https://doi.org/10.7840/kics.2025.50.3.406

Optimal Power Allocation and Sub-Optimal Channel Assignment for Downlink NOMA Systems Using Deep Reinforcement Learning

WooSeok Kim◆, Jeonghoon Lee*, Sangho Kim**,
Taesun An**, WonMin Lee**, Dowon Kim**, Kyungseop Shin°

요 약

최근 몇 년 동안, 딥러닝의 발전에 따라 비직교 다중 접속(Non-Orthogonal Multiple Access, NOMA) 시스템에 딥러닝을 통합하고자 하는 노력으로써 NOMA 시스템은 다중 접속 프레임워크의 유망한 후보로 떠오르고 있다. 이러한 활발한 연구의 주된 동기로는, 사물인터넷(Internet of Things, IoT)의 확장으로 인한 네트워크 자원의 한정성에 대응하고자 네트워크 자원의 활용을 최적화할 필요성이 증가하고 있기 때문이다. NOMA는 사용자들이 네트워크에 동시다발적으로 접속하게 해주는 전력 다중화를 통해 이러한 요구를 해결한다. 그럼에도 불구하고, NOMA 시스템은 몇 가지 한계점이 존재한다. 전력 할당 최적화를 하는 Joint Resource Allocation(JRA) 방법, JRA 방법과 심층 강화학습(deep reinforcement learning, DRL)을 통합하는 JRA-DRL 방법 등을 포함한 다양한 방법들이 이러한 한계점들을 보완하고자 제안되었다. 그러나 채널 할당 문제는 여전히 불명확하며 추가적인 연구가 필요 하다. 본 논문에서는, NOMA 시스템에서 네트워크 자원을 할당하는 심층 강화학습 프레임워크를 제안하며, 이는 학습을 일반화시키기 위해 on-policy 알고리즘에 리플레이 메모리(replay memory)를 통합하는 방식이다. 또한 본 논문에서는, 학습률, 배치 사이즈, 모델의 종류, 그리고 상태 정보(state)의 특징 개수에 변화를 주었을 때의 효과를 평가하기 위해 다양한 실험 결과들을 제공한다.

Key Words : Non-orthogonal multiple access (NOMA), deep reinforcement learning (DRL), wireless network, resource allocation

ABSTRACT

In recent years, Non-Orthogonal Multiple Access (NOMA) system has emerged as a promising candidate for multiple access frameworks due to the evolution of deep machine learning, trying to incorporate deep machine learning into the NOMA system. The main motivation for such active studies is the growing need to optimize the utilization of network resources as the expansion of the internet of things (IoT) caused a scarcity of network resources. The NOMA addresses this need by power multiplexing, allowing multiple users to access the network simultaneously. Nevertheless, the NOMA system has few limitations. Several works have proposed to mitigate this, including the optimization of power allocation known as joint resource allocation(JRA) method, and integration of the JRA method and deep reinforcement learning (JRA-DRL). Despite this, the channel assignment problem remains unclear and requires further investigation. In this paper, we propose a deep reinforcement learning framework incorporating replay memory with an on-policy algorithm, allocating network resources in a NOMA system to generalize the learning. Also, we provide extensive simulations to evaluate the effects of varying the learning rate, batch size, type of model, and the number of features in the state.

◆ First Author : Sangmyung University, Department of Computer Science, 3suksw@gmail.com, 학생회원
° Corresponding Author : Sangmyung University, Department of Computer Science, ksshin@smu.ac.kr, 정회원
* Sangmyung University, Department of Game Design and Development, 2jh0926@naver.com
** Sangmyung University, Department of Computer Science, ghtkdrla321@naver.com; asdgqe1@gmail.com; wonmin98@naver.com; wlsgur479@gmail.com, 학생회원
논문번호 : 202408-192-A-RN, Received August 29, 2024; Revised October 27, 2024; Accepted November 11, 2024

Ⅰ. Introduction

Over the past few years, rapid development in Internet of Things (IoT) has resulted in a drastical increase in network demands, leading to the new challenge of guaranteeing massive connectivity and quality of service (QoS). To fulfill such demands and challenges, recents studies are focusing on integrating artificial intelligence (AI) to networking system[1,2]. For instance, Yu *et al.*[3] used an AI to learn the optimal wireless resource allocation method for MAC protocols and Ye *et al.*[4] focused on packet delay and power efficiency.

To be specific, there have been multiple attempts to use AI to fully use the advantages of Non-Orthogonal Multiple Access (NOMA). Compared to a conventional technique called Orthogonal Multiple Access (OMA) which is to allocate network resources orthogonally, NOMA mainly utilizes the ability of successive interference cancellation (SIC) which enables differentiation of users through different power assignments even in the same resource block. SIC is a technique which decodes received signals sequentially, treating unrelated signals as interference and then remove the signals[5]. NOMA is a spectrum-efficient wireless networking technique, allowing multiple users to share common resources such that time and frequency and it is anticipated that NOMA will play a pivotal role in

5G era and future wireless networking system as the technological advance in AI.

Although, NOMA has some limitations in IoT environments. Representatively, solution of assigning channels and allocating powers is known to be NP-hard[6] and the complexity of the system increases as the nature of dynamic environment and the SIC. Reinforcement learning is suggested as a potential solution to resolve such issues. Not only reinforcement learning has the ability to process complex system, but also it can learn the optimal policy off of dynamic environmental systems, allocating channels and assigning powers. Furthermore, researchers propose various algorithms to enhance the performance for optimal resource allocation problems. Solving a power allocation and an assigning channel problems is the key to the optimal resource allocation in NOMA system. He *et al.*[5] suggested a power assignment method improving channel gain, and also proposed a joint resource allocation (JRA) and channel allocation method to maximize the NOMA system using DRL framework.

Ahsan *et al.*[7] utilized the potential of the NOMA power domain. The following paper proposed an efficient and optimized algorithm to enhance IoT connectivity, utilizing DRL and State-Action-Reward-State-Action (SARSA). SARSA is an on-policy algorithm in which the agent selects an action based on the current policy then evaluates and update the policy based on the action taken. The paper shows that the IoT networking utilizing NOMA outperforms the IoT networking using OMA system in terms of the number of processes agent can take in.

Under downlink NOMA system which is to estimate imperfect channels, Wang *et al.*[8] proposed an approach to allocate power. Considering the two information, channel estimation and Mean Squared Error (MSE), the objective is to set the upper bound of System Outage Probability (SOP) using two users' throughput requirements. Afterwards, in order to make the SOP minimized under overall power constraints, outage power allocation solution for two users is driven. The solution can be driven within few calculations for power allocation coefficient, by the MSE of the channel estimation, and is way less complex than the previous outage or repetitive solutions. The simulation results are showing that the proposed method achieves great performance in various transmission rate requirements resulting a SOP.

Zhang *et al.*[9] proposed a power allocation algorithm for one BS and two user clusters assigned to mmWave-NOMA system. To be specific, the proposed algorithm meets the individual service quality constraint requirements and maximizes achievable sum rate (ASR) and energy efficiency (EE), in consequence, it formulated the optimization problems. In order to guarantee stability of SIC, the algorithm added power order constraint which is commonly dismissed from the previous related works. The algorithm divides the formulated problem into sub-problems as clustering problem to make the problem

easier to solve, deriving the solutions for ASR Maximization-based Power Allocation (ASRMax-PA) algorithm and EE Maximizationbased PA (EEMax-PA). The proposed ASRMax-PA (or EEMax-PA) algorithm outperforms than the latest methods in the aspect of ASR (or EE) and performs great in EE (or ASR) as well. Not only that, the two proposed methods can assure the stability of SIC which is a critical factor for performance of NOMA system.

Although aforementioned researches do not explicitly point out the limitations that they have, the simulations took place were conducted in a specific, restricted environment rather than a dynamic environment. To address this issue, in this paper, we propose an effective framework where a Deep Reinforcement Learning (DRL) agent efficiently allocates limited networking resources in a downlink NOMA system. The main difference between the proposed framework and previous works is that it uses an experience replay memory to generalize learning, rather than solving the onpolicy problem with traditional policy gradient methods. Policy gradient methods evaluate and update the policy every iteration and experience replay is to save series of experiences that agent had and sample the experiences with batch resulting a generalized learning. See Section III for more detailed explanation why experience replay memory has been applied to this framework. The goal of the agent is to learn a policy for a downlink NOMA system under various profiles to enhance understanding of the generalized NOMA system attempting for a maximum sum throughput (*i.e.*, sum rate).

The verification of our proposed framework is conducted through multiple simulations with varying controls to assess its performance. In detail, changes in the types of neural networks such that fully connected neural network (FCNN), convolutional neural network (CNN), and attention-based neural network (ANN), batch sizes, learning rates, and number of NOMA users are made. Also, the comparisons between the frameworks such that Joint Resource Allocation (JRA), JRA-DRL, the proposed framework, and Exhaustive Search (ES) are conducted as well. The paper will thoroughly analyze the experimental results by inspecting loss, loss convergence speed and resulted sum rate. Especially, since the environment of networking is dynamic and ever-changing, the convergence speed is crucial in networking system. The problem addressed in this paper is maximizing data throughput through efficient resource allocation. Therefore, overall environmental definition will be settled first.

Contributions. Since the JRA-DRL method learns the policy of the NOMA system directly from current experience, it may lack generality and be unable to handle various scenarios. To address this, we have incorporated a policy gradient method and replay memory to enhance the generality. Therefore, we have incorporated a policy gradient method and replay memory to enhance generality. Additionally, deep learning techniques require extensive fine-tuning, such as changing the model architecture, tuning hyperparameters, and, in DRL, the design of the state directly influences the training results. To demonstrate the effectiveness of our method, we provide extensive simulation results. We make the following contributions:

- Incorporation of policy gradient method and replay memory: This approach avoids biased training since the experiences used for training are sampled from a replay memory, unlike the conventional policy gradient method, which trains on current experience and may result in overfitting. The use of replay memory results in more balanced and generalized learning of the NOMA system, reducing the variability in training outcomes while improving the convergent stability and reliability.
- Extensive simulation results: We observed that small changes in hyperparameters, model architectures, or the design of the DRL state lead to significant differences in simulation results. Therefore, we carefully evaluated a range of settings to identify the most robust configuration and provided numerous simulation results. This thorough analysis examines the impact of the DRL state size, which gradually increases the number of key features in the given NOMA system, convergence speed when modifying model archi-

tectures, and the fine-tuning of hyperparameters.

Ⅱ. System Model

In this paper, we assume a downlink NOMA system where BS sends data to multiple users in wireless channels. Given this environment, a comprehensive definition of overall environment is necessary.

In a wireless channel, note that power and channel information is required in order to describe the relationships between BS and users. The BS's job is to distribute limited joint resources (*i.e.*, channel and power) and multiplex signals, then transmit the multiplexed signals to users. After users receive multiple signals which contain independent signals, users use decoder to perform SIC and specify the signal for their own. Additionally, there is an upper bound for power allocation, as well as for channel bandwidth. Given these constraints, by using two methods, JRA and DRL, an optimal solution allocating joint resources can be found.

Fig. 1 briefly illustrates the transmission and the reception between BS and users in a downlink NOMA system, where we assume there are N users and K channels. The total bandwidth is B_{tot} and since all channels have the same bandwidth, the bandwidth for each channel is represented as $B_c = B_{tot}/K$. Also, the number of users assigned to channel k is N_k . Consider a user i's signal as b_i, then the BS will transmit multiple signals as follows,

$$x_k = \sum_{i=1}^{N_k} \sqrt{p_i^k} b_i \tag{1}$$

where p_i^k is a power of i-th user assigned to k-th channel, x_k is a multiplexed signal from k-th channel. As receiver receives signal from transmitter, the signal is corrupted by environmental noises. The receiver will eventually receive a signal y_k and can be written as,

$$y_n^k = \sqrt{p_n^k} h_n^k b_n + \sum_{i=1, i\neq n}^{N_k} \sqrt{p_n^k} h_n^k b_i + z_n^k \tag{2}$$

where h_n^k is a k-th channel response between BS and n-th user, z_n^k is user n's Additive White Gaussian Noise (AWGN) with zero mean and variance of σ_{zk}^2. When the signals from multiple users are multiplexed

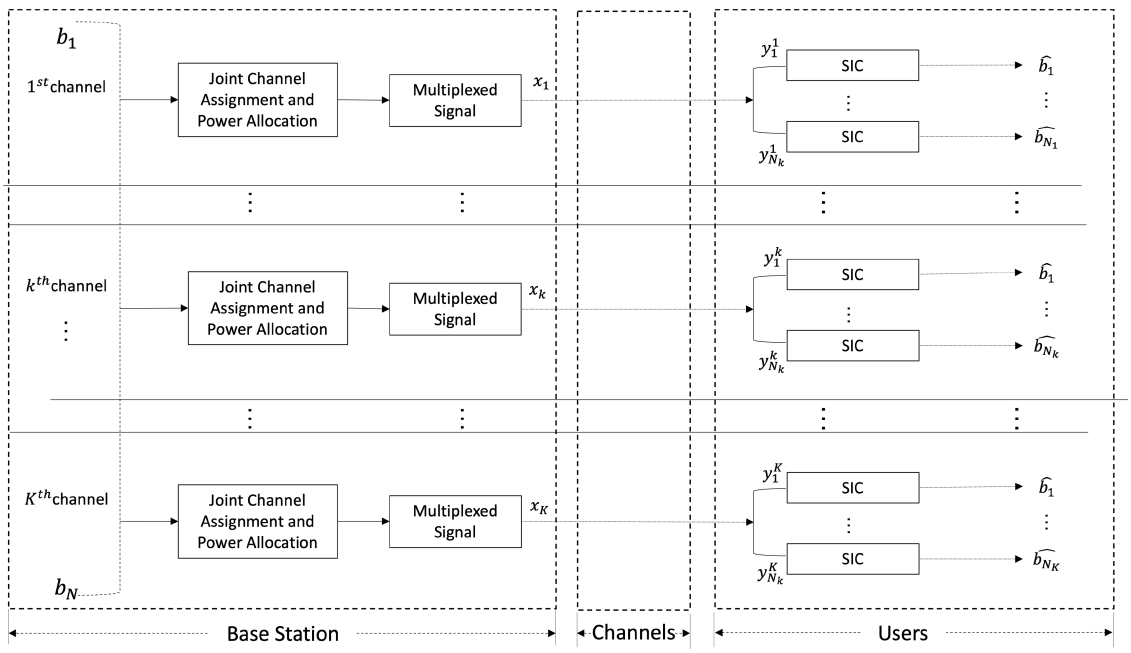


Fig. 1. Block diagram illustrating the transmission of BS and reception of users of the downlink NOMA system.

as shown in (1) and the receiver's noise-added final received signal is given by (2), SIC is applied to decode each users' signal, differentiating multiple signals.

In order for SIC to successfully be performed, channel-to-noise-ratio (CNR) should be considered which can be represented as $\Gamma_n^k = |h_n^k|^2/\sigma_n^k$. Since the greater power is assigned to users with lower CNR, according to the NOMA protocol, powers and CNRs on k-th channel can be ordered as such:

$$\Gamma_1^k > \Gamma_2^k > \ldots > \Gamma_n^k > \ldots > \Gamma_{N_k}^k,$$
$$p_1^k < p_2^k < \ldots < p_n^k < \ldots < p_{N_k}^k.$$

This behavior allows other weak signals to be treated as noises while the signal with greater power to be decoded primarily. Also the corresponding data rate is as follows,

$$R_n^k(\Gamma_n^k, p_1^k, \ldots, p_n^k) = B_c \log_2\left(1 + \frac{p_n^k \Gamma_n^k}{1 + \sum_{i=1}^{n-1} p_i^k \Gamma_n^k}\right). \tag{3}$$

As stated in Ding *et al.*[10,11], the number of users allocated in channel k is fixed to two (*i.e.*, $N_k = 2$), because the increase in N_k directly affects the hardward implementation complexity and processing time. Reflecting the mentioned change, the data rates for two users allocated at channel k are represented as,

$$R_1^k(\Gamma_1^k, p_1^k, p_2^k) = B_c \log_2(1 + p_1^k \Gamma_1^k),$$
$$R_2^k(\Gamma_2^k, p_1^k, p_2^k) = B_c \log_2(1 + \frac{p_2^k \Gamma_2^k}{1 + p_1^k \Gamma_2^k}). \tag{4}$$

Note that the problem we are trying to solve is to maximize the data rates of NOMA users. In other words, in this paper, we focus on maximizing sum rate (MSR) metric. To maximize the sum rate, it is necessary to consider the data rates of all users. This often means that it may be inevitable to sacrifice the data rates of few users in order to achieve overall higher sum rate.

As mentioned above, we assume two users can be allocated to each channel (*i.e.*, $N = 2K$), and the objective is to maximize the sum throughput of users. We assumed that there is a limit of total power P_T for BS which needs to be distributed to all users across the channels. It is important to note that the sum of users' power p_1^k and p_2^k must not exceed P_T as follows,

$$\sum_{k=1}^{K} (p_1^k + p_2^k) \le P_T \tag{5}$$

For power assignment, the JRA method[12] will be used which is an optimal power assignment solution using mathametical derivation. Since JRA method is able to find the optimal power given the channel allocations and resources, we propose to apply JRA method alongside with DRL channel allocations. The problem for MSR metric formulation can be written as,

$$\begin{aligned} &\max_{p_1, p_2} \sum_{k=1}^{K} \left[R_1^k(p_1^k, p_2^k) + R_2^k(p_1^k, p_2^k)\right], \\ &s.t.\ R_n^k \ge (R_n^k)_{min}, n = 1, 2, \forall k = 1, \ldots, K, \\ &\quad \sum_{k=1}^{K} (p_1^k + p_2^k) \le P_T, \\ &\quad 0 \le p_1^k \le p_2^k, \forall k = 1, \ldots, K, \end{aligned} \tag{6}$$

where $(R_n^k)_{min}$ is a minimum data rate requirement for user n allocated to k-th channel. In order to solve the (6), the optimization problem decomposes into the following subproblems for each channel k,

$$\begin{aligned} &\max_{p_1^k, p_2^k} R_1^k(p_1^k, p_2^k) + R_2^k(p_1^k, p_2^k) \\ &s.t.\ R_n^k \ge (R_n^k)_{min}, n = 1, 2, \forall k = 1, \ldots, K, \\ &\quad p_1^k + p_2^k = q^k, \\ &\quad 0 \le p_1^k \le p_2^k, \forall k = 1, \ldots, K, \end{aligned} \tag{7}$$

where q^k is a power budget for channel k.

As Zhu *et al.*[12] proposed, the solution for MSR metric is given by solving the subproblems,

$$p_1^k = \frac{\Gamma_2^k q^k - A_n^k + 1}{A_2^k \Gamma_2^k},$$
$$p_2^k = q^k - p_1^k, \tag{8}$$

where $A_n^k = 2^{\frac{(R_n^k)_{min}}{B_c}}$ and $A_n^k \geq 2$. As noted from (8), q^k plays a pivotal role to solve MSR metric. q^k is given by the waterfilling form,

$$q^k = \left[\frac{B_c}{\lambda} - \frac{A_2^k}{\Gamma_1^k} + \frac{A_2^k}{\Gamma_2^k} - \frac{1}{\Gamma_2^k}\right]_{\gamma^k}^{\infty}, \quad \gamma^k = \frac{A_2^k(A_1^k - 1)}{\Gamma_1^k} + \frac{A_2^k - 1}{\Gamma_2^k}. \tag{9}$$

To derive power budget q^k, Lagrangian multiplier method and bisection method are used, and its upper bound is set to infinity and lower bound is set to γ^k, meaning that if the derived power budget q^k is smaller than γ^k, q^k is set to γ^k, otherwise keep q^k.

The optimal power allocation problem is solved with the mathematical solution (*i.e.*, JRA), channel allocation problem remains to be solved. The optimal channel allocation method can be driven by using exhaustive search (ES) method, however, it consumes extraordinarily long time to find the optimal allocation. Previous works focused on solving an optimal power allocation problem, while leaving the channel allocation problem left with randomization, resulting an inefficient channel allocation. On the other hand, we are integrating JRA, achieving optimal power assignment and applying our own DRL method to find a sub-optimal solution for channel allocation with more efficient way.

Ⅲ. Reinforcement Learning Algorithm

In this paper, we utilized reinforcement learning algorithm to solve channel assignment problem by using multiple neural networks. In this section, details of how channels are assigned to users, using fully-connected neural network (FCNN), convolutional neural network (CNN), and attention-based neural network (ANN) will be explained.

Exhaustive Search (ES) method is significantly more inefficient compared to the approach we propose. ES method explores all possible channel allocations, represented by $\prod_{i=0}^{\frac{N-2}{2}} C(N-2i, 2)$ where N users are allocated and Nk = 2, in a given environment and calculates the sum rate for each case. However, as the number of users increases, the number of possible channel allocations grows exponentially. Therefore, in realistic scenarios with many users, finding the maximum sum rate through optimal channel allocation by using ES method is highly inefficient and nearly impossible. On the other hand, our proposed channel allocation method using DRL with replay memory utilizes previously learned experiences to train the model for near-optimal channel allocations demonstrating results close to the maximum sum rate, as shown in the Section Ⅳ. After training, our method consumes near-linear time complexity, typically within a second, compared to the ES method.

The fundamental of DRL formulation is to define state, action, and reward. Each component is represented as s_t, a_t, and r_t at time step t, corresponding to the state, action and reward, respectively.

A state is defined as a pair of user and channel information. The state space is $N \times K \times F$, where N is the number of users, K is the number of channels, and F is the number of features. By forming the state with the space of $N \times K$, every possible combination of user and channel information can be represented. The feature of the state represents the user and channel information itself and the number of features F vary from one to three; a state with $F = 1$ contains CNR information, a state with $F = 2$ contains CNR and distance information between users and the BS, and a state with $F = 3$ contains CNR, distance, and channel assignment status information. A CNR value of channel k is represented as CNR_k, a distance between user n and the BS is represented as d_n, and channel k's assignment status is represented as C_k. The value of the channel assignment status C_k is equal to the number of users assigned to the following channel. For instance, if the channel k has zero user assignment, then the channel status is $C_k = 0$. When the user n is assigned to channel k, then the status changes to $C_k = 1$. The channel status information allows the agent to be aware of assignable channels. It is important to know which channel is assignable, due to each channel can hold fixed number of users. In this case, since the N_k is set to 2, C_k ranges from 0 to 2.

Under NOMA system, to solve the channel assignment problem, action is assigning a user to a channel. Therefore, the action can be represented with one user and one channel as $a_t = (n, k)$. The selected state can also be interpreted as an action taken.

Reward is defined as a data rate (throughput) of each user. The reward for channel k at time step t can be expressed in two cases as follow,

$$r_t^k = \begin{cases} R_1^k(s_t), & \text{if the user in } s_t \text{ is} \\ & \text{first assigned to channel } k, \\ R_2^k(s_t), & \text{otherwise,} \end{cases} \quad (10)$$

due to the constraint $N_k = 2$. The objective of the downlink NOMA system is to maximize the sum of all users' data rates (rewards) as follows,

$$G_N^{MSR} = \max \sum_{i=1}^{N} r_i. \quad (11)$$

To solve the channel assignment problem, let's say the set of actions taken is $\zeta = \{a_1, a_2, \cdots, a_N\}$. Given some state set S, the conditional probability of ζ is as follows,

$$p_\theta(\zeta|S) = \prod_{i=1}^{N} p_\theta(a_t|S) \quad (12)$$

where θ is a policy parameter, used to update the policy using loss function. Variation of the reinforcement estimator[13] is used for the loss function, stabilizing the training by using the baseline model. The loss function is defined as the average rewards corresponding to the state set ζ as below,

$$Loss(\zeta|S) = \mathbb{E}_\zeta \left[G_N^{MSR}(\zeta) \right], \quad (13)$$

and the parameter θ from policy $p(\cdot)$ is updated via policy gradient method, utilizing the difference between the loss of online model and baseline model:

$$\nabla Loss(\zeta|S) = \mathbb{E}_\zeta \left[\left(Loss(\zeta|S) - Loss(\zeta^{bl}|S) \right) \nabla \log_{p_\theta}(\zeta|S) \right]. \quad (14)$$

The algorithm used for training is as Algorithm. 1. The training is performed on an episode basis, and performed until it reaches stopping criteria (line 1). Each episode creates a user with randomized location (line 3). Based on the coordinate of the corresponding user, the algorithm calculates the corresponding user's CNR.

After initializing the user information, the channel allocation for every user is excuted. Every time step, the online model samples a user to allocate to a channel based on the model's probability distribution (line 6). On the other hand, baseline model selects an user-channel pair with a highest probability (line 7). Because selected user and channel can not be selected once again, the process of masking selected pairs is required, leading a more efficient calculation. After repeating the mentioned steps, every user is allocated to all channels, then by using JRA method, powers are assigned, leading a sum rate.

When the actions are all taken, the result is saved into a replay memory (ζ, R, R^{bl}) (line 9). Sum rate of the online model R and the baseline model R^{bl} is calculated using the state sets ζ and ζ^{bl}, by performing the JRA method. Then, batch sized experiences are randomly sampled from replay memory (line 10), represented as δ and is as,

$$\delta \leftarrow \{\zeta_1, \zeta_2, \ldots, \zeta_{batch}\}. \quad (15)$$

δ is used for calculating loss and the loss is derived using the gradient from (14) (line 11).

The used optimizer in this paper for updating the policy gradient is the Adam optimizer[14]. With the optimizer, online model updates its own parameter θ as below,

$$\theta \leftarrow Adam(\theta, \nabla Loss(S|\delta)). \quad (16)$$

The update of the baseline model's parameter θ^{bl} is performed when the loss of the online model exceeds the loss of the baseline model as $\theta^{bl} \leftarrow \theta$ (*i.e.*, online model's sum of rewards exceeds baseline model's sum of rewards) (line 13, 14).

By following the steps of (15) and (16), it is indeed true that the computational complexity increases, due

to the use of replay memory despite the problem being an on-policy problem. The inefficiency of incorporating replay memory into an on-policy method arises from the nature of on-policy learnig. On-policy methods involve iterative evaluation and updating of the policy based on the most recent experiences. However, the experience replay memory contains past experiences that may not be representative of the current policy. This mismatch means that experiences stored in the memory are more likely to be irrelevant to the updated current policy. Still, the reason behind we incorporated the experience replay memory with REINFORCE algorithm is that we have noticed that utilizing the REINFORCE algorithm results in a narrow learning, unable to respond to generalized data. On the other hand, the incorporation improves the model to be able to optimally allocate joint resources within various generalized data, although consumes more time on learning the policy.

After each episode is completed, validation is performed. A validation set is created at the beginning of the training, when the NOMA environment is created. The number of validation performed is equal to the number of validation seeds. When the validation set is decided, maximum sum rate R_{max} and minimum sum rate $Rmin$ of each validation seed is calculated by using exhaustive search. The deriven sum rates are used to calculate the error rate. In order to calculate the error rate for each seed number, a sum rate from the baseline model R^{bl} is used. The error rate from the baseline model for validation *seed* is defined as (line 18),

$$Error_{seed}(R^{bl}) = \frac{R_{max} - R^{bl}}{R_{max} - R_{min}}. \tag{17}$$

If the error rate is below the predefined threshold, the validation is considered passed (line 23, 24). When every validation is passed for all seeds, the validation set is passed.

The stopping criteria for the training process are determined by whether the validation set passes and the loss threshold. At the end of every episode, stopping criteria are evaluated. When the validation set is passed and the loss between the online and baseline model is below the threshold, the stopping criteria are met. Then the parameters from baseline model are saved and the training process terminates.

Algorithm 1 Training algorithm for channel assignment

Input: State S, two models of NN; p_θ and $p_{\theta^{bl}}$, NOMA environment
Output: Trained model NN; channel assignment

1: **while** stopping criteria not met **do**
2: **for** each episode **do**
3: generate N user profiles with random seeds
4: **for** each step **do** ▷ env.step(action)
5: $p_\theta(s_t|S) \leftarrow$ output of NN
6: $\zeta \leftarrow$ sampling per $p_\theta(s_t|S)$
7: $\zeta^{bl} \leftarrow$ argmax sampling per $p_{\theta^{bl}}(s_t|S)$
8: **end for**
9: replay memory $\leftarrow (\zeta, R, R^{bl})$
10: $\delta \leftarrow \{\zeta_1, \zeta_2, ..., \zeta_{batch}\}$
11: $\nabla Loss(\delta|S)$ $= \mathbb{E}_\zeta[(Loss(\zeta|S) - Loss(\zeta^{bl}|S))\nabla \log p_\theta(\zeta|S)]$
12: $\theta \leftarrow Adam(\theta, \nabla Loss(S|\delta))$
13: **if** $Loss(\zeta|S) > Loss(\zeta^{bl}|S)$ **then**
14: $\theta^{bl} \leftarrow \theta$
15: **end if**
16: **end for**
17: **if** validation time step **then**
18: run validations $Error_{seed}(R^{bl}) \leftarrow \frac{R_{max} - R^{bl}}{R_{max} - R_{min}}$
19: **if** $Error_{seed} \leq threshold$ **then**
20: validation passed
21: **end if**
22: **end if**
23: **if** validation passed and $Loss < threshold$ **then**
24: save model $p_{\theta^{bl}}$
25: break
26: **end if**
27: **end while**

Ⅳ. Evaluations

The simulations were conducted in the following simulation settings. We assumed that there is a single BS which is an agent allocating and assigning channels and powers. Around the BS, there are N users scattered randomly from $50m$ to $300m$ which their minimum data rate is set to $(R_n^k)_{min} = 2bps/Hz$ $\forall k = 1, \ldots, K, n = 1, \ldots, N$. Since $N_k = 2$, the number of channels is $K = N/2$. The total bandwidth for the agent BS can use is $B_{tot} = 5MHz$.

The noise power spectral density for the environment is N_0 = -170dBm/Hz. The channel response of n-th user assigned to k-th channel is represented as h_n^k and it is defined as follows,

$$h_n^k = g_n^k d_n^{-\alpha} \tag{18}$$

where g_n^k is a Rayleigh fading distribution corresponds to user n and channel k, and $d_n^{-\alpha}$ is a distance loss between user n and BS. Here, α is a path loss coefficient which is set to α = 2.

The validation set was predefined before the training so that the maximum sum rate and minimum rate can be searched beforehand by exhaustive search. Every training episode utilized an unique instance by using a seed number enabling the model to learn more generalized knowledge about the NOMA policy. The validation was performed every 200 episodes of training to determine whether the loss and error rate from (14) and (17) met the stopping criteria.

In this section, to evaluate the performance of the proposed JRA-DRL method, various simulations effects of changes in learning rate, batch size, number of features, models and comparisons between JRADRL, JRA, and exhaustive search— are analyzed. Experimental parameters for evaluation of the framework were set as follows: learning rates vary by 0.001, 0.0005, and 0.0001, and batch sizes are 20, 40, and 80. The simulations are conducted observing the actual sum rates and convergence speeds to assess the performance of the proposed framework.

Fig. 2 illustrates achievable sum rates in different learning rates; 0.001, 0.0005, 0.0001. The largest learning rate (0.001) converged the fastest, completing training in just 280 seconds; however, resulted in the lowest overall sum rate. On the other hand, the smallest learning rate (0.0001) took the longest to converge, requiring 3,990 seconds to meet the stopping criteria, leading to the highest sum rate. The learning rate of 0.0005 would be a great alternative to achieve the reasonable results, as it significantly reduces the time took for the training compared to the learning rate of 0.0001, while still achieving high sum rates. However, the variance of the sum rates is large, therefore the fluctuations of sum rates were resulted.

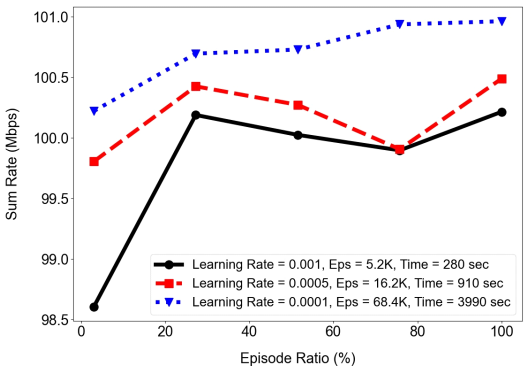


Fig. 2. Sum rate comparison of different learning rates with $N \times K \times F = 6 \times 3 \times 3$, batch size = 40 model = FCNN and P_T = 12W .

Fig. 3 shows the sum rates in different batch sizes; 20, 40, and 80. As the comparison of different learning rates from Fig. 2 did, the results for different batch sizes were very similiar. The smallest batch size of 20 was able to finish the training the fastest, nevertheless it yielded the lowest overall performance. Compared to this, the sum rate of the largest batch size of 80 was the highest, taking the longest time to complete the training. Similarly with the comparison of different learning rates, the intermediate batch size of 40 completed the training relatively quickly compared to the larger batch size of 80, while yielding high sum rates.

Fig. 4 represents the comparison of loss convergences when using different number of features for the state; (CNR_k), (CNR_k, d_n), and (CNR_k, d_k, C_k) for

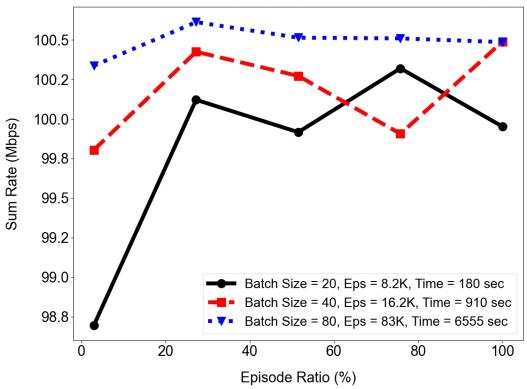


Fig. 3. Sum rate comparison of different batch sizes with $N \times K \times F = 6 \times 3 \times 3$, learning rate = 0.005, model = FCNN and P_T = 12W .

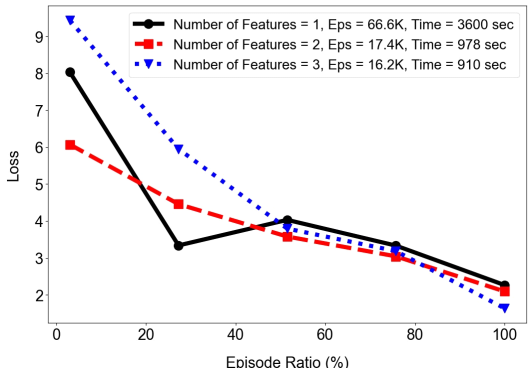


Fig. 4. Loss comparison of different number features with $N \times K = 6 \times 3$, learning rate = 0.005, batch size = 40, model = FCNN and $P_T = 12$W .

all $k = 1, \cdots, K$ and $n = 1, \cdots, N$. At the beginning of the training, all three agents with different state spaces have high loss values. Then as the number of training episodes increases, the loss converges towards zero. The agent with three features requires the most training time to converge, meaning that it took the longest to figure out the meaning of the state. In contrast, the agents with one feature and two features require much less training time to converge. Nevertheless, in the middle of the training, the agent with one feature's loss value abruptly increases, indicating unstable learning.

Fig. 5 represents the comparison of loss convergences when using different models; FCNN, ANN, and CNN. The loss values for the agents using the FCNN and CNN models decrease gradually and converge towards zero, while the CNN model requires

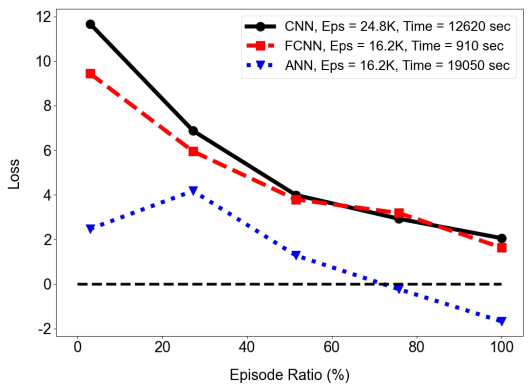


Fig. 5. Loss comparison of different models with $N \times K \times F = 6 \times 3 \times 3$, learning rate = 0.005, batch size = 40 and P_T = 12W.

the longest training episodes. However, although the ANN model required the fewest episodes to converge among the three models, its loss oscillates constantly. This means that learning the NOMA policy from the baseline model is not sufficiently stable, leading to consistent changes in the sign of the loss values. The reason for implementing a baseline model to policy gradient methods is to stabilize training and reduces the variance of the gradient estimates. Since the baseline model provides a stable reference value, monitoring the sign of the loss value enables the agent to stabilize the learning and reduce the variance. However, for the ANN model, frequent changes in the baseline model resulted in unstable training and learning. Furthermore, despite the fact that the number of episodes required for training the ANN and the FCNN models is nearly identical, the actual time consumed by the ANN model significantly exceeded the time when using the FCNN model.

Fig. 6 illustrates the performance comparison between three methods: the exhaustive search (ES) method, the JRA method, and the proposed JRA-DRL method. The methods are evaluated using the trained model and the evaluation is based on the resulting sum rates in four different NOMA environments (seeds). The four different NOMA environments used for the validation (or evaluation) are entirely different with the environments used for the training. The key difference is that the validation set comprised more generalized environmental settings, which makes it challenging for the standard policy gradient method to adapt effectively. Obviously, the exhaustive search

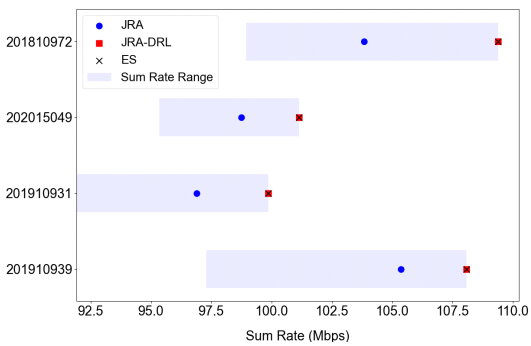


Fig. 6. Sum rate performance comparison of different training algorithms.

method achieves the maximum attainable sum rates across all seed numbers. The JRA method achieves high sum rates, though not the highest. The proposed method, JRA-DRL method, achieves sum rates that are very close to the maximum attainable sum rates in all four simulations. This shows that implementing experience replay into the policy gradient method stimulates the agent to adapt effectively to generalized environment.

Fig. 7 illustrates the sum rate comparison of different training algorithm with different total power P_T for BS; $2W$, $4W$, $8W$ and $12W$. The above shows that as the total power for BS P_T increases, the attainable sum rate also increases. As the JRA-DRL method yields higher sum rates near to maximum attainable sum rates from exhaustive search than the JRA method, the JRA-DRL method is proven to be able to achieve superior performance in all power levels.

Fig. 8 illustrates the sum rate comparison of different training algorithms with different number of NOMA users N; 4, 6 and 8. As the number of the NOMA users increases, the attainable sum rates decrease, because the users are forced to shared limited joint resources, resulting in a lower sum rates. In all simulation results, the JRA-DRL method achieves higher sum rates than the JRA method, which are very close to the sum rates of exhaustive search method.

Additionally, the sum rate performance of applying larger state sizes was evaluated with respect to minimum data rate R_{min} for all users, and the result is shown in Fig. 9. As mentioned in Section Ⅲ, the number of possible channel allocations grows exponentially when the number of users and channels increases. Due to this nature, the calculation of the error rate from Algorithm 1 was omitted in this simulation. As the minimum data rate R_{min} increases, the sum rate gradually decreases for all different states sizes of $N \times K = 10 \times 5$, $N \times K = 20 \times 10$, and $N \times K = 30 \times 15$. However, compared to Fig. 8, the simulation for smaller state sizes, increasing the state size leads to an improvement in sum rate performance.

Finally, Fig. 9 also illustrates the performance comparison between JRA-DRL method with replay memory and without replay memory. The proposed JRADRL method with replay memory shows the higher sum rate performance over the JRA-DRL method without replay memory for all state sizes and

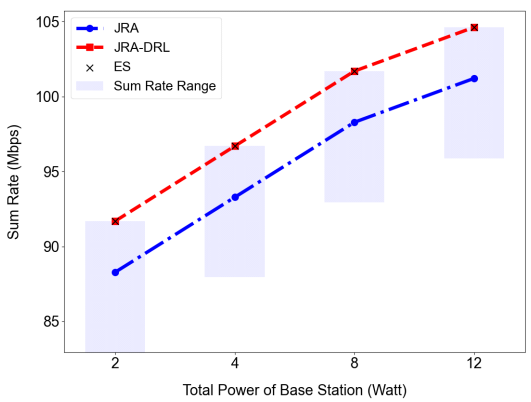


Fig. 7. Sum rate comparison of different training algorithms with different total power for base station.

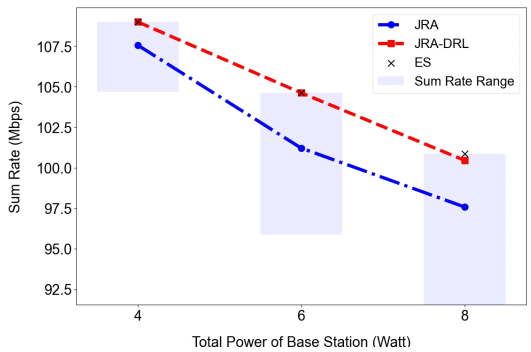


Fig. 8. Sum rate comparison of different training algorithms with different number of NOMA users.

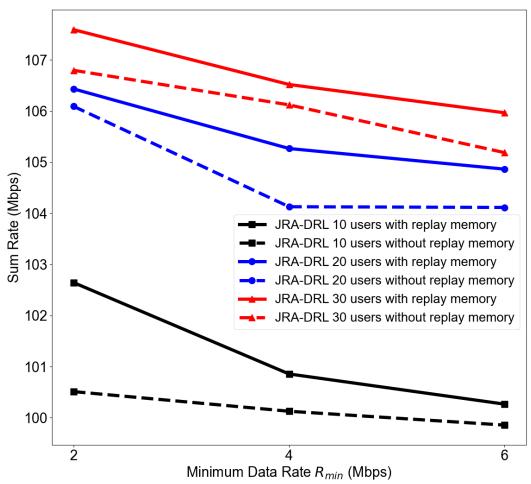


Fig. 9. Sum rate comparison for JRA-DRL method with and without replay memory of different state sizes when requiring different minimum data rate (R_{min}).

minimum data rates R_{min} . This not only demonstrates that our proposed method can handle large input spaces, but also shows that the incorporation of the policy gradient method with replay memory ensures the generalization of policy learning, leading to higher sum rate performance.

As demonstrated by the simulations shown in this section, the proposed JRA-DRL method exhibits superb performance. The JRA method is capable of allocating users with optimal powers but lacks the ability to assign users to adequate channels. Due to this, the JRA method performs above average but is incapable to reach near to the highest performance. In contrast, the proposed JRA-DRL method reaches nearmaximum performance with reasonable training time, also showing the ability to adapt to completely new (or generalized) environments by using the experience replay.

Ⅴ. Conclusions

In this paper, we propose a reinforcement learning-based framework to solve channel and power allocation problem in a downlink NOMA system, and provide various simulation results. The framework takes two steps which are channel allocation and power assignment. At each time step, the model considers the current channel allocation status of users and decides which user should be allocated to which channel. The channel allocation is performed by integration of replay memory and the REINFORCE algorithm which enables more generalized learning of the NOMA policy. After the initial channel allocation step is completed, the subsequent power assignment step is carried out using the JRA method, which has been proven to be an effective solution for the power optimization problem.

The simulations were conducted with respect to the number of state features, batch sizes, types of models, and learning rates. Overall, the simulations demonstrates that the proposed framework can successfully learn policy from NOMA system with fast convergence and has the ability to handle comprehensive data.

References

[1] C. Zhang, P. Patras, and H. Haddadi, “Deep learning in mobile and wireless networking: A survey,” *IEEE Commun. Surv. and Tuts.*, vol. 21, no. 3, pp. 2224-2287, 2019. (https://doi.org/10.1109/COMST.2019.2904897)

[2] D.-W. Kim and K.-S. Shin, “Multiple access control protocol using deep-reinforcement learning in heterogeneous wireless networks,” *J. KICS*, vol. 49, no. 1, pp. 88-94, Jan. 2024. (https://doi.org/10.7840/kics.2024.49.1.88)

[3] Y. Yu, T. Wang, and S. C. Liew, “Deep-reinforcement learning multiple access for heterogeneous wireless networks,” *IEEE J. Sel. Areas in Commun.*, vol. 37, no. 6, pp. 1277-1290, Jun. 2019. (https://doi.org/10.1109/JSAC.2019.2904329)

[4] D. Ye and M. Zhang, “A self-adaptive sleep/wake-up scheduling approach for wireless sensor networks,” *IEEE Trans. Cybernetics*, vol. 48, no. 3, pp. 979-992, Mar. 2018. (https://doi.org/10.1109/TCYB.2017.2669996)

[5] C. He, Y. Hu, Y. Chen, and B. Zeng, “Joint power allocation and channel assignment for noma with deep reinforcement learning,” *IEEE J. Sel. Areas in Commun.*, vol. 37, no. 10, pp. 2200-2210, Oct. 2019. (https://doi.org/10.1109/JSAC.2019.2933762)

[6] Y.-F. Liu and Y.-H. Dai, “On the complexity of joint subcarrier and power allocation for multi-user ofdma systems,” *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 583-596, Feb. 2014. (https://doi.org/10.1109/TSP.2013.2293130)

[7] W. Ahsan, W. Yi, Z. Qin, Y. Liu, and A. Nallanathan, “Resource allocation in uplink noma-iot networks: A reinforcement-learning approach,” *IEEE Trans. Wireless Commun.*, vol. 20, no. 8, pp. 5083-5098, Aug. 2021. (https://doi.org/10.1109/TWC.2021.3065523)

[8] C.-L. Wang, C.-C. Hsieh, Y.-C. Ding, and S.-H. Huang, “Power allocation for downlink noma systems with imperfect channel estimation,” in *2021 IEEE WCNC*, pp. 1-7,

2021.
(https://doi.org/10.1109/WCNC49053.2021.9417462)

[9] Y. Zhang, X. Zhao, S. Geng, et al., "Power allocation algorithms for stable successive interference cancellation in millimeter wave noma systems," *IEEE Trans. Veh. Technol.*, vol. 70, no. 6, pp. 5833-5847, Jun. 2021. (https://doi.org/10.1109/TVT.2021.3077270)

[10] Z. Ding, P. Fan, and H. V. Poor, "Impact of user pairing on 5g nonorthogonal multiple-access downlink transmissions," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6010-6023, 2016. (https://doi.org/10.1109/TVT.2015.2480766)

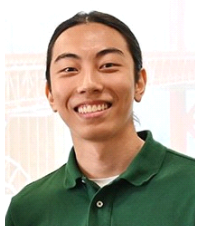
[11] Z. Ding, M. Peng, and H. V. Poor, "Cooperative non-orthogonal multiple access in 5g systems," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1462-1465, Aug. 2015. (https://doi.org/10.1109/LCOMM.2015.2441064)

[12] J. Zhu, J. Wang, Y. Huang, S. He, X. You, and L. Yang, "On optimal power allocation for down-link non-orthogonal multiple access systems," *IEEE J. Sel. Areas in Commun.*, vol. 35, no. 12, pp. 2744-2757, Jul. 2017. (https://doi.org/10.1109/JSAC.2017.2725618)

[13] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," *Advances in NIPS,* vol. 12, S. Solla, T. Leen, and K. Müller, Eds., 1999. [Online] Available: https://proceedings.neurips.cc/paper_files/paper/1999/file/464d828b85b0bed98e80ade0a5c43b0fPaper.pdf

[14] D. Kingma, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

WooSeok Kim



2019-2025 : B.S. degree, Seoul, Sangmyung University.
<Research Interests> Model-based planning using reinforcement learning, applied machine learning, and robotics.
[ORCID:0009-0008-3626-6659]

Jeonghoon Lee

2019-2025 : B.S. degree, Seoul, Sangmyung University
2025-Present : B.S. degree, Seoul, Sangmyung University
<Research Interests> Parallel Programming, Game Server, Computer Network, Data Synchronization and Consistency, and Distributed System
[ORCID:0009-0008-9141-0473]

Sangho Kim

2019-Present : B.S. degree, Seoul, Sangmyung University.
<Research Interests> Reinforcement learning.
[ORCID:0009-0003-5422-2899]

Taesun An

2022 : Associate of Science in Computer Engineering degree, Seoul, Myongji College.

2022-Present : B.S. degree, Seoul, Sangmyung University.

<Research Interests> Artificial intelligence, non-orthogonal multiple access (NOMA), resource allocation, deep reinforcement learning, and wireless network.

[ORCID:0009-0000-6225-7440]

WonMin Lee

2018 Present : B.S. degree, Seoul, Sangmyung University.

<Research Interests> Internet of Things, and Reinforcement Learning.

[ORCID:0009-0008-1606-2288]

Dowon Kim

2016-2022: B.S. degree, Seoul, Sangmyung University.

2022-2024 : M.S. degree, Seoul, Sangmyung University.

<Research Interests> Internet of Things, wireless communication protocols, network information system, and reinforcement learning.

[ORCID:0009-0003-9155-7890]

Kyungseop Shin

2005-2009 : B.S. Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea.

2009-2011 : M.S. Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea.

2011-2015 : Ph.D. Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), South Korea.

2015-2017 : Senior Researcher, KT Cooperation, Institute of Convergence Technology Infrastructure Research Institute 5G TF.

2017 2020 : Assistant Professor, School of Computer Science, Semyung University.

2020-Present : Associate Professor, Department of Computer Science, Sangmyung University.

<Research Interests> Wireless Communication for IoT, Battery Health Management System, and Reinforcement Learning.

[ORCID:0000-0002-3867-1921]