE Transactions on Cognitive Communications and Networking*, vol. 6, no. 2, pp. 534-547, 2020. <http://dx.doi.org/10.1109/TCCN.2020.2988486>
- [13] P. Tam, S. Math, and S. Kim, "Priority-Aware Resource Management for Adaptive Service Function Chaining in Real-Time Intelligent IoT Services," *Electronics*, vol. 11, no. 19, 2022. <http://dx.doi.org/10.3390/electronics11192976>
- [14] P. Tam, S. Math and S. Kim, "Deep Neural Network-Based Critical Packet Inspection for Improving Traffic Steering in Software-Defined IoT," *Journal of Internet Computing and Services*, vol. 22, no. 6, pp. 1-8, 2021. <http://dx.doi.org/10.7472/jksii.2021.22.6.1>

- [15] Y. Azimi, S. Yousefi, H. Kalbkhani and T. Kunz, "Energy-Efficient Deep Reinforcement Learning Assisted Resource Allocation for 5G-RAN Slicing," in *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 856-871, 2022.
<http://dx.doi.org/10.1109/TVT.2021.3128513>
- [16] A. Nassar and Y. Yilmaz, "Deep Reinforcement Learning for Adaptive Network Slicing in 5G for Intelligent Vehicular Systems and Smart Cities," in *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 222-235, 2022.
<http://dx.doi.org/10.1109/JIOT.2021.3091674>
- [17] A. Nassar and Y. Yilmaz, "Deep Reinforcement Learning for Adaptive Network Slicing in 5G for Intelligent Vehicular Systems and Smart Cities," in *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 222-235, 2022.
<http://dx.doi.org/10.1109/JIOT.2021.3091674>
- [18] P. Tam, S. Math and S. Kim, "Efficient Resource Slicing Scheme for Optimizing Federated Learning Communications in Software-Defined IoT Networks," *Journal of Internet Computing and Services*, vol. 22, no. 5, pp. 27-33, 2021.
<http://dx.doi.org/10.7472/jksii.2021.22.5.27>

● Authors ●



로스세이하(Seyha Ros)

2017: Royal University of Phnom Penh (Cambodia), Telecommunication and Electronic Engineering (Bachelor Degree)

2021~Present: Soonchunyang University (Korea), Software Convergence (Combined Master and Ph.D.)

Research interest: 5G, B5G, QoS, SDN, NFV, Edge intelligence.

E-mail : rosseyha003@gmail.com



담프로힘(Prohim Tam)

2015: Paragon International University (Cambodia), Management of Information System (Bachelor Degree)

2019~Present: Soonchunyang University (Korea), Software Convergence (Combined Master and Ph.D.)

Research Interests: 5G Service-Based Orchestration, QoS, SDN/NFV, Edge Intelligence, and IoT.

E-mail : prohingtam@gmail.com



김석훈(Seokhoon Kim)

2000: Kyunghee University (Korea), Computer Engineering (Bachelor Degree)

2004: Kyunghee University (Korea), Computer Engineering (Master and Ph.D. Degree)

2016~Present: Professor at Department of Computer Software Engineering, Soonchunhyang University, Korea

Research Interests: Cloud and Mobile Edge Computing, Internet of Things, Heterogenous Internet of Things, Software Defined Networking and Network Function Virtualization, Mobile System and Communications, and Machine Learning based on Bigdata.

E-mail : seokhoon@sch.ac.kr