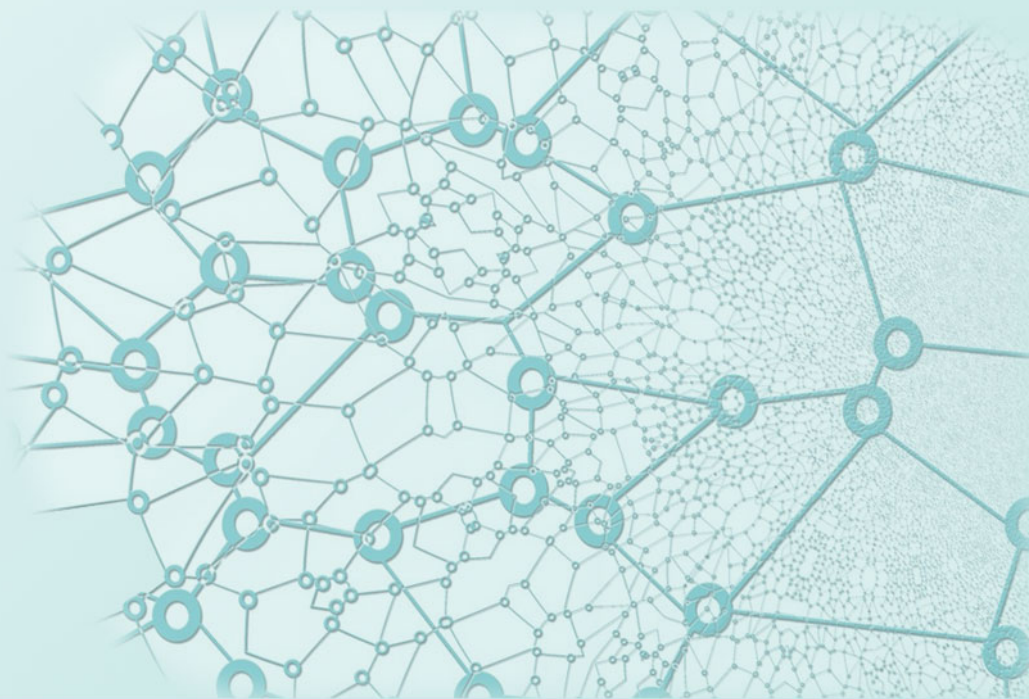


KNOM Review | 28권 1호, 2025년 8월 | ISSN 2287-1543 (Online)

KNOM REVIEW

Network Operations & Management

Vol. 28 No.1 | AUG. 2025



한국통신학회 통신망운용관리연구회

The Committee on Korean Network Operations and Management (KNOM)

Korean Institute of Communication Sciences (KICS)

<http://www.knom.or.kr>

목 차

제28권 제1호, 2025년 8월

자율주행 엣지의 Continual Learning 적응 성능 분석	
..... 이지은, 테오도라 아두푸, 김윤희	1
FedAttnPSO: Non-IID 환경에서 연합학습 정확도 향상을 위한 개선된 알고리즘 제안	
..... 장선영, 김종욱, 최미정	10
그래프 신경망 기반 초기 해집단 평가 모델을 이용한 동적 네트워크 라우팅 최적화	
..... 이도영, 김태연	21

자율주행 엣지의 Continual Learning 적응 성능 분석

이 지 은*, 테오도라 아두푸*, 김 윤 희[°]

A Performance Analysis of Continual Learning on an Edge for Autonomous Driving

Jieun Lee*, Theodora Adufu*, Yoonhee Kim[°]

요 약

자율주행 시스템의 안정성이 중요시되면서 로컬 엣지(Edge)에서 GPU를 기반으로 딥러닝을 적용하는 연구가 진행 중이다. 그러나 자율주행용 엣지 딥러닝 학습 과정에서 실시간으로 유입되는 새로운 데이터에 대한 적응력이 뛰어나지 못한다는 문제점이 제기되고 있다. 본 논문에서는 실시간 적응 능력 개선과 정확도 향상을 위해 엣지 상에서 Continual Learning(CL, 연속학습) 알고리즘을 딥러닝 학습 과정에 도입할 것을 제안한다. NVIDIA Jetson AGX 기반 엣지 환경에 EWC와 Rehearsal을 적용한 실험을 통해 CL 알고리즘이 정확도를 손해보지 않으면서 실시간 환경에 빠르게 적응할 수 있음을 보였다.

Key Words : Edge Computing, Continual Learning, EWC, Rehearsal, Resource-constrained Systems, On-device Learning

ABSTRACT

As the reliability of autonomous driving systems becomes increasingly critical, research has been actively conducted on applying deep learning using GPUs as local edge devices. However, a major challenge remains: edge-based deep learning for autonomous driving often lacks the ability to quickly adapt to newly incoming data in real time. In this paper, we propose integrating Continual Learning (CL) algorithms into the deep learning training process on edge devices to enhance real-time adaptability and improve accuracy. Through experiments applying EWC and Rehearsal on an NVIDIA Jetson AGX-based edge environment, we demonstrate that CL algorithms can rapidly adapt to real-time environments without compromising accuracy.

※ 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (과제번호: RS-2025-24534879).

♦ First Author : Sookmyung Women's University, Department of Computer Science, mariewldms@sookmyung.ac.kr

° Corresponding Author : Sookmyung Women's University, Department of Computer Science, yulan@sookmyung.ac.kr

* Sookmyung Women's University, Department of Computer Science, theoadufu@sookmyung.ac.kr

논문번호 : KNOM2025-01-01, Received July 14, 2025; Revised August 08, 2025; Accepted August 25, 2025

I. 서 론

최근 고도화된 GPU를 기반으로 자율주행용 엣지 컴퓨팅에 딥러닝을 도입하기 위한 방안이 활발하게 제안되고 있다. 특히, NVIDIA Jetson AGX Orin [1], NVIDIA Jetson AGX Xavier [2] 등 AI 추론이 가능하도록 GPU 및 자원 환경이 최적화된 자율주행용 엣지에서 딥러닝을 수행한 사례가 보고된다. 안전성과 실시간성이 중요하게 여겨지는 자율 주행 환경에서, 다양한 특성의 교통 데이터를 학습하기 위한 엣지에서의 딥러닝은 필수 불가결한 요소로 여겨지며 실 시스템에 도입하기 위한 접근이 시도되고 있다.

일반적으로 딥러닝 모델을 통해 학습 및 추론이 이루어지는 고성능 GPU와 달리 엣지는 GPU의 제한된 연산 능력과 메모리 용량의 한계로 인해 딥러닝 시간, 성능 측면에서 제약이 따른다. 문제 해결을 위해 엣지에 적합한 추론 알고리즘 최적화 기법 [3]이 제안되고 있지만, 추론이 아닌 새로운 데이터 학습 과정에서의 문제를 해결하지는 못한다. 또한 자율주행용 엣지는 실시간 유입되는 다양한 교통 데이터에 적응해야 하기 때문에 빠른 적응이 더욱 중요하다.

Continual Learning(CL, 연속 학습)은 동적인 환경에서 새로운 과제를 학습했을 때 기존 과제에 대한 성능이 급격하게 저하되는 파괴적 망각(Catastrophic forgetting) 현상을 방지하기 위해 제시된 방법이다 [4]. 그림 1은 CL의 이론적 핵심 및 학습 라이프라인을 도식화한 구조도다 [4]. 그림 1의 a는 CL이 다른 성격의 데이터 유입에 적응하는 것의 중요성을, 그림 1의 b는 CL의 기준점(안정성, 일반화, 유연성)을, 그림 1의 c는 학습 과정에 방법론별 CL이 도입되는 지점을, 그림 1의 d는 이론과 방법을 기반으로 CL의 실제 적용 과정을 보인다. CL 방법론을 학습 과정에 적용해 과거 데이터 학습 기억을 보존하면서 새로운 데이터를 학습하는 과정의 균형을 맞추는 것이 CL의 핵심 주안점이다. CL 알고리즘은 Computer Vision(CV), Natural Language Processing(NLP), Reinforcement Learning(RL)등의 분야에 적용될 수 있으며, 최근 연구에서는 실시간 얼굴 이미지 검출 및 보호 [5] 등 실생활 문제 해결 목적으로 활용되었다. 이처럼 CL은 유연성(Plasticity)을 기반으로 실시간으로 유입되는 새로운 데이터에 신속하게 적응할 수 있다

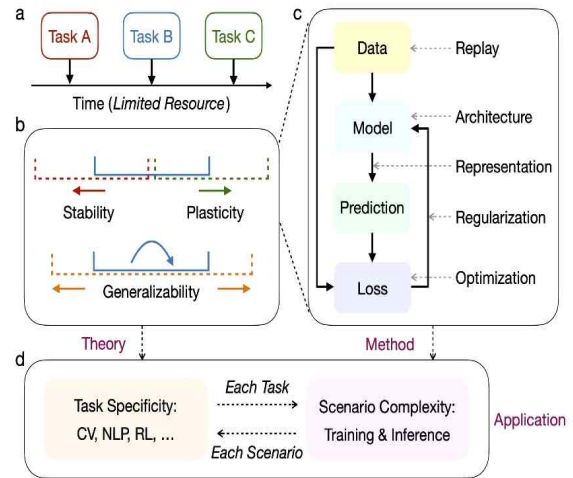


그림 1. CL(Continual Learning)의 개념적 구조 [4]
Fig. 1. A conceptual framework of continual learning

는 장점을 가져 자원 및 환경 제한된 상황에서 해결책이 될 수 있다. 따라서 본 논문은 추론용 엣지의 딥러닝 과정에서 새로운 데이터에 빠르게 적응할 수 있는 방법인 CL 알고리즘의 성능 분석을 제안한다.

본 논문의 구성은 다음 순서를 따른다. 1장의 서론에 이어 2장은 본 논문의 주제와 관련된 연구 및 연구 동기를 서술한다. 3장은 선행 연구를 기반으로, 실험에 활용할 대표적인 CL 알고리즘과 신경망 모델을 소개한 뒤 성능 측정 기준이 될 주요 지표를 제시한다. 4장에서 실험 과정 및 분석 결과를 설명하고, 5장은 결론을 도출한 후 향후 연구를 기술한다.

II. 관련 연구 및 연구 동기

본 절에서는 엣지에서 GPU를 활용해 딥러닝을 수행한 선행 연구 사례를 서술하고, 기존 CL 연구를 기반으로 자율주행 엣지에 CL 알고리즘을 도입해야 할 필요성을 보인다.

1. 관련 연구

기존 연구 [6,7]에서는 엣지 내 GPU를 활용해 딥러닝 연산을 수행하고 추론했다. 데이터셋을 통해 학습된 YOLOv8, YOLOv9 모델을 기반으로 추론용 임베디드 컴퓨터인 NVIDIA Jetson AGX Orin에서 추론 과정을 거쳐 모델별 성능을 분석했다 [6]. 또한, 엣지 컴퓨팅의 자원 한계를 개선하기 위해 ConvNet, ResNet50, EfficientNet-B0을 변형한

새로운 아키텍처(CorPAR)를 제안해 추론 과정에서 초당 22프레임을 처리하고, TP Rate(True Positive Rate)가 71%에 달해 고성능 GPU와 비슷한 성능을 보였다 [7]. 이처럼 엠티는 딥러닝 기반 추론은 가능하지만, 모델 학습은 고성능 GPU에서 진행되었다는 점에서 학습은 미흡하다.

2. 연구 동기

지속적으로 변화하는 환경에 노출되는 자율주행 엠티는 정확한 객체의 주행 위치 예측을 위해 실시간으로 유입되는 데이터에 적응해야 한다. 그러나 엠티 환경은 GPU 연산 성능과 메모리가 제한적이기 때문에 고성능 GPU와 유사한 정확도의 성능을 내기 위해서 더 많은 시간 자원이 필요하다. 이러한 상황에서 CL이 효과적인 해결책이 될 수 있다. CL은 새로운 데이터의 유입에 유연하게 적응하면서 짧은 시간 안에 비슷한 정확도를 도출해 자율주행 엠티 상 자원 제약을 완화할 수 있다.

III. CL 기법 분석

본 절에서는 CL 알고리즘 Elastic Weight Consolidation(EWC), Rehearsal을 단순 학습 방식에 적용한 방식을 서술하고, Convolutional Neural Network(CNN) 기반 이미지 분류 모델을 설명한다. 마지막으로 실험 결과를 분석을 위한 주요 성능 지표를 서술한다.

1. CL 알고리즘

그림 1의 c는 다섯 종류의 CL 방법론이 학습 과정에서 도입되는 구간을 표기한다. CL 방법론에는 Regularization-Based Approach, Replay-Based Approach, Optimization-Based Approach, Representation-Based Approach, Architecture-Based Approach가 있다 [4]. 본 연구는 대표적인 CL 기법 중 Regularization-Based Approach와 Replay-Based Approach 방식을 활용한다.

EWC는 Regularization-Based Approach에 해당하는 알고리즘이다. 기존 loss에 정규화된 term을 더해 모델의 가중치 변화를 최소화하는 방식으로, 중요한 가중치일수록 패널티로 값이 변화하지 않아 이전 기억을 보존한다 [8]. EWC를 백본(Backbone) 모델에 적용하는 과정에서 배치 단위로 학습이 수행될 때마다 수식 (1) 방식과 같이 새로 들어온 데이터에 대한 loss $L_{new}(\theta, D)$ 와 $F_i(\theta_i - \theta_i^*)^2$ 을 더

한 값에 가중치 $\lambda/2$ 를 곱한 결과를 더해 loss를 갱신한다. F_i 는 각 파라미터의 중요도를 의미하며, $(\theta_i - \theta_i^*)^2$ 는 파라미터의 변화량을 제공한 결과다.

$$L_{EWC}(\theta, D) = L_{new}(\theta, D) + \frac{\lambda}{2} \sum_i F_i (\theta_i - \theta_i^*)^2 \quad (1)$$

Rehearsal은 Replay-Based Approach에 해당하는 알고리즘이다. 버퍼에 기존 학습 샘플을 저장하고 새로운 학습이 진행될 때 현재 태스크와 함께 재학습하여 망각 현상(catastrophic forgetting)을 방지하는 방법이다 [9]. 리플레이 버퍼에 학습 값의 일부를 저장하고, 새로운 학습 데이터를 이용해 학습할 때 버퍼 내 샘플을 재학습한다. 따라서 수식 (2)의 방식으로 산출한 새로운 학습 내용 기반 loss $L(\theta, D_{new})$ 와 버퍼 기반 loss $L(\theta, D_{mem})$ 에 가중치 μ 를 곱한 값을 더해 최종 loss를 갱신한다 [10].

$$L_{Re}(\theta, D) = L(\theta, D_{new}) + \mu L(\theta, D_{mem}) \quad (2)$$

2. CNN 모델 선정

이미지 데이터셋을 학습하며 CL 알고리즘을 적용할 대표적인 합성곱 신경망 기반 이미지 분류 모델로 ResNet50과 EfficientNet_B0을 선정했다. ResNet50은 잔차 연결(residual connection)을 도입해 딥러닝 모델의 깊이를 효과적으로 늘릴 수 있도록 설계된 안정적인 대표적 CNN 모델이다 [11]. EfficientNet_B0은 모델의 깊이, 너비, 해상도를 균형 있게 확장하는 방식(compound scaling)을 통해 적은 파라미터로 높은 정확도를 달성하는 경량화된 최신 모델이다 [12].

3. 성능 지표

CL의 성능을 파악하기 위한 3가지 기준으로 [4] 실험 결과를 분석할 성능 지표를 선정했다.

첫 번째로, 새로운 데이터셋이 학습되는 과정에서 이전 기억을 보존하는 안정성(Stability) 평가 지표로 Root Mean Squared Error(RMSE)를 활용한다. RMSE는 CL 알고리즘 적용 학습이 완료된 후 평가용 데이터셋의 일부를 기반으로 미래 위치를 예측하고, 이 예측값과 실제값의 차이를 측정하는 회귀 평가 지표이다.

두 번째로, 불규칙하고 다양한 입력에 대응할 수 있는지 확인하는 일반화(Generalizability) 지표로 딥

러닝 과정에서의 loss를 활용한다. 새로운 분포의 데이터셋이 투입되었을 때의 loss 변화를 기반으로 모델의 일반화 능력을 간접적으로 평가한다.

마지막으로, 새로운 데이터셋에 빠르게 적응하는지를 확인하기 위한 유연성(Plasticity) 지표로 Converge Step(CS)과 Loss Reduction Rate(LRR)를 활용한다.

$$CS = \min \{t \in [1, T] | \bar{L}_t \leq \epsilon\} \quad (3)$$

CS는 누적 평균 loss값의 추세에서 최종 평균 loss와 차이를 비교하며 오차가 ϵ 이 되는 지점 최소 스텝을 확인해 기준점에 얼마나 빠르게 수렴하는지 판정한다. 수식 (3)에서 t 는 스텝 변수, T 는 전체 스텝 수, $\bar{L}_t$ 는 t 스텝의 누적 평균 loss 값이다.

$$LRR = \frac{\bar{L}_{\max} - \bar{L}_T}{T - t_{\max}} \quad (4)$$

LRR은 수식 (4)와 같이 최대 loss값에서 최종 loss값을 뺀 값을 step 간격으로 나누어 절댓값 기울기를 이용해 개선 속도를 파악한다. $\bar{L}_{\max}$ 는 최대 누적 평균 loss, $\bar{L}_T$ 는 최종 누적 평균 loss, T 는 전체 스텝 수, $t_{\max}$ 는 $\bar{L}_{\max}$ 인 지점의 스텝이다.

IV. 성능평가

본 절에서는 실험에 사용된 자율주행 차량의 모션 예측을 위한 데이터셋과 실험용으로 선정한 엣지 Nvidia Jetson Orin AGX에 대해 설명한다. 이후 시나리오 설계 과정 및 엣지에서 CL 성능 분석을 진행한 결과를 서술한다.

1. 데이터

실험은 Woven by Toyota [13] 에서 제공한 예측 데이터셋(prediction-dataset)을 사용했다. 해당 데이터셋은 zarr 포맷으로 제공되며, 자율주행 모빌리티 장치의 주변 환경을 평균 10Hz의 속도로 112시간동안 캡처한 것이다. 해당 데이터 셋에는 100개의 장면(여러 에이전트가 존재하는 도로 위의 연속적인 상황), 24838개의 프레임(한 시점의 스냅샷), 1893736개의 에이전트(자동차, 사람 등 도로 위의 객체), 316008개의 신호등이 존재한다. 해당 구조의

zarr 파일을 기반으로 l5kit 파이썬 라이브러리의 AgentDataset 객체를 통해 학습에 적합한 데이터셋을 추출했다. 데이터 산출에 필요한 설정 조건은 제공된 agent motion config.yaml 파일의 기준을 따랐으며 [14], 그 결과 111634개의 데이터셋이 산출되었다. 생성된 각각의 데이터셋은 하나의 에이전트에 초점을 맞춘 단일 에이전트 예측용 데이터셋이다. 데이터셋은 한 에이전트의 래스터화된 이미지, 현재 및 과거 위치, 미래 타겟 위치, 방향 각도, 차량 크기 정보 등 에이전트의 주행 상황에 대한 지표를 포함한다.

생성된 111634개의 데이터셋은 PyTorch의 DataLoader를 통해 배치 단위로 분류되어 ResNet50과 EfficientNet을 통해 학습된다. 백본 모델은 해당 에이전트의 미래 위치를 예측하며, 해당 값을 실제 미래 위치와 비교해 loss를 산출한다.

본 연구에서는 엣지 상에서 CL이 수행되는 상황을 구현하기 위해 전체 데이터셋을 동일한 크기의 세 개의 데이터셋으로 분할한다. CL 학습 알고리즘 적용을 위해 세 가지로 쪼개진 각 데이터셋은 명확히 구분될 수 있는 특징을 가져야 하므로, 에이전트의 속도를 기준으로 단계를 분류했다.

데이터셋은 세 가지의 다른 주행 환경에서 데이터를 추출하는 상황을 가정한다. 각 데이터셋의 속도(target velocities) 속성의 첫 번째 원소값을 기준으로 모든 데이터셋을 작은 값부터 차례대로 정렬한 뒤 세 개의 단계로 쪼갰다. 첫 번째 단계는 0.0m/s-0.2m/s의 속도로 주행하는 데이터셋의 집합이다(Low-velocity Task). 주차장에서 정지되어 있거나 정차하는 환경에서 데이터를 수집하는 상황을 가정한다. 두 번째 단계는 0.2m/s - 2.09m/s의 속도로 주행하는 데이터셋의 집합이다(Moderate-velocity Task). 주행속도가 제한되어 있거나 정체되는 구간에서 서행하는 에이전트들의 데이터를 수집하는 상황을 가정한다. 마지막 단계는 2.09m/s - 30.99m/s의 속도로 주행하는 데이터들의 집합이다(High-velocity Task). 고속도로를 비롯한 일반적인 도로주행 상황에서 데이터를 수집하는 상황을 가정한다.

2. 실험 환경

엣지인 NVIDIA Jetson AGX Orin에서 데이터를 학습하고 추론했다. Jetson AGX Orin 시리즈는 AI 시스템을 엣지에서 구현할 수 있도록 설계된 임베디드 시스템이다. 학습 및 추론이 진행된 Jetson

AGX Orin 32GB는 Ampere Architecture 기반 GPU와 8코어 CPU가 하나의 32GB 메모리를 공유하는 SoC(System on Chip) 아키텍처다 [15]. 실제 AI 기반 안전운전 시스템, 지능형 교통 모니터링 시스템에 사용 [16]되므로 자율주행 데이터셋을 활용한 검증에 적합하다. 표 1은 본 연구에 사용된 엣지의 연산 성능 및 사양을 기술한다 [15].

표 1. 실험에 사용된 엣지(Jetson AGX Orin 32GB)주요 사양

Table 1. Configuration of the Edge AI (NVIDIA Jetson AGX Orin)

Specification	Details
Model	Jetson AGX Orin 32GB
AI Performance	200 TOPS(INT8)
GPU	NVIDIA Ampere architecture (1792 NVIDIA CUDA cores, 56 Tensor Cores)
CPU	8-core ARM Cortex-A78AE v8.2 64-bit CPU 2MB L2 +4MB L3
Memory	32GB 256-bit LPDDR5 204.8 GB/s
Power	15W-40W

3. 실험 시나리오

CL 알고리즘의 성능 평가 전 자원 활용 및 결과에 영향을 미치는 변인을 통제했다. 그림 2, 그림 3은 학습에 앞서 데이터셋 샘플링 비율과 배치(batch) 크기를 결정하기 위해 수행한 실험 결과다.

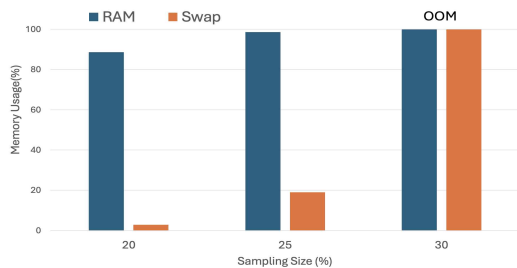


그림 2. 데이터셋 샘플링 크기에 따른 학습 시작 시점 RAM, Swap 점유율
Fig. 2. RAM and Swap Usage at Training Start According to Dataset Sampling Size

실험 전 데이터셋을 메모리에 적재하는 과정에서 주 메모리 용량이 32GB인 Jetson AGX Orin은 전체 데이터 적재가 불가능했다. 10% 단위로 샘플링

한 결과 그림 2와 같이 30%의 데이터셋을 샘플링했을 때부터 더 이상 잔여 메모리가 존재하지 않아

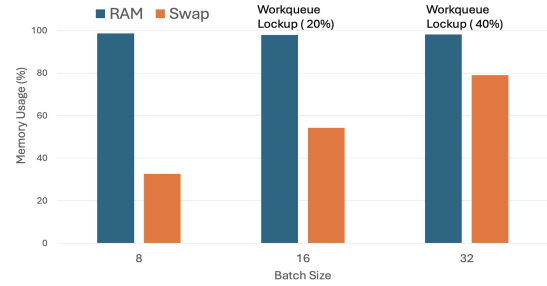


그림 3. 딥러닝 중 배치 크기에 따른 엣지의 최대 RAM, Swap 점유율

Fig. 3. Maximum RAM and Swap Usage on Edge During Deep Learning by Batch Size

OOM(Out Of Memory)으로 작업이 중단됐다. 25% 샘플링 결과 OOM은 발생하지 않았으나 RAM 점유율은 98.6%, Swap 메모리 점유율은 20%였다. Swap 메모리는 RAM이 가득 찼을 때 비활성 페이지를 디스크로 이동해 물리 메모리를 확보하기 위한 보조 수단으로, 메모리 부족을 완화하지만 RAM보다 속도가 느리다 [17]. 따라서 RAM 잔여 공간이 없는 상태에서 CPU, GPU기반 추론 시 속도가 느려져 12%의 잔여 RAM 공간이 있는 20% 데이터셋으로 수치를 고정했다. 실험에 활용된 전체 데이터셋은 22325개이며, 각 세 개의 단계는 전체 데이터의 1/3개인 7442개의 데이터셋을 포함한다.

일반적으로 모델 학습 과정에서 배치 크기는 GPU 메모리 할당량에 비례한다. 따라서 데이터셋을 7442개로 고정하고 배치 크기를 조정했다. 그림 3은 배치 크기에 따른 최대 메모리 점유율을 나타낸다. 학습 상황에서 배치 크기가 8일 때는 5번의 실험 중 5번 모두 정상적으로 학습 및 추론이 가능했다. 반면 배치 크기가 16일 때는 20%, 배치 크기가 32일 때는 40%의 확률로 워크큐 락업(Workqueue Lockup) [18,19]이 발생했다. 워크큐 락업은 8개의 CPU 코어 중 특정 CPU 내의 커널 워크큐 내 워커에서 특정 작업이 오랫동안 진행되어 대기한 채로 작업이 중단되는 버그다. 배치 크기가 16, 32일 때 Swap 메모리를 50% 이상 사용해 CPU가 Swap으로부터 데이터를 불러오는 과정에서 스레드가 중단되었다. 따라서 중단없이 결과를 추출하기 위해 배치 크기를 8로 결정했다.

4. 실험 분석

실험은 2가지 딥러닝 모델 ResNet50과

표 2. CL 알고리즘 학습 성능 지표 비교
Table 2. Continual Learning Performance Comparison

Backbone	Method	Average Loss			Accuracy	Training Time	CS ($\epsilon=10$)	LRR
		Low velocity Task	Moderate velocity Task	High velocity Task	RMSE			
EfficientNet	Naive	0.62	2.46	35.86	12.03	592.66	406	0.21
	EWC	0.61	2.45	35.97	12.81	1135.94	443	0.35
	Rehearsal	1.19	3.89	146.07	8.30	1127.38	208	0.09
ResNet50	Naive	0.62	2.56	43.63	13.83	639.64	428	0.21
	EWC	0.62	2.56	42.68	11.96	1098.96	488	0.37
	Rehearsal	1.26	4.53	168.64	10.45	1192.22	181	0.08

EfficientNet에서 CL 미적용 단순 학습 방식, EWC, Rehearsal의 성능을 측정하여 총 6가지 상황에서 안정성, 유연성, 일반화 능력을 비교했다. 딥러닝 모델과 CL 알고리즘을 기반으로 속도별로 분할된 세 데이터셋을 학습한 뒤, 평가 과정에서 성능을 평가한다. 각 실험은 다섯 번씩 수행되었으며, 산출된 결과물은 다섯 번의 실험 결과의 평균값이다. CPU 연산량을 최소화하는 선에서 동일 조건으로 성능을 비교하기 위해 배치 단위의 데이터 로딩 과정에서 하나의 프로세스를 병렬 처리하지 않고 순차 처리했다.

표 2는 6가지 상황에 따른 단계별 loss, 학습이 진행된 학습 시간, 평가 시 정확도를 측정하는 RMSE, Converge Step(CS), Loss Reduction Rate(LRR)이다.

실험 결과, EfficientNet에서 EWC와 Rehearsal 모두 단순 학습 방식에 비해 학습 시간이 1.9배 길었고, ResNet50에서 EWC는 1.7배, Rehearsal은 1.8배 길었다. EWC는 추가로 Fisher Information Matrix를 적용하고, Rehearsal은 리플레이 함수를 기반으로 과거 loss를 추가로 계산해 새로운 loss를 갱신해 매 스텝마다 추가 연산으로 학습 시간이 약 2배 길어졌다.

안정성과 일반화 능력 분석을 위해 학습 과정의 세 단계에서 loss를 측정하고, 학습 종료 후 평가 과정에서 RMSE를 측정했다. loss는 배치 단위로 데이터가 로드될 때 매 스텝마다 산출되며, 표 2에 명시된 loss는 각 단계별 스텝 수인 930개 loss의 평균값이다. RMSE는 속도 별로 분류되지 않은 평가용 데이터셋에서 각 장면의 초반 100개 프레임을 학습한 후, 이어지는 50개 프레임을 예측해 예측 경로와 실제 경로의 차이를 측정해 계산한 평균의

제공된 오차다. EWC는 EfficientNet, ResNet50에서 각 단계별 loss와 RMSE가 단순 학습 대비 ± 2 이

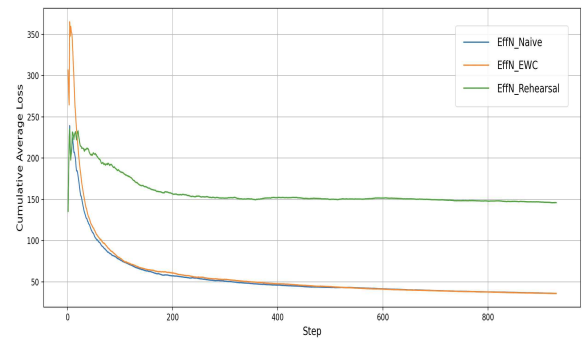


그림 4. 세 번째 학습 데이터셋(High-velocity Task)의 step 별 누적 평균 loss 변화 추이
Fig. 4. Cumulative Average Loss Trend per Step on High-Velocity Task (Task 3)

내로 미미했다. Rehearsal은 EfficientNet, ResNet50 모두 마지막 단계에서 loss가 단순 학습 방식보다 2배 크지만, RMSE는 3.7, 3.4 만큼 작았다. Rehearsal이 과거 데이터의 일부를 버퍼에 넣어 재 학습하기 때문에 새 데이터에 대한 일반화 능력은 부족해 loss가 크고, 이전 기억을 보존하는 안정성이 뛰어나 평가용 데이터에 대한 정확도는 높아 RMSE가 낮은 트레이드오프를 보였다.

유연성과 자원 한계를 분석하기 위해 적응 속도, 개선 속도, 메모리 사용량을 측정했다. 첫 번째로, 적응 속도는 Rehearsal이 가장 빨랐다. 그림 4는 마지막 단계(High-velocity Task)에서 스텝별 누적 loss 평균값을 나타낸 그래프로, 노이즈를 제거해 변화 추이를 명확히 보여준다. 세 가지 방식은 930번의 스텝을 거치며 점차 학습되면서 특정 값으로 수렴되는 추이를 보이지만, 속도 및 최종 loss 값에는 차이가 있다. 스텝 1에서 loss는 Rehearsal은

135, 단순 학습 방식은 136으로 차이가 없지만 스텝 930의 최종 loss와 ϵ 차이 나는 값에 도달하는

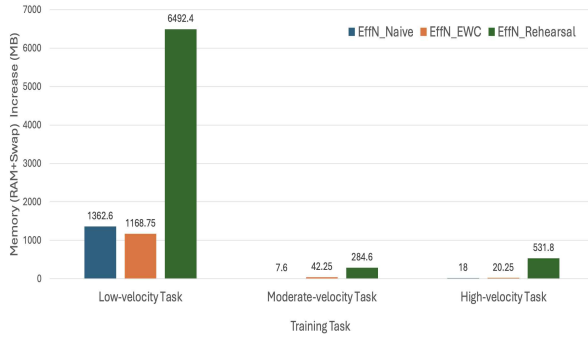


그림 5. 데이터셋 딥러닝 학습 과정에서의 각 단계별 엣지 메모리(RAM+Swap) 변화량(MB)
Fig 5. Memory Usage Change (RAM + Swap, MB) per Task During Deep Learning on Edge Device

데 걸리는 스텝은 Rehearsal이 더 짧았다. 기준점 ϵ 을 10으로 설정 후 CS(Converge Step)를 측정한 결과 표 2와 같이 ResNet50, EfficientNet에서 모두 Rehearsal이 208, 181로 가장 작았다. Rehearsal은 새로운 학습 loss에 버퍼 기반으로 기존의 학습 loss가 더해져서 새로운 loss가 갱신되어 새로운 loss가 반영되는 비율이 단순 학습에 비해 적어 loss값 변동에 제약이 있기 때문이다. 따라서 실제 학습 과정에서 CS를 전체 스텝 수로 설정해도 전체 스텝 수인 930과 모델 최적화 정도에 차이가 없다.

두 번째로, 개선 속도는 EWC가 가장 뛰어났다. LRR을 계산했을 때, 표 2와 같이 EfficientNet과 Resnet50에서 EWC의 절댓값이 각각 0.35, 0.37로 가장 컸다. 그림 4에서 EWC는 첫 번째 스텝 loss가 단순 학습 방식보다 170 만큼 크고, 최대 누적 평균 loss가 350 이상으로 가장 크지만 930 스텝의 loss는 35로 동일하다. EWC는 이전 단계 학습으로 모델이 업데이트된 상황에서, 페널티로 큰 값의 가중치가 변하지 않아 초반 loss는 크지만 개선 속도가 빨라 단순 학습 방식과 동일한 누적 loss 평균으로 수렴하기 때문이다.

마지막으로 CL은 단순 학습 방식보다 비슷하거나 더 많은 메모리를 사용했다. 그림 5는 단계별 세 종류의 데이터가 투입되는 상황에서 전 단계와 비교했을 때 증가한 메모리량(RAM+Swap)을 나타낸다. 단순 학습, EWC, Rehearsal 모두 첫 번째 단계인 가장 느린 속도의 데이터셋이 투입되었을 때 모델 초기화 및 데이터셋 로딩으로 메모리 증가량이 가장 크다. 또한 이후로는 증가량이 10% 미만으

로 크게 감소한다는 점은 동일하지만, 메모리 증가 정도에 차이가 있다. 세 단계에서 모두 EWC와 단순 학습의 RAM, Swap 메모리 증가량은 194MB, 35MB, 2MB 차이로 유사했다. EWC는 스텝마다 가중치에 따라 페널티를 조정한다. 매 스텝마다 추가적인 연산이 필요하지만 메모리를 더 사용하지 않아 추가 메모리 부담이 발생하지 않는다. 반면 Rehearsal은 단순 학습 방식보다 메모리 사용량이 5310MB, 277MB, 513MB 높았다. Rehearsal은 이전 데이터셋을 버퍼에 저장해 재학습하며 loss를 두 번 산출하기 때문에 추가적인 메모리 할당이 필요하기 때문이다. CL 알고리즘은 추가 loss 및 행렬 연산으로 CPU 및 GPU 연산량이 많아지면서 단순 학습 방식과 메모리를 비슷하게 사용하거나 더 많이 사용했다.

V. 결 론

본 논문은 자율주행 엣지에서 딥러닝 모델 학습 과정에 CL 알고리즘을 적용한 뒤 학습 과정에서 모델 오차와 추론 정확도를 실험했다. 이어서 엣지의 모델 적응 속도 및 메모리 사용량을 비교했다. 분석 결과 딥러닝 모델에 CL 알고리즘을 적용하면 정확도를 손해보지 않으면서 적응 속도, 개선 속도가 빨라져 실시간 적응 능력이 향상되었다. 향후 연구에서는 제한된 연산 자원과 실시간성이 요구되는 다양한 엣지 환경에서 안정적으로 CL이 수행될 수 있도록 CL 알고리즘의 엣지 환경별 적응성을 높이는 작업이 필요하다.

References

- [1] NVIDIA, NVIDIA Jetson Orin, Retrieved July, 14,2025, from <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/>
- [2] NVIDIA, NVIDIA Jetson Xavier, Retrived July, 14,2025, from <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-xavier-series/>
- [3] Zihen Jiang, Tianqi Chen and Mu Li, "Efficient Deep Learning Inference on Edge Devices", SysML18, Stanford, CA USA, pp. 1-3, April 2018
- [4] Liyuan Wang, Xingxing Zhang, Hang Su and

- Jun Zhu, "A Comprehensive Survey of Continual Learning: Theory, Method and Application", in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 8, pp. 5362-5383, Aug. 2024
- [5] 신중훈, "Faster R-CNN과 Continual Learning을 적용한 실시간 얼굴 검출 및 보호.", 충북대학교, 국내석사학위논문, pp.1-52, 2024
- [6] 조홍석, 권희준, 정해영, 이수안, "NVIDIA Jetson AGX Orin 기반의 YOLOv8, YOLOv9 모델을 이용한 화재 탐지", 한국정보과학회, 학술논문집, pp.1764-1766, 2024-06-26
- [7] J. N. Chaudhari, H. Galiyawala, P. Sharma, P. Shukla and M. S. Raval, "Onboard Person Retrieval System With Model Compression: A Case Study on Nvidia Jetson Orin AGX," in IEEE Access, vol. 13, pp. 8257-8269, 2025
- [8] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska et al., "Overcoming Catastrophic Forgetting in Neural Networks," Proc. Natl. Acad. Sci., vol. 114, no. 13, pp. 3521 - 3526, 2017
- [9] E. Verwimp, M. De Lange, and T. Tuytelaars, "Rehearsal revealed: The limits and merits of revisiting samples in continual learning"(2021), Retrived July, 14, 2025, <https://arxiv.org/abs/2104.07446>.
- [10] Shengqin Jiang, Daolong Zhang, Fengna Cheng, Xiaobo Lu, Qingshan Liu, "DuPt: Rehearsal-based continual learning with dual prompts", Neural Networks, Volume 187, pp.1-10, 2025
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun "Deep Residual Learning for Image Recognition"(2015), Retrived July, 14, 2025, <https://arxiv.org/abs/1512.03385>
- [12] Mingxing Tan, Quoc V.Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"(2019), Retrieved July, 14, 2025, <https://arxiv.org/abs/1905.11946v5>
- [13] Woven By Toyota, Retrived July, 14, 2025, <https://woven.toyota/en/>
- [14] woven-planet, l5kit, Retrieved July, 14, 2025, https://github.com/woven-planet/l5kit/blob/master/examples/agent_motion_prediction/agent_motion_config.yaml
- [15] NVIDIA, NVIDIA Jetson AGX Orin Series, Retrived July, 14, 2025, <https://www.nvidia.com/content/dam/en-zz/Solutions/gtc/tcf21/jetson-orin/nvidia-jetson-agx-orin-technical-brief.pdf>
- [16] Advantech, MIC-733-AO5A1, Retrived July, 14, 2025, <https://iotmart.advantech.co.kr/>
- [17] Red Hat, Chapter 15. Swap Space, https://docs.redhat.com/en/documentation/red_hat_enterprise_linux/7/html/storage_administration_guide/ch-swapspace
- [18] NVIDIA, Kernel locked on Orin, Retrived July, 14, <https://forums.developer.nvidia.com/t/kernel-locked-on-orin/245450/12>
- [19] The Linux kernel, Workqueue, Retrived July, 14, 2025, <https://docs.kernel.org/core-api/workqueue.html>

이 지 은 (Jieun Lee)



2022년 3월~현재 : 숙명여자대
학교 컴퓨터과학과 학사과정

<관심분야> 엣지 컴퓨팅, 클라우드 컴퓨팅, 딥러닝,
임베디드 시스템

테오도라 아두푸 (Theodora Adufu)



2013년 : 가나대학교 컴퓨터과
학 및 경제학과 졸업(학사).
2016년 : 숙명여자대학교 컴퓨
터과학과 졸업(석사).
2022년~ : 현재 숙명여자대
학교 컴퓨터과학과 박사
과정.

<관심분야> 엣지 컴퓨팅, 퀀텀 컴퓨팅, 컨테이너,
클라우드 컴퓨팅, GPU 스케줄링, 딥러닝

김 윤 희 (Yoonhee Kim)



1991년 : 숙명여자대학교 전
산학과 졸업(학사).
1996년 : Syracuse University
전산학과 졸업(석사).
2000년 : SyracuseUniversity
전산학과 졸업(박사).
1991년~1994년 : 한국전자통

신연구원 연구원

2000년~2001년 : Rochester Institute of
Technology 컴퓨터공학과 조교수.

2001년~2016년 : 숙명여자대학교 컴퓨터과학부 교
수.

2017년~현재 : 숙명여자대학교 소프트웨어학부 교
수.

<관심분야> 엣지 컴퓨팅, 퀀텀 컴퓨팅, 클라우드 시
스템, 워크플로우 제어

FedAttnPSO: Non-IID 환경에서 연합학습 정확도 향상을 위한 개선된 알고리즘 제안

장 선 영*, 김 종 욱*, 최 미 정[°]

FedAttnPSO: A Novel Algorithm for Improving Federated Learning Accuracy in Non-IID Environments

Sun-young Jang*, Jong-Wouk Kim*, Mi-Jung Choi[°]

요 약

연합학습은 데이터를 공유하지 않고 분산된 환경에서 학습을 수행할 수 있도록 설계된 방법으로, 개인정보보호가 중요한 현대사회에서 실용성을 높이 평가받고 있다. 하지만 각 클라이언트가 보유한 데이터의 양과 분포 및 특성이 모두 상이한 Non-IID 환경에서 모델 성능이 저하되는 한계가 존재한다. 이 문제를 해결하기 위해 알맞은 클라이언트 선택과 가중치 집계 문제를 해결해야 한다. 본 논문에서는 이 문제들을 해결하기 위해 FedAttnPSO를 제안한다. FedAttnPSO는 각 클라이언트의 손실값을 기준으로 성능이 우수한 상위 N개의 클라이언트를 선택한다. 선택된 클라이언트들 중 Attention Mechanism을 통해 성능이 높은 클라이언트의 기여도를 높이고 성능이 낮은 클라이언트의 영향력을 제한하여 동적으로 집계 가중치를 부여한다. 또한 클라이언트 및 서버 측에서 PSO (Particle Swarm Optimization) 알고리즘을 활용한 학습 최적화를 통해 전체적인 모델의 성능을 향상시킨다. FedAttnPSO와 다른 연합학습 알고리즘을 비교하기 위해 표준 이미지 데이터셋인 CIFAR-10, CIFAR-100을 사용하여 비교하였으며, Non-IID 환경에서의 실험을 위해 FEMNIST를 사용하여 FedAttnPSO를 평가 및 비교했다. 실험 결과, 모든 데이터셋에서 기존 알고리즘 대비 성능이 향상됨을 확인하였다.

Key Words : Federated Learning, Particle Swarm Optimization, Attention Mechanism, Non-IID

ABSTRACT

Federated learning is a distributed training approach designed to operate without sharing raw data, making it particularly valuable in modern society where data privacy is paramount. However, in non-IID environments—where the amount, distribution, and characteristics of data vary significantly across clients—the performance of federated models tends to degrade. To address this issue, it is essential to carefully select clients and effectively aggregate their model updates. In this paper, we propose FedAttnPSO, a novel aggregation algorithm designed to tackle these challenges. FedAttnPSO first selects the top-N performing clients based on their local loss values. Among the selected clients, it dynamically assigns aggregation weights using an attention mechanism, enhancing the contributions of high-performing clients while limiting the influence of low-performing ones. To evaluate the performance of FedAttnPSO, we conducted experiments using the standard image datasets CIFAR-10 and CIFAR-100, as well as FEMNIST to simulate non-IID conditions. The results demonstrate that FedAttnPSO consistently outperforms existing federated learning algorithms across all datasets.

※ 이 논문은 보건복지부의 재원으로 한국보건산업진흥원의 보건의료기술연구개발사업지원에 의하여 이루어진 것임(과제 번호: HI23C0733).

* First Author : Kangwon National University Department of Data Science, jsy0708@kangwon.ac.kr

[°] Corresponding Author : Kangwon National University Department of Computer Science and Kangwon National University IGP, in Medical Bigdata Convergence, mjchoi@kangwon.ac.kr

* Kangwon National University Department of Computer Science, jw.kim@kangwon.ac.kr

논문번호 : KNOM2025-01-04, Received July 14, 2025; Revised August 08, 2025; Accepted August 25, 2025

I. 서 론

최근 스마트폰, 웨어러블 기기와 같은 엣지 디바이스의 발전과 함께 병원, 금융 기관 등에서 생성되는 데이터 양 또한 기하급수적으로 늘어나고 있다. 이러한 데이터는 개인 정보 보호 관점에서 민감한 정보를 포함하고 있는 경우가 많아 외부로 직접 공유하거나 중앙 서버에 집중시키는 것이 어렵다. 이러한 배경 속에서 연합학습(Federated Learning, FL)은 원시 데이터를 중앙 서버로 전송하지 않고 분산된 환경에서 협력적으로 머신러닝 모델을 훈련하는 새로운 패러다임으로 주목받고 있다. 연합학습은 각 클라이언트가 자신의 로컬 데이터로 모델을 개별 훈련한 뒤, 학습된 모델의 파라미터만을 중앙 서버로 전송한다. 서버는 수집된 파라미터를 종합하여 하나의 전역 모델을 구축하기 때문에 데이터 프라이버시를 효과적으로 보호할 수 있다. 연합학습에서 가장 널리 사용되는 알고리즘은 FedAvg (Federated Averaging)으로, 매 라운드마다 무작위로 선택된 클라이언트들의 로컬 모델의 파라미터를 데이터 크기에 비례하여 단순 평균하는 방식으로 동작한다.

하지만 실제 연합학습 환경은 각 클라이언트가 보유한 데이터의 양, 분포, 특성이 다른 Non-IID (Non-Independent and Identically Distributed) 라는 본질적인 한계를 가진다. 각 클라이언트의 데이터는 고유한 사용 패턴에 따라 생성되므로 편향된 특성을 가지며, 이는 전역 모델의 일반화 성능을 저해하는 핵심 원인이 된다. FedAvg 알고리즘은 이러한 Non-IID 환경에서 명확한 문제를 드러낸다. FedAvg의 무작위 클라이언트 선택 방식은 데이터의 품질이나 대표성과 무관하게 클라이언트를 선택하므로 편향되거나 노이즈가 많은 데이터를 가진 클라이언트가 학습에 참여하여 전역 모델의 성능을 저하시킬 수 있다[2],[3]. 또한 데이터의 양에만 비례하여 가중치를 부여하는 단순 가중치 집계 방식은 데이터의 품질이나 학습 기여도를 전혀 고려하지 못하며, 이로 인해 편향된 대용량 데이터가 모델에 과도한 영향을 미치는 결과를 초래할 수 있다[4]. 이러한 문제를 해결하기 위한 연구가 진행되었지만 실제 연합학습 환경에서는 아직 많은 어려움을 겪고 있다[5]. 따라서 Non-IID 환경에서 연합학습의 성능을 높이기 위해서는 클라이언트 선택과 가중치 집계의 고도화를 넘어, 다양한 클라이언

트 모델로부터 최적의 전역 모델을 탐색하는 고도화된 최적화 관점의 접근이 필요하다.

이에 본 논문에서 제안하는 FedAttnPSO는 두 가지 문제를 다음과 같이 해결한다. 클라이언트 선택 문제에 대응하기 위해 FedAvg의 무작위 방식 대신 모델의 손실값(loss)을 기반으로 학습에 기여할 가능성이 높은 클라이언트를 선별하여 학습의 효율성과 안정성을 높인다. 가중치 집계 문제를 해결하기 위해 기존의 데이터 크기 기반 단순 평균 방식을 어텐션 기반 가중치 계산과 PSO 기반 점진적 업데이트라는 2단계의 정교한 방식으로 대체한다. 어텐션 메커니즘을 활용하여 각 클라이언트 모델이 전역 모델에 기여하는 중요도를 계산한다. 이렇게 계산된 중요도와 입자 군집 최적화(Particle Swarm Optimization, PSO) 알고리즘을 통해 기존 전역 모델을 새로운 전역 모델로 안정적이고 점진적으로 이동시킨다. 이러한 방식은 특정 라운드의 편향된 집계 결과가 전역 모델을 한 번에 오염시키는 것을 방지하는 역할을 할 수 있다. 더 나아가, 클라이언트 측에서도 PSO를 활용한 로컬 학습을 수행하여 로컬 모델 자체의 품질을 높이고 전체 학습 과정의 안정성을 더한다. 결론적으로 FedAttnPSO는 명확한 기준의 클라이언트 선택, 정교한 가중치 집계 방식, 그리고 모델 학습 최적화를 통해 Non-IID 환경의 한계를 극복하고 안정적으로 전역 모델을 구축 및 배포하는 것을 목표로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 알고리즘에 적용된 핵심 기술들과, Non-IID 환경에서 연합학습 성능 향상을 위해 수행된 관련 연구들을 소개한다. 3장에서는 제안 알고리즘의 구체적인 구조와 동작 원리를 설명하고, 4장에서는 실험 설정 및 결과 분석을 다룬다. 마지막으로 5장에서는 본 연구의 결론을 제시하고, 향후 연구 방향에 대해 논의한다.

II. 배경 지식 및 관련 연구

이 장에서는 본 논문에서 제안한 FedAttnPSO의 이해를 돕기 위한 관련 핵심 기술들의 배경 지식과 연합학습 성능 향상을 위한 기존 연구들을 소개한다. 나아가 본 논문이 기존 연구와 구별되는 차별점을 설명한다.

2.1 배경 지식

2.1.1 Particle Swarm Optimization

Particle Swarm Optimization (PSO)[6]는 Kernedy와 Eberhart가 제안한 전역 최적화 알고리즘이며, Swarm Intelligence 개념에 기반을 두고 있다. 군집 지능은 단순한 개체 간의 지역적인 상호작용을 통해서 복잡하고 지능적인 집단 행동이 자발적으로 일어나는 자연 현상을 인공지능에 적용한 분야이다. 특히 PSO 알고리즘은 새 떼의 이동, 물고기 무리의 행동 등에서 영감을 얻은 방식이며, PSO 알고리즘에서는 최적해를 찾기 위해 입자(Particle)를 가지고 있다. 각 입자는 자신의 위치와 속도를 가지며, 이를 사용하여 집단적으로 최적해를 찾아간다. 입자는 매 반복마다 자신의 최적 경험과 전체 입자 중 가장 우수한 위치를 참조하여 새로운 위치를 결정하며, PSO의 핵심 수식은 다음 수식 (1) - (2)와 같다:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i - x_i^t) + c_2 \cdot r_2 \cdot (g - x_i^t) \quad (1)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (2)$$

- v_i : 입자 i 의 속도
- x_i : 입자 i 의 위치
- p_i : 입자 i 의 개인 최적 위치
- g : 전역 최적 위치
- w : 관성 계수
- c_1, c_2 : 가속 계수(인지 계수, 사회적 계수)
- r_1, r_2 : [0,1] 범위의 난수

관성 계수(w)는 이전 속도를 얼마나 유지할지 결정하는 계수로서, 관성 계수가 클수록 더욱 멀리 이동하여 탐색이 강화되고, 작을수록 수렴이 강화된다. 가속 계수(c_1, c_2)는 목표 지점으로 얼마나 빠르게 끌려갈지 조절한다. 인지 계수(c_1)는 개인적 요인으로, 자신의 최적 경험으로 되돌아가려는 정도이다. 사회적 계수(c_2)는 사회적 요인으로써 전역 최적 위치로 끌려가는 정도이다. r_1 과 r_2 는 무작위 난수로서 매 반복마다 새로 생성된다. 입자마다 다른 방향성과 크기로 이동하도록 해줌으로써 무작위성을 부여하여 탐색 공간을 넓히고 지역 최적점에 갇히지 않도록 한다. 이러한 PSO 알고리즘은 탐색과 수렴의 균형을 유지하면서 점진적으로 최적해에 접근하는 능력이 뛰어나기 때문에 연합학습의 집계를 넘어 각 클라이언트의

로컬 모델 최적화와 중앙 서버에서 글로벌 모델 최적화 단계에 적용하여 전체적인 학습 성능을 높이는 데 효과적이다.

2.1.2 Attention Mechanism

어텐션 메커니즘(Attention Mechanism)은 입력 정보 중 중요한 부분에 집중하여 더 큰 비중을 두고 처리하는 방식으로, 자연어처리 분야에서 시작되어 이후 컴퓨터 비전 등 다양한 분야로 확장되었다. 특히 입력 데이터의 각 요소가 서로에게 어떤 영향을 미치는지 학습하고, 이 상호작용의 중요도를 동적으로 조절하는 데 유용하다. 기존의 단순한 평균 방식에서는 모든 입력 요소가 동일한 중요도로 처리되지만 Attention Mechanism은 각 요소의 상대적 중요도를 학습하여 이를 기반으로 출력에 반영한다. 예를 들어 문장을 번역할 때 특정 단어가 문장의 의미에 더욱 큰 영향을 미친다면 그 단어에 더욱 높은 가중치를 부여하여 번역 품질을 높일 수 있다.

기존에는 클라이언트마다 상이한 데이터의 양, 품질, 분포를 가지고 있음에도 불구하고 단순 평균 방식으로 글로벌 모델로 집계를 진행하였기 때문에 전체 모델의 성능이 저하될 수 있다는 문제가 존재한다. 이때 어텐션 메커니즘을 연합학습에 적용하여 각 클라이언트가 글로벌 모델 업데이트에 얼마나 영향을 미쳤는지 파악할 수 있다. 이에 따라 각 클라이언트에 대한 중요도를 파악하고, 중요도에 따라 글로벌 모델 업데이트를 위해 집계할 때 집계 가중치를 다르게 부여함으로써 보다 효과적이고 유연한 집계가 가능하다.

2.2 관련 연구

2.2.1 클라이언트 선택 개선 연구

Chen 외[7]는 FedAvg의 무작위 클라이언트 선택 방식이 Non-IID 데이터와 클라이언트 간 자원 이질성으로 인해 학습 효율을 크게 저하한다고 지적하며, 이를 개선하기 위한 새로운 방법으로 FedGRA(Federated Grey Relational Analysis)를 제안하였다. FedGRA는 무작위 선택 대신 그레이 관계 분석(GRA) 이론을 기반으로 클라이언트를 선택한다. 장치 이질성(CPU 및 RAM의 사용량), 데이터 이질성(훈련 손실 및 가중치 발산), 그리고 모든 클라이언트가 특정 라운드 수 내에서 최소 한 번은 훈련에 참여하도록 보장하는 공

정성 보장이라는 세 가지 주요 지표를 고려하여 각 클라이언트의 그레이 관계 등급(GRG)을 계산하고 이 등급이 높은 클라이언트를 선택하는 방식이다. 하지만 선택된 클라이언트를 글로벌 모델로 집계하는 과정에는 FedAvg의 기본적인 집계 방식인 가중 평균 방식을 사용하였다. FedGRA는 MNIST 2NN 실험에서 80% 정확도 달성에 FedAvg보다 69%의 라운드 감소로, 수렴 속도의 향상을 보였다.

Nguyen 외[8]는 데이터 프라이버시를 침해하지 않으면서 클라이언트 데이터를 기반으로 클라이언트를 선택하는 FedDSS (Federated Data Similarity Selection) 프레임워크를 제안한다. 데이터 통계 유사성과 왜도 유사성을 측정하여 각 클라이언트의 데이터 분포의 모양이 얼마나 유사한지 측정한다. 이를 바탕으로 각 클라이언트마다 자신과 가장 유사한 N개의 이웃 클라이언트를 찾아 N-SNN을 구성하고, N-SNN을 기반으로 공정성을 고려한 K-단계 random walk 알고리즘을 사용하여 높은 유사성을 가진 클라이언트를 선택한다. 선택된 클라이언트를 연합 평균(FedAvg) 방식으로 집계하여 전역 모델을 업데이트한다. 실험 결과 IID 및 Non-IID 환경에서 FedAvg 대비 더욱 빠른 수렴을 달성하였다.

Marfo 외[9]는 효율적인 클라이언트 선택과 견고한 개인 정보 보호 메커니즘이 핵심 과제를 언급하며 FedAvg의 무작위 클라이언트 선택 방식의 한계를 개선하는 새로운 방법을 제안한다. 먼저, 각 라운드에서 사용 가능한 클라이언트들을 데이터 품질 및 계산 용량과 같은 요소를 사용하여 계산된 유틸리티 점수(utility scores)를 기반으로 평가한다. 유틸리티 점수가 높은 K개의 클라이언트를 모델 훈련에 선택하고, 이 과정은 성능과 시스템 제약 조건에 따라 동적으로 조절한다. 그리고 개인 정보 보호를 위해 차등 프라이버시 및 Fault Tolerance를 통합하여 데이터 추론 방지와 실패로부터의 복구 가능성을 높인다. 선택된 클라이언트들을 집계하는 과정에 구체적인 집계 방식은 언급되지 않았지만, Aggregate updates라는 용어를 통해 표준적인 연합 학습 집계 절차가 사용되었음을 유추할 수 있다. 비교 대상으로는 ACFL 및 FedL2P가 사용되었으며, 기존 베이스라인 대비 정확도가 7% 향상되었으며, 훈련 시간 또한 25% 가량 감소되어, 제안하는 방법이 정확도와 훈련 시간을 크게 개선하고 시스템의 견고성을 높였음을 입증하였다.

2.2.2 가중치 집계 개선 연구

Shi 외[10]는 FedAvg의 데이터 크기 기반 집계 의 모델 성능 저하 및 불안정한 수렴 문제를 해결하기 위해 FedAWA (Federated learning with Adaptive Weight Aggregation)라는 새로운 방법을 제안한다. FedAWA는 클라이언트 벡터라는 개념을 도입하여 집계 가중치를 적응적으로 최적화한다. 클라이언트 벡터는 로컬 데이터의 변화와 모델 업데이트의 방향을 반영하고, 글로벌 벡터는 클라이언트 벡터들을 집계 가중치로 가중 평균한 값이다. 클라이언트 벡터와 병합된 글로벌 벡터 간의 정렬 정도를 사용하여 전역 최적화 방향과 밀접하게 정렬되는 로컬 모델에 높은 집계 가중치를 할당하여 전역 모델의 안정성과 일반화 성능을 향상시킨다. 제안하는 방법은 Non-IID 환경에서 전반적으로 높은 성능을 달성하였다.

Ye 외[11]는 로컬 및 전역 카테고리 분포 간의 불일치가 유익한 보완 지표가 될 수 있다는 분석을 기반으로 새로운 방법인 FedDisco (Federated Learning with Discrepancy-Aware Collaboration)를 제안한다. 각 클라이언트는 자신의 로컬 카테고리 분포와 전역 카테고리 분포 간의 불일치 수준을 계산한다. 이 계산은 클라이언트 측에서 이루어지므로 추가적인 데이터 공유 없이 프라이버시를 보존할 수 있으며, 서버는 클라이언트로부터 수집한 불일치 수준 값을 바탕으로 각 클라이언트의 데이터셋 크기와 로컬 불일치 수준을 활용하여 집계 가중치를 결정한다. 새로 결정된 불일치 인자 집계 가중치를 사용하여 로컬 모델을 집계하여 전역 모델을 업데이트한다. 하지만 클라이언트 선택에 대해서는 FedAvg에서 사용되는 무작위 클라이언트 샘플링 방법을 사용한다. FedDisco는 다양한 데이터셋과 Non-IID 환경에서 기존 알고리즘 대비 높은 성능 향상을 보였다.

Bang 외[12]는 딥 강화학습(DRL)을 활용해 배치 크기와 가중치 집계를 조정하는 BWA (Batch size and Weighted aggregation Adjustment) 메커니즘을 제안하였다. 전역 모델 정확도를 최대화하기 위한 최적화 문제는 여러 개의 국소 최적해를 가질 수 있는 복잡한 구조로, 기존의 볼록 최적화 기법으로는 해결이 어려워 이를 해결하기 위해 DRL을 적용하였다. DRL의 PPO (Proximal Policy Optimization) 방법을 기반으로 최적의 클라이언트 집계 가중치를 계산한다.

서버는 이 값을 사용하여 로컬 모델을 집계하여 전역 모델을 업데이트한다. 클라이언트 선택 시에는 클라이언트 집계 가중치가 특정 임계값 이하인 경우는 제외하는 방식으로 동작한다. BWA는 MNIST, FashionMNIST, CIFAR10 데이터셋의 Non-IID 환경에서 FedAvg 대비 목표 정확도 달성까지 57.78%의 라운드 감소를 달성하였다.

2.2.3 FedPSO

Park 외[13]는 연합학습에서 기존의 FedAvg가 모델 가중치 전체를 서버에 전송하고 데이터 크기만을 기반하는 집계하는 방식에서 발생하는 문제를 해결하기 위해 FedPSO (Federated Particle Swarm Optimization) 알고리즘을 제안한다. 기존 FedAvg는 학습된 로컬 모델의 모든 가중치를 서버로 전송하는데, 이는 네트워크 통신 비용을 증가시키고, 불안정한 네트워크 환경에서 정확도 감소를 유발할 수 있다. 따라서 FedPSO는 클라이언트가 학습된 로컬 모델의 모든 가중치 대신 손실값(loss)만을 서버로 전송한다. 그 다음 분산 환경에서 최적의 해를 찾는 데 적합한 PSO 알고리즘을 사용하는데, 각 클라이언트 모델을 입자로 간주하여 자신의 최고 성능과 전체 군집의 최고 성능을 공유하며 반복적으로 최적값을 찾아가는 방식으로 동작한다. 그리고 모든 클라이언트를 평균내어 집계하는 것이 아닌, loss가 가장 낮은 클라이언트만 식별하여 전역 모델로 업데이트한다. 실험 결과, CIFAR-10 데이터셋에서 FedAvg 대비 9.47% 더 높은 정확도를 달성하였다. 하지만 FedPSO는 통신 효율성을 높이기 위해 집계 과정을 loss가 가장 낮은 클라이언트만을 선택하는 방식을 사용하였다. 이는 Non-IID 환경에서 클라이언트 다양성 반영이 미흡하다는 분명한 한계가 존재한다. 클라이언트 간 데이터 분포의 편차가 존재하는 상황에서 단일 클라이언트의 weight로 글로벌 모델을 대체하는 방식은 전체 모델의 일반화를 저해할 수 있다.

대다수의 기존 연구는 클라이언트 선택 연구와 가중치 집계 연구 중 하나에 집중하는 경향이 있었으며, 클라이언트 선택 및 가중치 집계에 집중하여 클라이언트 모델과 전역 모델을 학습하는 과정에서의 최적화를 고려하지 않았다. 따라서 본 논문에서는 이 세 가지 요소를 모두 고려하는 FedAttnPSO (Federated Learning Attention Particle Swarm Optimization) 알고리즘을 제안한다. 기존 연구와의 차이점 비교를 위해 표 1과 같이 정리

표 1. 기존 연구와 차이점

Table 1. Differences with related works

연구	클라이언트 선택	집계 가중치	최적화
[7, 8, 9]	○	×	×
[10, 11]	×	○	×
[12]	○	○	×
[13]	×	○	○
Ours	○	○	○

하였다. 제안하는 알고리즘은 연합학습 매 라운드마다, loss가 가장 낮은 상위 N개의 선별한 뒤, 각 클라이언트의 로컬 모델을 어텐션 메커니즘을 사용하여 전역 모델에 끼치는 중요도를 계산한다. 이러한 어텐션 중요도를 기반으로 집계 가중치를 부여하여 전역 모델을 업데이트한다. 즉, 전역 모델 업데이트에 중요한 영향을 끼치는 클라이언트에는 높은 가중치를 부여하고, 그렇지 않은 클라이언트에는 낮은 가중치를 부여하여 Non-IID 환경에서도 보다 안정적이고 정밀한 학습이 가능하다. 또한 PSO 알고리즘을 통하여 클라이언트 로컬 모델과 서버 전역 모델 모두 안정적이고 점진적인 모델 학습을 통해 지역 최적해에 빠질 위험을 줄여 전체적인 수렴 안정성과 성능을 동시에 향상시키는 것을 목표로 한다.

III. FedAttnPSO

이 장에서는 본 연구에서 제안하는 FedAttnPSO 알고리즘의 세부적인 동작 방식을 설명한다. FedAttnPSO는 클라이언트와 서버 간의 상호작용을 통해 전역 모델을 점진적으로 최적화하는 연합학습 알고리즘이다. 전체 과정은 클라이언트 측 로컬 최적화와 서버 측 글로벌 최적화라는 두 개의 큰 축으로 구성되며, 이 과정이 여러 라운드에

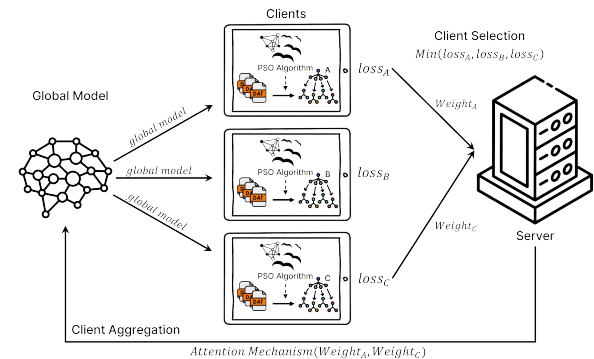


그림 1. 시스템 개요

Fig. 1. System overview

걸쳐 반복된다. 전체적인 개요는 그림 1과 같으며, 각 요소마다 세부적인 동작 절차를 설명한다.

3.1 클라이언트 측 연산

연합학습의 각 라운드가 시작되면 각 클라이언트는 중앙 서버가 가지고 있는 현재의 전역 모델 파라미터를 수신받는다. 각 클라이언트는 수신받은 전역 모델을 기반으로 자신의 로컬 데이터를 사용하여 로컬 모델 학습을 수행한다. FedAttnPSO 알고리즘에서 클라이언트는 모델을 학습할 때 PSO 알고리즘을 통한 최적화 단계를 결합하여 로컬 모델을 업데이트한다. 여기서 각 클라이언트 모델의 가중치는 PSO의 입자(Particle)로 간주되며, 자신의 과거 최고 성능과 서버로부터 받은 전역 모델을 모두 고려하여 최적의 방향으로 이동한다. 클라이언트 모델은 수식 (3) - (4)을 통해 업데이트되며, 알고리즘 1과 같다:

$$v_i^{t+1} = w \cdot v_i^t + c_1 \cdot r_1 \cdot (p_i^{best} - x_i^t) + c_2 \cdot r_2 \cdot (g^{best} - x_i^t) \quad (3)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (4)$$

- x_i^t : 클라이언트 i 의 현재 모델
- v_i^t : 클라이언트 i 의 모델 변화 방향
- p_i^{best} : 클라이언트 i 의 로컬 최적 모델
- g^{best} : 전역 최적 모델
- w : 관성 계수
- c_1, c_2 : 가속 계수
- r_1, r_2 : $[0, 1]$ 범위의 난수

사전에 정의된 로컬 에폭만큼 학습을 마친 클라이언트는 최종적으로 업데이트된 로컬 모델의 손실값을 중앙 서버로 전송한다.

3.2 서버 측 연산

3.2.1 성능 기반 클라이언트 선택

서버는 수신된 손실값을 기준으로 성능이 가장 우수한 상위 N 개의 클라이언트를 선택한다. 손실값이 가장 낮을수록 성능이 좋은 모델이라고 판단하며, 손실값이 가장 낮은 N 개의 클라이언트만을 식별한다. 식별된 클라이언트들에게 파라미터 정보를 요청하고 다음 집계 과정에 포함시킨다. 이 과정을 통해 성능이 낮거나, 학습에 부정적인 영향을 줄 수 있는 클라이언트를 사전에 필터링하여 전역 모델의 품질을 높인다.

Algorithm 1 Client-Side PSO Model Update

Input from Server:

g^{best} : The current global best model received from the server.

Persistent State (for each client i):

p_i^{best} : The personal best model found by client i across all rounds.

v_i : The current velocity of the client's model.

Parameters:

E : Number of local epochs.

w, c_1, c_2 : PSO hyperparameters

$\mathcal{D}_i$: The client's local dataset.

```

1: Begin
2:    $x_i \leftarrow g^{best}$ 
3:   for each local epoch  $j = 1, \dots, E$  do
4:      $r_1, r_2 \sim U(0, 1)$ 
5:      $v_i \leftarrow w \cdot v_i + c_1 \cdot r_1 \cdot (p_i^{best} - x_i) + c_2 \cdot r_2 \cdot (g^{best} - x_i)$ 
6:      $x_i \leftarrow x_i + v_i$ 
7:     Evaluate the loss  $L(x_i)$  on the local dataset  $\mathcal{D}_i$ 
8:     if  $L(x_i) < L(p_i^{best})$  then
9:        $p_i^{best} \leftarrow x_i$ 
10:    end if
11:  end for
12: End

```

알고리즘 1. 클라이언트 측 PSO 알고리즘
Algorithm 1. Client-side PSO algorithm

3.2.2 어텐션 기반 가중치 집계

선택된 상위 N 개의 클라이언트 모델들을 전역 모델로 집계하는 과정에 어텐션 메커니즘을 사용한다. 이는 단순히 평균을 내는 것이 아니라, 각 클라이언트의 성능, 즉 손실값에 기반하여 차등적인 가중치를 부여하는 방식이다. 손실값이 낮은 클라이언트가 더 높은 어텐션 점수를 할당받고, 이에 따라 가중치가 부여된다. 각 클라이언트의 어텐션 점수는 수식 (5)와 같이 해당 클라이언트의 손실값의 역수를 취하여 계산된다. 이때 분모에 더해지는 작은 상수를 손실값이 0에 가까워질 때 발생할 수 있는 수치적 불안정성을 방지하는 역할을 한다. 수식 (6)을 통해 소프트맥스 함수를 통해 모든 클라이언트의 가중치 합이 1이 되도록 정규화한다:

$$s_i = \frac{1}{loss_i + \epsilon}, \quad \epsilon = 1 \times 10^{-8} \quad (5)$$

$$a_i = \frac{e^{s_i}}{\sum_{j=1}^N e^{s_j}} \quad (6)$$

이렇게 계산된 어텐션 가중치를 사용하여 새로운 전역 모델을 생성하는데, 선택된 클라이언트 모델들의 각 파라미터에 해당 가중치를 곱하여 가중합을 구하는 방식으로 이루어진다.-

3.2.3 PSO 기반 전역 모델 업데이트

서버는 수식 (5) - (6)으로 계산된 전역 모델을

Algorithm 2 Server-Side PSO Model Update

Input:
 g_{new}^{best} : The new global model aggregated/selected from clients in the current round.

Persistent Server State:
 x_s : The server's current global model state (position).
 v_s : The current velocity of the server's model.

Parameters:
 S : Number of server-side stabilization steps.
 w : Inertia weight.
 c : Acceleration coefficient.

```

1: Begin
2:   Let the target model  $g^{best} \leftarrow g_{new}^{best}$ 
3:   for each stabilization step  $k = 1, \dots, S$  do
4:      $r \sim U(0, 1)$ 
5:      $v_s \leftarrow w \cdot v_s + c \cdot r \cdot (g^{best} - x_s)$ 
6:      $x_s \leftarrow x_s + v_s$ 
7:   end for
8:   return  $x_s$ 
9: End

```

알고리즘 2. 서버 측 PSO 알고리즘
 Algorithm 2. Server-side PSO algorithm

즉시 다음 라운드의 모델로 채택하는 것이 아니라, 서버 측 PSO 알고리즘을 통해 현재의 전역 모델을 안정적이고 점진적으로 이동시킨다. 서버는 개인 최적해 없이 전역 최적해만을 참조하여 자신의 위치를 업데이트한다. 이는 특정 라운드의 결과가 전역 모델에 급격한 변화를 주는 것을 막고, 전체 학습 과정의 안정성을 확보하는 역할을 할 수 있다. 서버 측 업데이트는 수식 (7) - (8) 및 알고리즘 2와 같다:

$$v_i^{t+1} = w \cdot v_i^t + c \cdot r \cdot (g^{best} - x_i^t) \quad (7)$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (8)$$

- x_i^t : 서버 모델의 현재 상태
- v_i^t : 서버 모델의 변화 방향(속도)
- g : 서버가 기억하는 가장 좋은 모델
- w : 관성 계수
- c : 가속 계수
- r : $[0, 1]$ 범위의 난수

3.3 전체 시스템 요약

지금까지 설명한 클라이언트와 서버의 동작을 통합한 FedAttnPSO의 전체 흐름은 알고리즘 3과 같다. 이처럼 FedAttnPSO는 성능 기반의 클라이언트 선택, 어텐션 메커니즘을 이용한 가중치 집계,

Algorithm 3 FedAttnPSO

Input:
 C : The set of all clients.
 R : The number of total communication rounds.
 E : The number of local epochs for client training.
 N : The number of top clients to select for aggregation.

Output:
 g_{best} (the final trained global model)

```

1: begin
2:   for each round  $i = 1, \dots, R$  do
3:     for each  $C_k \in C$  do
4:        $w_{C_k} \leftarrow g_{best}$ 
5:       for each epoch  $e = 1, \dots, E$  do
6:          $w_{C_k} \leftarrow$  Update model using Client-Side PSO Algorithm
7:       end for
8:        $loss_{C_k} \leftarrow$  Calculate loss for  $w_{C_k}$ 
9:       local  $p_{best} \leftarrow loss_{C_k}$ 
10:      Send  $w_{C_k}$  and  $loss_{C_k}$  to Server
11:    end for
12:     $C_{target} \leftarrow$  Select top N clients
13:     $\alpha \leftarrow$  Calculate attention weights( $C_{target}$ )
14:     $w_{attention} \leftarrow \alpha \cdot C_{target}$ 
15:     $w_{global} \leftarrow$  Update model using Server-Side PSO Algorithm
16:    Update global best( $g_{best}$ ) if  $w_{global}$  is better
17:  end for
18:  return  $g_{best}$ 
19: end

```

알고리즘 3. FedAttnPSO 알고리즘
 Algorithm 3. FedAttnPSO algorithm

클라이언트 및 서버에서의 PSO 기반 점진적 업데이트를 결합하여 Non-IID 환경에서도 강건하고 효율적인 학습을 가능하게 한다.

IV. 실험 및 결과

4.1 데이터셋

본 논문에서 제안하는 FedAttnPSO의 성능을 평가하기 위해 CIFAR-10, CIFAR-100, 그리고 FEMNIST 데이터셋을 사용하였다. 표준 이미지 데이터셋인 CIFAR-10과 CIFAR-100에서는 IID 환경과 Non-IID 환경에서의 성능 차이를 보기 위해 두 환경으로 구분하여 실험을 진행하였다. FEMNIST는 연합학습의 실제 환경을 구현하기 위한 데이터셋으로, Non-IID하게 구분되어 있기에 본 논문에서 제안하는 알고리즘의 성능을 입증하기 위해 사용하였다.

CIFAR-10 및 CIFAR-100은 32x32 컬러 이미지 데이터셋이다. CIFAR-10은 10개의 클래스를 포함하며, CIFAR-100은 100개의 세분화된 클래스를 가져 분류 문제의 복잡도를 높인 데이터셋이다. IID 환경 구성을 한 경우, 전체 학습 데이터를 클라이언트 수에 따라 중복 없이 균등하게 분할하였으며, 이를 통해 각 클라이언트는 모든 클래스에 대해 동일한 분포의 데이터를 보유하도록 설계되었다.

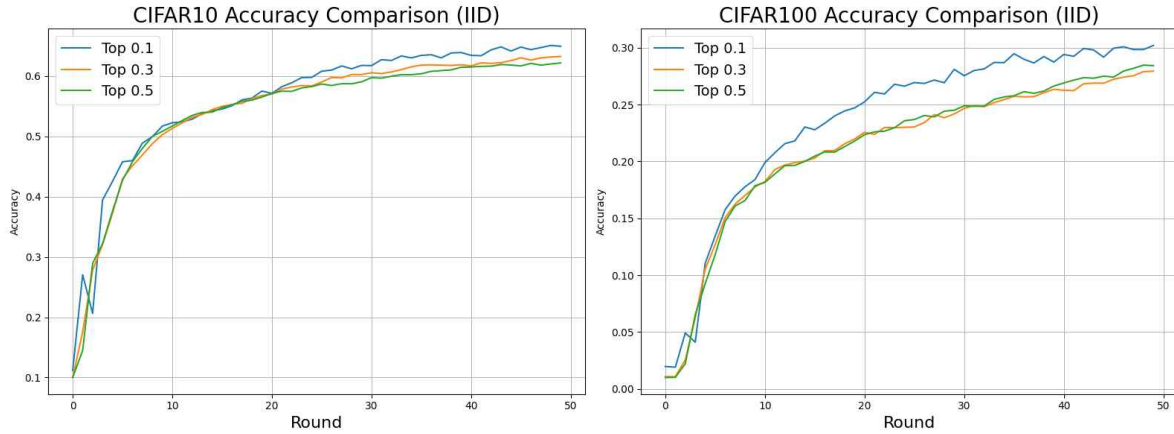


그림 2. IID 환경에서 상위 N 비율 비교
Fig. 2. Top N comparison in IID

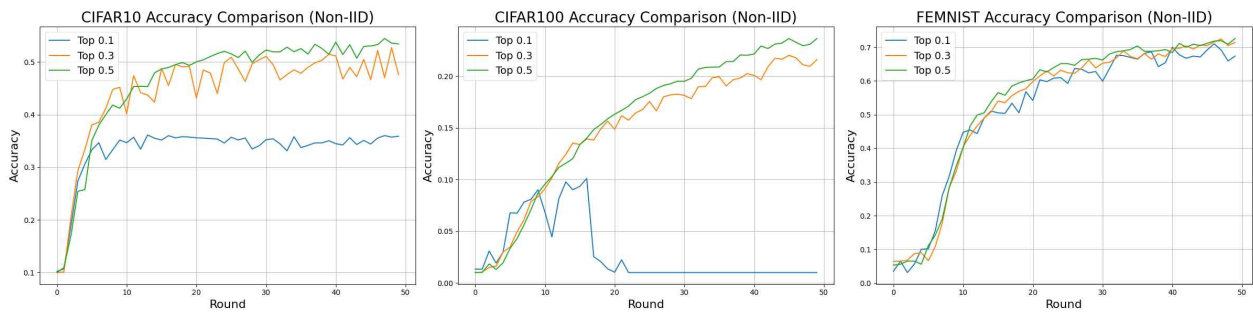


그림 3. Non-IID 환경에서 상위 N 비율 비교
Fig. 3. Top N comparison in Non-IID

반면에 Non-IID 환경을 구성한 경우, 현실 세계의 데이터 불균형을 묘사하기 위해 디리클레 분포 기반의 데이터 분할 방법을 사용하였다. 이 방식은 각 클라이언트가 특정 클래스에 편향되게 데이터를 갖도록 유도하기 때문에 데이터 이질성을 구현할 수 있다.

FEMNIST[14]는 62개의 클래스(숫자 0-9, 영어 대소문자 a-z, A-Z)로 구성된 28x28 흑백 손글씨 이미지 데이터셋이다. 이 데이터셋은 실제 연합학습 환경에서의 실험을 목적으로 구성된 것으로, 실제 작성자별로 데이터가 그룹화되어 있다. 사용자별로 각 클래스가 가지고 있는 데이터가 매우 상이하며, 클래스가 없는 경우도 존재하여 본질적으로 Non-IID 환경을 구성하고 있다. 본 실험에서는 사전처리된 3,500명의 사용자 데이터 중 일부를 클라이언트로 지정하여 사용하였으며, FEMNIST 데이터셋의 특성 상 각 클라이언트는 작성자에 따라 고유한 데이터 양과 클래스 분포를 가지고 있다.

4.2 실험 시나리오

논문에서는 2개의 실험 시나리오(최적 N 선정

표 2. 최적 N 선정 실험을 위한 파라미터

Table 2. Parameters for the optimal N selection experiment

Parameters	Value	
Number of clients	20	
Ratio for selecting clients	0.1, 0.3, 0.5	
Environment	IID	Non-IID
Dataset	CIFAR 10	CIFAR 10
	CIFAR 100	CIFAR 100
		FEMNSIT

표 3. 성능 비교 실험을 위한 파라미터

Table 3. Parameters for the accuracy comparison experiment

Parameter	Value	
Algorithm	FedAvg, FedPSO, FedAttnPSO	
Rounds	10, 30, 50	
Environment	IID	Non-IID
Dataset	CIFAR 10	CIFAR 10
	CIFAR 100	CIFAR 100
		FEMNSIT

실험, 성능 비교 실험)를 통해 FedAttnPSO를 검증하였다. 최적 N 선정 실험에서는 글로벌 모델 통합 과정에 사용하는 상위 N개의 클라이언트를 선택할 때, 가장 최적의 비율을 도출하는 실험을 진행하

였다. 이때, IID 환경과 Non-IID 환경에서의 최적의 N의 비율을 찾기 위해 각각 실험을 진행하였다. 성능 비교 실험에서는 FedAttnPSO 알고리즘과 기존에 널리 사용되는 FedAvg 알고리즘[1], 그리고 PSO를 사용한 선행연구인 FedPSO 알고리즘[13]과의 성능을 비교 및 분석한다. 최적 N 선정 실험을 위한 파라미터는 표 2와 같고, 성능 비교 실험을 위한 파라미터는 표 3과 같다.

4.3 실험 결과

4.3.1 최적 N 선정 실험

최적의 상위 N의 비율을 찾기 위한 첫 번째 실험의 결과는 그림 2 및 그림 3과 같다. 표 2와 같이, 최적 N 선정 실험에서 사용한 클라이언트의 수는 20개이며, 비율을 각각 0.1, 0.3, 0.5로 진행하였다. IID 환경과 Non-IID 환경을 구분하여 실험을 진행했다. IID 환경에서는 CIFAR-10과 CIFAR-100 데이터셋을 사용하였으며, Non-IID 환경에서는 CIFAR-10, CIFAR-100, 그리고 FEMNIST 데이터셋 모두를 사용하여 진행하였다. 그림 2와 같이 IID 환경에서의 실험 결과는 N의 비율을 0.1로 했을 때, CIFAR-10에서 65%, CIFAR-100에서 30%로 가장 높은 성능을 달성하였다. 이는 데이터가 모두 균일하게 분포된 환경에서는 소수의 클라이언트만으로도 전체 데이터의 대표성을 확보할 수 있기 때문에 가장 낮은 클라이언트 비율을 선택할 때 성능이 높은 것으로 해석된다.

Non-IID 환경에서 최적의 N의 비율은 그림 3에서 볼 수 있듯이 0.5이다. CIFAR-10은 53%, CIFAR-100은 24%, FEMNIST는 73% 성능을 달성하였다. 본 실험 결과에서 CIFAR-100 데이터셋의 경우, 100개의 클래스로 세분화되어 분류의 복잡성이 증가되었을 때 클라이언트 선택 비율이 낮은 경우 이에 수렴하지 못하는 것을 볼 수 있다. 이를 바탕으로 IID 환경과 다르게 Non-IID 환경에서는 개별 클라이언트가 가지는 데이터 편향성이 크기 때문에 보다 다양한 클라이언트를 글로벌 모델에 통합시켜야 전체 데이터 분포의 다양성을 반영할 수 있기에 가장 높은 비율인 0.5를 선택한 경우 성능이 높은 것으로 해석된다.

4.3.2 성능 비교 실험 결과

최적 N 선정 실험 결과를 바탕으로, 성능 비교 실험에서 상위 N의 비율을 IID 환경과 Non-IID

표 4. CIFAR10 실험 결과

Table 4. CIFAR10 test accuracy by epoch

Environment	Algorithm	Accuracy (%)		
		10 Rounds	30 Rounds	50 Rounds
IID	FedAvg	38.9	57.3	57.9
	FedPSO	53.1	56.1	54.8
	FedAttnPSO	<u>54.5</u>	<u>63.4</u>	<u>64.9</u>
Non-IID	FedAvg	33.4	52.1	53.2
	FedPSO	29.2	32.2	32.7
	FedAttnPSO	<u>45.4</u>	<u>56.5</u>	<u>58.0</u>

표 5. CIFAR100 실험 결과

Table 5. CIFAR100 test accuracy by epoch

Environment	Algorithm	Accuracy (%)		
		10 Rounds	30 Rounds	50 Rounds
IID	FedAvg	18.0	23.6	24.3
	FedPSO	16.3	22.5	20.7
	FedAttnPSO	<u>21.4</u>	<u>27</u>	<u>29.8</u>
Non-IID	FedAvg	14.2	22.9	24.1
	FedPSO	Nan	Nan	Nan
	FedAttnPSO	<u>14.7</u>	<u>24.3</u>	<u>27.4</u>

표 6. FEMNIST 실험 결과

Table 6. FEMNIST test accuracy by round

Environment	Algorithm	Accuracy (%)		
		20 clients	30 clients	50 clients
Non-IID	FedAvg	66.1	65.4	65.1
	FedPSO	30.7	38.4	35.3
	FedAttnPSO	<u>72.6</u>	<u>71.3</u>	<u>69.7</u>

환경에서 각각 0.1, 0.5로 선택하였다. 먼저 CIFAR-10 및 CIFAR-100 데이터셋의 경우, 전체 데이터의 양이 많지 않아 클라이언트의 개수를 20으로 고정하고, 전체 라운드의 수를 증가시키며 실험을 진행하였다. 라운드별 자세한 수치는 표 4, 5과 같다. CIFAR-10의 경우, IID 환경에서는 제안한 알고리즘이 FedAvg 대비 최대 15.6%, FedPSO 대비 최대 10.1% 높은 정확도를 달성하였고, Non-IID 환경에서는 FedAvg 대비 최대 12%, FedPSO 대비 최대 25.3% 향상되어 제안하는 알고리즘의 타당성을 입증하였다. 다음으로 CIFAR-100의 경우, IID 환경에서 제안하는 알고리즘이 FedAvg 대비 최대 5.5%, FedPSO 대비 최대 9.1% 높은 정확도를 달성하였으며, Non-IID 환경에서는 FedAvg 대비 최대 3.3% 정확도 향상을 달성하였다. FedPSO의 경우는 분류의 복잡성이 높은 CIFAR-100 데이터셋의 경우 Non-IID 환경에서는 수렴하지 못하는 것에 비해 제안하는 알고리즘은 최대 27.4%까지 달성하는 것을 보였다.

앞선 두 데이터셋과 다르게 FEMNIST의 경우, 실제 연합학습 환경 실험을 위한 데이터셋으로 작성

자별로 데이터가 Non-IID 분포를 따라 구성되어 있다. 전체 데이터는 작성자 3,500명으로 구성되어 있어, 본 실험에서는 전체 라운드 수를 고정하고 클라이언트의 개수를 20, 30, 50으로 증가시켜 실험을 진행하였다. 클라이언트 수 별 자세한 수치는 표 6와 같으며, 제안하는 알고리즘이 FedAvg 대비 최대 6.5%, FedPSO 대비 최대 41.9% 향상되었다.

세 가지 데이터셋의 실험 결과, 특히 FedPSO의 경우 Non-IID 환경에서의 성능 저하가 크게 나타났는데, 이는 통신비용 감소를 위해 매 라운드마다 단일 클라이언트의 모델만을 글로벌 모델에 반영하기 때문에 PSO 알고리즘을 사용하여 모델 최적화를 진행함에도 불구하고 Non-IID 환경에서는 제대로 학습하지 못하는 것을 알 수 있다. 반면 본 연구에서 제안하는 FedAttnPSO 알고리즘의 경우, 상위 N개의 클라이언트 모델을 어텐션 매커니즘을 통해 중요도 기반으로 가중 집계하고, 서버 측 PSO 최적화까지 결합하여 기존 연구인 FedPSO의 한계를 해결하며, FedAvg보다도 빠르고 안정적인 수렴을 보여 제안하는 알고리즘의 견고성을 입증하였다.

V. 결 론

본 논문에서는 연합학습의 한계인 Non-IID 환경에서의 성능 저하를 극복하기 위해 성능 기반 클라이언트 선택, 어텐션 메커니즘 기반 가중치 집계, PSO 최적화를 결합한 새로운 알고리즘인 FedAttnPSO를 제안하였다. 기존에 널리 사용되는 FedAvg 알고리즘의 무작위 클라이언트 선택과 단순 평균 집계를 개선하기 위해 각 클라이언트의 손실값을 기준으로 상위 N개를 선택하는 방식과, 어텐션 메커니즘을 사용하여 가중치를 차등 부여하는 방식, 더 나아가 학습 과정에 PSO 알고리즘을 결합하여 보다 안정적이고 점진적인 모델 업데이트를 통해 견고한 알고리즘을 제안하였다.

세 가지 데이터셋(CIFAR-10, CIFAR-100, FEMNIST)을 사용한 성능 비교 실험을 통해 제안하는 알고리즘의 우수성을 정량적으로 입증하였다. 특히 현실적인 Non-IID 환경을 구현한 FEMNIST 데이터셋을 사용한 실험에서 FedAttnPSO는 FedAvg 대비 최대 6.5%, FedPSO 대비 최대 41.9%의 정확도 향상을 달성했다. 또한 분류 복잡성이 높은 CIFAR-100 Non-IID 환경에서 FedPSO가 수렴에 실패한 반면, 제안하는 FedAttnPSO 알고리즘은 안정적인

로 학습을 진행하며 최대 27.4%의 정확도를 달성하였다. 이는 다양한 클라이언트를 고려하고, 안정적으로 통합하는 방식의 견고성을 증명하여 충분한 실험 가능성을 입증하였다.

본 연구에서는 이미지 데이터셋을 중심으로 실험을 진행하였으나 향후 연구에서는 보다 다양한 유형의 데이터셋을 활용하여 여러 환경에서의 적용 가능성을 검토하고, 제안하는 알고리즘의 일반성과 타당성을 입증하고자 한다. 아울러, 현재의 손실값 기반 클라이언트 선택 방식에서 나아가 강화학습 기반의 선택 정책을 도입함으로써 클라이언트의 신뢰도, 데이터 품질, 이전 라운드에서의 기여도 등을 종합적으로 고려하는 방향으로 확장하고자 한다. 이를 통해 보다 지능적이고 동적인 클라이언트 관리를 실현하고, 연합학습의 성능을 더욱 향상시키고자 한다. 또한 클라이언트 모델의 집계 과정에 PSO 알고리즘과 어텐션 메커니즘을 적용함에 따라 기존 FedAvg 알고리즘 대비 계산 복잡도가 증가할 수 있으므로, 정량적 분석을 통해 성능 향상과 복잡도 증가 간의 균형을 평가하고 실제 적용 가능성을 검토할 계획이다. 마지막으로, 모델 성능의 다각적 평가를 위해 정확도 뿐만 아니라 정밀도, 재현율, F1 Score 등의 지표를 활용한 실험을 추가로 수행함으로써 제안 기법의 전반적인 효과성을 검증할 것이다.

References

- [1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguerre y Arcas, "Communication-efficient learning of deep networks from decentralized data," *Proceedings of 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1273 - 1282, Florida, USA, April 2017.
- [2] Y. J. Cho, J. Wang, and G. Joshi, "Towards understanding biased client selection in federated learning," *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 10351 - 10375, Valencia, Spain, March 2022.
- [3] D. Zeng, Z. Xu, S. Liu, Y. Pan, Q. Wang, and X. Tang, "On the power of adaptive weighted aggregation in heterogeneous federated learning and beyond," *Proceedings of the 28th International*

- Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1 - 22, Phuket, Thailand, May 2025.
- [4] Z. Li, T. Lin, X. Shang, and C. Wu, "Revisiting weighted aggregation in federated learning with neural networks," *Proceedings of the 40th International Conference on Machine Learning*, pp. 19767 - 19788, Honolulu, Hawaii, USA, July 2023.
- [5] N. Kang, Y. Lim, and J. Im, "Optimizing federated learning: addressing key challenges in real-world applications," *IEEE Internet of Things Journal*, vol. 12, no. 12, pp. 19065 - 19089, April 2025.
- [6] J. Kennedy, and R. Eberhart, "Particle Swarm Optimization," *Proceedings of IEEE International Conference on Neural Networks*, WA, Australia, pp. 1942 - 1948, December 1995.
- [7] S. Chen, O. Tavalai, M. H. Hambali, S. M. Zandavi, H. Haddadi, N. Lane, S. Guo, and A. Y. Zomaya, "Optimization of federated learning's client selection for non-iid data based on grey relational analysis," *arXiv preprint arXiv:2310.08147*, October 2023.
- [8] T. M. Nguyen, K. L. Poh, S. L. Chong, and J. H. Lee, "FedDSS: a data-similarity approach for client selection in horizontal federated learning," *International Journal of Medical Informatics*, vol. 192, pp. 1 - 9, October 2024.
- [9] W. Marfo, D. K. Tosh, and S. V. Moore, "Efficient client selection in federated learning," *arXiv preprint arXiv:2502.00036*, January 2025.
- [10] C. Shi, H. Zhao, B. Zhang, M. Zhou, D. Guo, and Y. Chang, "FedAWA: adaptive optimization of aggregation weights in federated learning using client vectors," *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30651 - 30660, Nashville, Tennessee, USA, June 2025.
- [11] R. Ye, M. Xu, J. Wang, C. Xu, S. Chen, and Y. Wang, "FedDisco: federated learning with discrepancy-aware collaboration," *Proceedings of the 40th International Conference on Machine Learning*, pp. 39879 - 39902, Honolulu, Hawaii, USA, July 2023.
- [12] J. S. Bang, S. P. Woo, and J. H. Lee, "DRL-empowered joint batch size and weighted aggregation adjustment mechanism for federated learning on non-iid data," *ICT Express*, vol. 10,

no. 4, pp. 863 - 780, August 2024.

- [13] S. H. Park, Y. R. Suh, and J. W. Lee, "FedPSO: federated learning using particle swarm optimization to reduce communication costs," *Sensors*, vol. 21, no. 2, pp. 1 - 13, January 2021.
- [14] S. Caldas, S. M. K. Duddu, P. Wu, T. Li, J. Konecny, H. B. McMahan, V. Smith, and A. Talwalkar, "Leaf: a benchmark for federated settings," *arXiv preprint arXiv:1812.01097*, December 2018.

장 선 영 (Sun-young Jang)



2024년 2월 : 강원대학교 컴퓨터공학과 졸업
2024년 2월 - 현재 : 강원대학교 데이터사이언스학과 석사 과정
<관심분야> 연합학습, 네트워크 보안

김 종 욱 (Jong-Wouk Kim)



2019년 2월 : 강원대학교 컴퓨터과학 졸업
2021년 2월 : 강원대학교 컴퓨터과학 석사
2021년 3월 - 현재 : 강원대학교 컴퓨터과학 박사과정
<관심분야> 네트워크 보안, 악성코드 분석, 딥러닝

최 미 정 (Mi-Jung Choi)



1998년 2월 : 이화여자대학교 공학사 졸업
2000년 2월 : 포항공과대학교 공학석사
2004년 2월 : 포항공과대학교 공학박사
2004년 - 2005년 : 프랑스 INRIA

연구소 박사 후 연구원
2005년 - 2006년 : 캐나다 워터루대학 박사 후 연구원
2008년 - 현재 : 강원대학교 컴퓨터학부 컴퓨터공학 전공 교수
<관심분야> 네트워크 관리, 정보보안, 네트워크 보안

그래프 신경망 기반 초기 해집단 평가 모델을 이용한 동적 네트워크 라우팅 최적화

이 도 영*, 김 태 연*

Dynamic network routing optimization using a GNN-based initial population evaluation model

Doyoung Lee*, Taeyeon Kim*

요 약

차세대 6G 네트워크에 대한 기술적인 논의가 본격화되고 있는 오늘날, 다양한 요구사항을 갖는 서비스를 지원하기 위한 동적이고 유연한 네트워크 운용 및 제어 기술은 여전히 필수 요건으로 거론되고 있다. 특히, 저궤도 위성이나 비행체와 같은 새로운 통신 개체가 일반화되는 등의 구조적인 변화로 인해 네트워크는 보다 동적인 환경과 엄격한 제약 조건을 요구하고 있다. 이와 같은 변화는 기존 라우팅 최적화를 복잡한 서비스 특성과 네트워크 상황을 고려해야 하는 어려운 문제로 만들고 있다. 한편, 메타 휴리스틱 기법 중 하나인 유전 알고리즘은 라우팅 최적화에 효과적으로 활용되어 온 방법이지만, 초기 해집단의 무작위성으로 인해 수렴 안정성과 최적해 품질이 저하되는 한계 또한 존재한다. 본 연구에서는 네트워크 상태를 그래프 신경망을 통해 학습하여 초기 해집단의 품질을 예측한 후, 고품질 해집단을 기반으로 유전 알고리즘을 수행하는 라우팅 최적화 기법을 제안한다. 제안 방법에서는 통신 도메인과 연산 도메인의 운용 비용을 함께 고려하여 주어진 네트워크 환경에서 서비스 요구사항을 만족하는 최적 경로를 도출한다.

Key Words : 6G, Routing optimization, Genetic algorithm, Graph neural network, Meta-heuristic algorithm

ABSTRACT

As discussions on 6G networks progress, dynamic and flexible network control remains essential to support diverse service requirements. The emergence of new communication entities such as LEO satellites and aerial vehicles introduces dynamic environments and strict constraints, increasing the complexity of routing optimization. Although Genetic Algorithms (GAs) have been widely used for such optimization, their performance can suffer due to randomness in the initial population. This study proposes a routing optimization method that uses a Graph Neural Network (GNN) to learn the network state and predict high-quality initial solutions. Subsequently, the method integrates with a GA to derive optimal routing paths while considering both communication and computation costs.

※ 본 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임
(No. RS-2024-00435652, 6GARROW: 6G Ai-native 통합 RAN-Core 네트워크)

◆ First Author : Electronics and Telecommunications Research Institute, dylee90@etri.re.kr

◦ Corresponding Author : Electronics and Telecommunications Research Institute, dylee90@etri.re.kr

* Electronics and Telecommunications Research Institute, tykim@etri.re.kr

논문번호 : KNOM2025-01-05, Received July 14, 2025; Revised August 08, 2025; Accepted August 25, 2025

I. 서 론

5G 시대를 넘어 6G 네트워크 기술에 대한 논의가 본격화되고 있는 오늘날에 이르기까지, 네트워크 기술의 비약적인 발전은 이기종의 기기들이 네트워크를 통해 상호 연결되어 통신할 수 있는 토대를 마련해왔다. 특히, 소프트웨어 정의 네트워킹 및 네트워크 기술 가상화로 대표되는 네트워크 소프트웨어화 (Network Softwarization) 기술은 다양한 요구사항을 갖는 서비스를 운용하기 위한 환경을 동적으로 구성할 수 있는 핵심 원동력이 되어왔다 [1].

다가오는 차세대 통신 네트워크에서는 이러한 추세를 따라 5G보다 더욱 엄격한 서비스 성능 요구사항을 충족하는 것 뿐만 아니라 보다 확장된 서비스 용례들 (Use-cases)을 제공하는 것이 중요시 될 것으로 예상된다. 예시로, 기존 지상 통신의 범주를 벗어나서 저궤도 위성, 비행체 등이 통신 중계 개체로 포함되는 것이 일반화되고 [2, 3], 인공지능 기술이 보편화됨에 따라 인공지능 모델 운용 및 이를 기반으로 한 서비스를 지원하기 위한 맞춤형 네트워크 구성이 필수 선행조건이 되고 있다.

한정된 통신 및 연산 자원을 고려하여 각 서비스 특성에 맞는 맞춤형 네트워크를 구성하기 위해서는 유연하고 동적인 네트워크 설정이 선행되어야 한다. 이는 한정된 자원을 활용하여 복잡한 요구사항을 갖는 각 서비스를 위한 네트워크 운용 및 제어를 다루는 최적화 문제들로 정의될 수 있다. 그 중에서도 서비스 트래픽 라우팅 문제는 경로 비용, 에너지 소모량, 서비스 성능 요구사항 충족 여부 등을 고려하여 최적의 트래픽 경로를 결정하는 것이다.

복잡하고 다양화되고 있는 네트워크와 서비스 특성을 고려한 최적 라우팅을 수행하는 것은 더욱 도전적인 목표가 되고 있다. 특히, 여러 도메인으로 구성된 네트워크 환경에서 파생되는 중단 간 서비스 최적화에 대한 어려움과 성능 요구사항을 위해 응용 서비스가 배치되어야 하는 위치를 결정하는 요소 등 라우팅 문제에서 다루는 조건이 증가하고 있다. 이처럼 복잡해지는 라우팅 문제 해결을 위해 인공지능 기술을 적용한 성공적인 시도가 있었지만 [4], 모델 학습에 필요한 데이터 확보와 연산 비용은 여전히 해결해야 할 과제로 남아있다.

한편, 인공지능 기반 라우팅 방법 외에도 근사 최적해를 구하는데 활용되는 메타 휴리스틱 기법들을 라우팅에 접목한 사례도 존재한다. 그에 대한 예로,

진화 알고리즘을 대표하는 유전 알고리즘 (GA, Genetic Algorithm) [5]은 트래픽 라우팅에 요구되는 제약 조건을 만족하기 위한 결정 변수의 조합 최적화 문제에 주로 활용된다. 하지만, 유전 알고리즘은 탐색적 기법이라는 특징으로 인해 전역 최적해로 수렴한다는 보장이 어렵다는 한계가 존재한다 [6]. 또한, 초기 해집단 (Initial population)을 대상으로 각 세대에서 교차 및 돌연변이 연산을 통해 최적해를 찾기 때문에 해집단의 구성에 따라 수렴 시간과 최종 탐색된 해의 품질이 가변적일 수 있다. 이처럼 유전 알고리즘은 초기 해집단 구성이 중요한 요소이지만, 현재 대부분의 유전 알고리즘 기반 라우팅에서는 초기 해집단에 활용되는 각 해를 무작위로 생성하는 방법이 활용되고 있으며 [7, 8], 이는 효율적인 유전 알고리즘 수행을 저해하는 요소가 된다.

본 논문에서는 상기 문제의 해결 방안으로서, 주어진 네트워크 환경에서 유전 알고리즘 기반 최적 라우팅 방법을 수행하기 위한 초기 해집단 평가 방법을 제안한다. 제안하는 방법에서는 네트워크를 그래프 형태로 학습하는 그래프 신경망 (GNN, Graph Neural Network)을 활용하고, 각 라우팅 경로를 표현하는 염색체들의 집합을 초기 해집단으로 정의하여 품질을 예측한다. 유전 알고리즘 기반 라우팅은 그래프 신경망을 통해 높은 품질 값으로 예측된 초기 해집단을 기반으로 라우팅 경로를 결정한다.

본 논문의 나머지 내용은 다음과 같이 구성된다. 2장에서는 서비스 트래픽을 대상으로 하는 라우팅 기법과 관련된 연구 동향에 대해 살펴본다. 3장에서는 서비스 트래픽 라우팅에 대한 시스템 모델을 정의하고, 이어지는 4장에서는 제안하는 그래프 신경망을 활용한 유전 알고리즘 기반 라우팅 방법에 대해 서술한다. 5장에서는 구현된 라우팅 기능에 대한 성능 평가를 수행하며, 마지막인 6장에서는 논문 내용을 정리하고 향후 연구 방향에 대해 논의한다.

II. 연구 배경 및 관련 연구

네트워크 분야에서 라우팅은 트래픽의 출발지로부터 목적지까지의 경로를 결정하는 것이며, 이 때 각 경로는 트래픽의 성능 요구사항을 만족함과 동시에 운용 비용이 최소화되어야 한다 [9]. 일반적으로 트래픽 라우팅 문제에서는 네트워크 성능 지표 중 대역폭, 지연 시간, 가용성을 고려한 최단 경로를 결정하는 것을 목표로 한다 [10, 11].

동적으로 변화하는 네트워크 환경에서도 최적의 라우팅을 수행하기 위한 해결책으로 인공지능 기술이 적용되고 있다. 그에 대한 예로, 선행 연구 [12]에서는 라우팅에 활용할 수 있도록 기계학습을 통해 미래에 발생할 트래픽을 예측하는 방법을 제안하였으며, 선행 연구 [13]에서는 암호화된 트래픽을 기계학습을 통해 분류하는 기술을 제시하였다. 그 외에도 네트워크를 그래프로 표현할 수 있다는 특성을 활용하여 네트워크 상태를 그래프 신경망을 통해 학습하고, 강화학습과 결합하여 트래픽 경로를 결정하는 연구도 존재한다 [14].

네트워크를 그래프 형태로 표현하여 최적 라우팅을 수행하기 위해서는 토폴로지 뿐만 아니라 동적으로 변화하는 네트워크 상태를 반영한 그래프 정보가 요구된다. 따라서, 선행 연구 [15, 16]에서는 트래픽 정보와 네트워크 기기 종류를 고려한 그래프 임베딩 방법을 제안하고, 이를 강화학습에 활용하는 기술을 연구하였다. 또한, 선행 연구 [17]에서는 대규모 언어 모델 (LLM, Large Language Model)을 활용하여 네트워크 그래프를 생성하는 방법을 제안하였다.

트래픽 라우팅은 최적화 문제로 정의될 수 있는 만큼, 인공지능 기술 외에도 메타 휴리스틱한 방법으로 해결하는 연구도 수행 되어왔다. 선행 연구 [18]에서는 SDN 환경에서 부하 분산을 목적으로 이산 입자 군집 (Discrete particle swarm)을 활용하여 트래픽 경로를 설정하는 방법이 연구되었다. 또한, 유전 알고리즘을 다중 경로 라우팅 [19]과 네트워크 에너지 효율 개선을 위한 트래픽 경로 설정 [20]에 활용한 기술도 제안되었다.

한편, 트래픽 라우팅 문제는 최근 기후 문제 등과 맞물려 성능 요건 뿐 아니라 환경적인 요인도 고려하는 것이 요구되고 있다. 예를 들어, 탄소 배출량을 줄이는 것이 중요시되면서 라우팅을 통해 네트워크에서 발생하는 탄소 배출량을 절감하기 위한 연구가 수행되었다 [21]. 또한, 탄소 배출량은 전력 소모량과 비례 관계가 있기 때문에 라우팅 수행 과정에서 네트워크에서의 전력 소모와 함께 해당 트래픽이 처리되는 목적지의 연산 환경을 고려한 라우팅 방법도 연구된 바 있다 [22].

상기 언급된 것과 같이 네트워크에서의 트래픽 라우팅 문제는 새로운 제약 조건이 추가되며 복잡한 형태로 진화하고 있다. 따라서, 가변적인 네트워크 환경에서 각 서비스 요구사항에 맞는 최적의 라우팅을 효율적으로 결정하는 방법이 필요하다. 이를

위해 본 논문에서는 그래프 신경망과 유전 알고리즘을 활용한 라우팅 기법을 제안한다.

Ⅲ. 시스템 디자인

3.1. 시스템 모델

소프트웨어 정의 네트워크 구조에서는 중앙 집중화된 제어 평면을 통해 전체 네트워크 상태를 조회하여 동적으로 트래픽 라우팅 경로의 결정이 가능하다. 차세대 통신 네트워크에서도 유연하고 동적인 네트워크 설정이 필수적이므로, 제어 평면을 통한 라우팅 설정이 이루어질 것으로 예상된다. 그림 1은 제어 평면을 통한 서비스 트래픽 라우팅이 이루어지는 개념을 보인다.

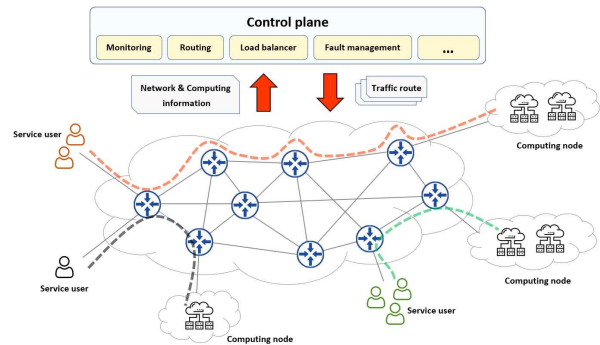


그림 1. 서비스 트래픽 라우팅 개념

Fig. 1. Concept of service traffic routing

본 논문에서 다루는 서비스 트래픽 라우팅 문제는 관련 연구 [22]를 기반으로, 트래픽 전달을 수행하는 네트워크 도메인 뿐만 아니라 해당 트래픽이 처리되는 연산 도메인을 함께 고려한다. 즉, 네트워크는 네트워크 노드 N^d 와 연산 노드 N^c 를 포함하는 노드들의 집합인 N 과 이들을 연결하는 링크 L 로 표현되는 그래프인 $G = (N, L)$ 로 정의된다. 라우팅은 각 서비스 트래픽의 요구사항인 $r \in R$ 이 존재할 때, 제어 평면을 통해 목적지인 연산 노드까지의 경로를 결정하는 것이다.

3.2 서비스 트래픽 라우팅 모델

서비스 트래픽 라우팅 문제는 서비스에 대한 트래픽 요구사항이 발생했을 때, 라우팅을 통해 결정된 경로를 따라 연산 노드로 전달된 트래픽이 처리되는데 소모된 운용 비용을 최소화하는 문제로 정의

할 수 있다. 마찬가지로, 본 논문에서는 관련 연구 [22]에서 정의된 최적화 수식을 기반으로 서비스 트래픽 라우팅 문제를 다룬다. 해당 최적화 문제는 수식 (1)과 같이 운용 비용 최소화에 대한 목적 함수로 정의된다.

$$\min \sum_{r \in R} \left(\sum_{l \in L} x_{rl} e_l + \sum_{n \in N^C} y_{rn} m_n \right) \quad (1)$$

$$\text{s.t.}, \quad \sum_{r \in R} x_{rl} t_r \leq b_l, \quad \forall l \in L \quad (2)$$

$$\sum_{l \in L} x_{rl} d_l \leq g_r, \quad \forall r \in R \quad (3)$$

$$\sum_{l \in L} x_{rl} \log a_l \geq \log k_r, \quad \forall r \in R \quad (4)$$

$$\sum_{l \in L} x_{rl} \leq h_r, \quad \forall r \in R \quad (5)$$

수식 (1)은 트래픽 요구사항 $r \in R$ 으로 인해 결정된 라우팅 경로에 포함되는 모든 링크 $l \in L$ 의 운용 비용 e_l 과 연산 노드 $n \in N^C$ 의 운용 비용 m_n 을 최소화하는 것을 목적으로 한다. 이 때, x_{rl} 와 y_{rn} 은 각각 해당 링크와 연산 노드가 라우팅 경로에 포함되는지 여부를 나타내는 이진 변수이며 다음의 수식 (6)과 같이 정의된다.

$$x_{rl}, y_{rn} \in \{0, 1\}, \quad \forall r \in R, l \in L, n \in N^C \quad (6)$$

수식 (2)~(5)는 목적 함수와 관련하여 라우팅 과정에서 충족해야 하는 제약 조건을 명시한 것이다. 수식 (2)는 네트워크 전송 용량을 나타내는 조건으로, 각 요청 r 이 요구하는 전송 용량 t_r 의 합이 해당 링크에서 제공하는 최대 전송 용량 b_l 보다 크지 않다는 것을 보장한다. 다음으로 수식 (3)은 요청 r 에 대한 트래픽 경로를 구성하는 각 링크 l 의 지연 시간 d_l 의 합이 해당 요청이 허용하는 최대 지연인 g_r 보다 작아야 하는 것을 명시한다. 이 외에도 수식 (4)는 경로 내 링크 l 의 가용성 a_l 이 요청 r 이 요구하는 최소 가용성 k_r 보다 커야 한다는 제약 조건을 나타내며, 수식 (5)는 각 경로의 최대 홉 수를 제한하는 조건을 명시한다.

그 외에도 최적 라우팅을 위한 목적 함수는 다음과 같이 트래픽이 처리되는 연산 노드와 관련된 제약 조건도 존재한다.

$$\sum_{n \in N^C} y_{rn} = 1, \quad \forall r \in R \quad (7)$$

$$\sum_{n \in (N - N^C)} y_{rn} = 0, \quad \forall r \in R \quad (8)$$

$$\sum_{r \in R} y_{rn} f_{rj} \leq c_{nj}, \quad \forall n \in N^C, j \in J \quad (9)$$

수식 (7)은 단일 요청 r 이 하나의 연산 노드에서 처리되어야 함을 나타내는 조건이며, 수식 (8)은 트래픽이 네트워크 노드가 아닌 연산 노드에서만 처리될 수 있음을 명시한다. 마지막으로, 각 요청 r 은 해당 요청에 대한 트래픽을 처리하는데 요구하는 연산 자원량이 존재한다. 연산 노드에서 고려하는 자원의 종류로는 CPU, GPU, 메모리가 있으며, 이는 $J = \{\text{CPU, GPU, RAM}\}$ 으로 표현된다. 즉, 수식 (9)는 트래픽이 자원 j 에 대해서 요구하는 자원량 f_{rj} 보다 해당 자원에 대한 연산 노드 n 의 가용 자원량 c_{nj} 이 충분해야 한다는 조건을 나타낸다.

IV. 제안 방법

앞서 서술한 서비스 트래픽 라우팅 문제는 제약 조건이 있는 최단 경로 문제 (CSP, Constrained Shortest Path)로 구분될 수 있다. 해당 문제는 일반적으로 NP-Hard이기 때문에 다항 시간 내 최적해를 찾는 것이 어렵다 [23]. 따라서, 본 논문에서는 네트워크 노드와 연산 노드에서의 운용 비용을 모두 고려한 최적 라우팅을 수행하기 위해 유전 알고리즘을 활용한다.

4.1 유전 알고리즘 기반 라우팅 방법

유전 알고리즘 기반 라우팅에서 해집단을 구성하는 각 염색체 (Chromosome)는 트래픽 경로를 나타낸다. 단일 염색체 내에서 항상 첫 번째 유전자 (Gene)는 트래픽의 출발점이 되는 네트워크 노드를 나타내며, 마지막 유전자는 트래픽이 처리되는 연산 노드를 나타낸다. 또한, 수식 (5)에 의해 라우팅 경로는 최대 홉 수 제약 조건이 있기 때문에 염색체 길이도 제한된다. 예를 들어, 최대 홉 수가 h 일 경우, 염색체가 가질 수 있는 유전자 수는 최대 $h+1$ 개가 된다.

제안하는 방법에서는 유전 알고리즘을 활용하여 라우팅을 수행하기 위해 서비스 트래픽 요청이 발생할 때마다 해당 요청에 대한 라우팅 경로를 나타내는 염색체들로 초기 해집단을 생성해야 한다. 이를 위해 일반적으로 활용하는 방식인 너비 우선 탐색 (BFS, Breadth-First Search) 방법을 통해 정해진 홉 수 한도 내에서 가능한 경로를 탐색한 후, 초기 해집단 크기에 맞는 개수를 임의로 선택하여 구성한다.

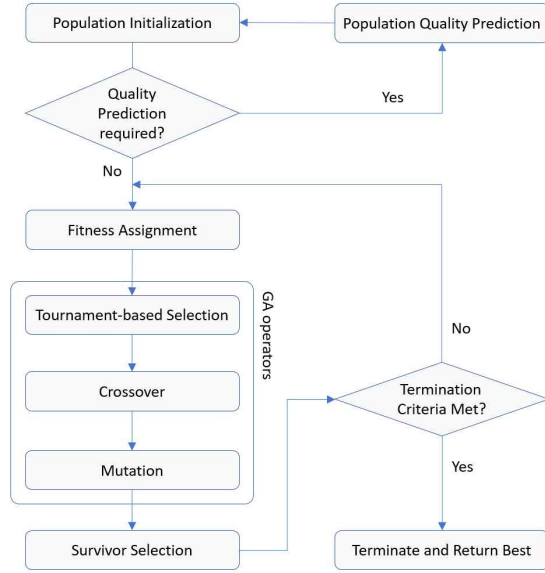


그림 2. 유전 알고리즘 기반 라우팅 방법 순서도

Fig. 2. Sequence chart for GA-based routing

다음으로는 각 세대를 거치면서 토너먼트 선택 기법을 통해 부모 염색체 쌍을 고른 후, 확률적으로 교차 (Crossover) 및 돌연변이 (Mutation) 연산을 순차적으로 수행한다. 해당 연산들로 인해 단절된 경로 등 유효하지 않은 결과물이 생성되는 것을 방지하기 위해 교차에서는 두 부모 염색체 간 공통으로 존재하는 노드 중 하나를 임의로 선택하여 교차 지점으로 사용하고, 돌연변이에서는 변경되는 노드를 기준으로 이전 노드와 이후 노드의 공통 이웃 노드들 중 하나로 변경한다. 유전 알고리즘 기반 라우팅에서는 이와 같은 과정을 종결 조건에 도달할 때까지 반복하며 적합도 함수를 통해 해의 품질을 측정하여 최적해를 산출한다. 이 때, 제안하는 방법에서는 종결 조건으로 일정 세대 수 (e.g., 5세대) 동안 최적해가 변하지 않거나 미리 정의된 종결 세대수에 이르면 종료되는 것으로 설정하였다. 그림 2는 유전 알고리즘 기반 라우팅 방법의 수행 순서도를 보인다.

$$F = C_{total} + P_{total} \quad (10)$$

$$C_{total} = C_{links} + C_{nodes} \quad (11)$$

$$P_{total} = \sum_{z \in Z} P_z^{nw} + \sum_{j \in J} P_j^{cp} \quad (12)$$

$$P_j = \max(0, f_{rj} - c_{nj}) \times w_j \quad (13)$$

제안하는 유전 알고리즘 기반 라우팅 방법에서는 서비스 요구사항을 만족하는 것과 동시에 최소 운

용 비용을 가지는 경로를 찾는 것을 목표로 한다. 이를 위해 특정 서비스 요청을 대상으로 하는 경로를 나타내는 각 해의 적합도는 수식 (10)을 통해 계산되며, 작은 값을 가질수록 적합한 해를 의미한다. 해당 수식에서 적합도는 수식 (11)을 통해 경로에 포함되는 링크의 운용 비용 C_{links} 과 연산 노드에서 트래픽 처리를 위한 운용 비용 C_{nodes} 를 합한 값 C_{total} 뿐 아니라 서비스 요구사항의 만족 여부에 따라 벌점 P_{total} 을 고려한다. 벌점은 수식 (12)에 정의된 것처럼 네트워크 측면의 성능 지표 $z \in Z$ 를 만족하지 못했을 경우에 부과되는 벌점 P_z^{nw} 과 연산 노드의 가용 자원 $j \in J$ 이 충분치 않은 경우의 벌점 P_j^{cp} 의 합으로 계산된다. 각 벌점은 요구사항 대비 위반된 정도에 가중치 w 를 곱한 값으로 계산되며, 수식 (13)은 특정 자원 j 에 대한 벌점을 계산하는 예시를 보인다.

4.2 그래프 신경망 기반 초기 해집단 품질 측정

유전 알고리즘 기반 라우팅에서 일반적으로 초기 해집단을 준비하기 위해 사용하는 무작위 경로 생성 기법은 네트워크 규모가 클수록 연산 부하가 크다는 한계가 존재한다. 따라서, 가능한 모든 임의 경로를 구하는 것 대신에 홑 수와 같이 제한 사항을 둔 상태에서 일부 경로를 탐색한 후 이를 초기 해집단으로 활용하는 방법이 대안이 되고 있다. 하지만, 이와 같은 초기 해집단 구성 방법은 해집단에 어떤 경로들이 포함되는지에 따라 유전 알고리즘의 수렴 시간과 최적해의 품질이 가변적이라는 특징이 있다. 즉, 초기 해집단의 구성이 편향적일 경우 지역 최적값으로 수렴하거나, 연산에 시간이 오래 걸릴 수 있다 [24].

이와 같은 현상을 방지하기 위해 제안하는 유전 알고리즘 기반 라우팅은 그래프 신경망을 활용하여 초기 해집단을 평가하여 높은 평가 값으로 예측된 해집단을 대상으로 유전 알고리즘을 수행함으로써 해당 문제를 해결한다. 그림 3은 그래프 신경망을 도입하여 유전 알고리즘 기반 라우팅 방법에서의 해집단 품질 평가 방법 구조를 나타낸다.

제안된 구조는 네트워크 토폴로지 정보를 처리하는 그래프 신경망과 유전 알고리즘의 해집단 정보를 처리하는 순방향 신경망을 결합한 구조를 가진다. 네트워크 정보 처리를 위한 그래프 신경망 부분은 가우시안 혼합 모델 합성 네트워크 (Gaussian Mixture Model Convolutional Networks) [25]를 사

용하여 노드 간의 연결 관계와 특성 정보를 학습한다. 각 노드는 노드 종류 식별자 (e.g., 네트워크 노드 또는 연산 노드), 운용 비용, 가용 CPU, 메모리, GPU 자원 정보를 포함하며, 링크는 가용 대역폭, 지연 시간, 가용성, 운용 비용 정보를 포함한다.

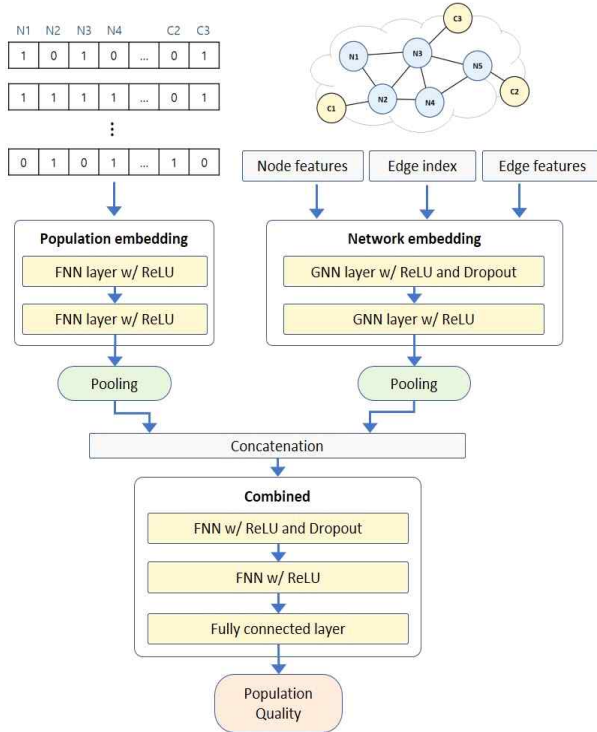


그림 3. 그래프 신경망 기반 유전 알고리즘 해집단 품질 평가
Fig. 3. Evaluation of population quality in GA based on GNN

한편, 순방향 신경망은 유전 알고리즘의 초기 해집단 정보를 처리하는 역할을 수행하며, 각 경로를 표현하는 염색체 집합을 처리하여 해당 해집단의 특성을 통합하여 학습한다. 이 때, 경로를 표현하는 기존 염색체는 각 노드 ID에 해당하는 유전자가 1의 값을 가지고, 그렇지 않은 경우에는 0이 할당되는 형태로 변환되어 처리된다. 즉, 변환된 염색체의 크기는 네트워크 내 경로가 될 수 있는 모든 노드 ID를 수용할 수 있는 규격으로 정의된다. 그래프 신경망과 순방향 신경망에서 각각 추출된 네트워크 특성과 해집단 특성은 풀링 (Pooling)된 후 결합 계층을 통해 통합되어 최종적으로 해당 네트워크에서 해집단이 갖는 품질 점수를 예측한다. 이를 기반으로, 본 논문에서 제안하는 유전 알고리즘 기반 라우팅 방법은 예측되는 품질 점수가 높은 초기 해집단을 대상으로 연산을 수행함으로써 효율적으로 각 서비스별 트래픽 경로를 결정할 수 있다.

V. 구현 및 성능 평가

5.1. 실험 환경 구성 및 성능 평가 시나리오

그래프 신경망 모델을 통해 우수한 품질을 갖춘 것으로 평가되는 초기 해집단을 활용한 유전 알고리즘 기반 라우팅 기능은 파이썬을 기반으로 구현되었다. 이 때, 초기 해집단과 네트워크로부터 특성을 추출하기 위한 순방향 신경망과 그래프 신경망은 PyTorch로 구현되었다.

라우팅이 수행되는 네트워크 토폴로지는 실제 통신 네트워크 정보를 제공하는 Internet Topology Zoo [26]에서 Xspedius 토폴로지와 Rnp 토폴로지를 사용하였다. Xspedius 토폴로지에는 34개의 노드로 구성되며, 유방향 그래프로 표현했을 때 98개의 링크로 연결된다. 한편, Rnp 토폴로지는 31개의 노드와 68개의 유방향 링크로 표현된다. 그림 4와 그림 5는 각 토폴로지의 예시를 보인다.

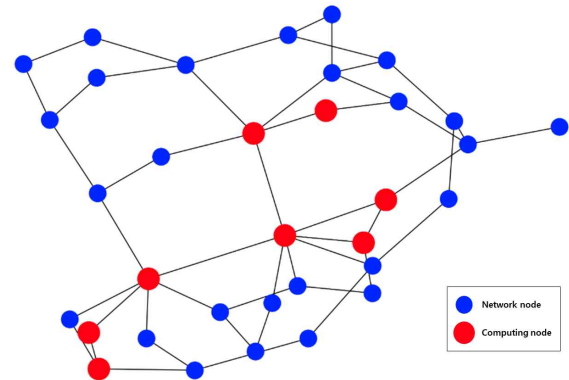


그림 4. Xspedius 네트워크 토폴로지 예시
Fig. 4. Example of Network topology for Xspedius

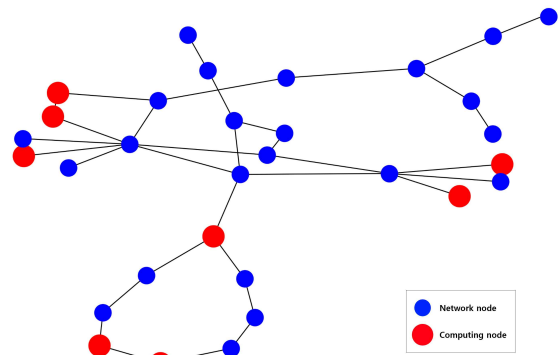


그림 5. Rnp 네트워크 토폴로지 예시
Fig. 5. Example of Network topology for Rnp

라우팅은 임의의 네트워크 노드를 출발지로 하여 단일 연산 노드를 목적지로 하는 경로를 탐색하는 것이다. 따라서, 각 토폴로지의 노드 중 연산 노드의 역할을 수행하는 노드를 할당해야 하며, 이를 위해 총 노드 수의 1/4보다 작은 임의 개수의 노드를 연산 노드로 선택한다.

네트워크에서 라우팅이 필요한 서비스 요청은 최소 전송 용량, 최대 지연 시간, 최소 가용성, 연산 자원 (CPU, 메모리, GPU)을 요구사항으로 가지며, 무작위 개수를 생성하여 성능 평가를 수행한다. 유전 알고리즘 관련 변수로는 해집단 크기, 세대 수 등과 적합도 함수의 벌점에 반영되는 가중치가 있다. 또한, 모델 학습은 미니배치 (Mini-batch) 학습으로 수행되며, 데이터셋은 총 10,000개의 샘플을 활용하였고 그 중 80%를 학습용으로 쓰고 20%는 검증용으로 사용했다. 모델 학습은 Xspedius 토폴로지에서 수행되었다. 상세한 환경 설정 정보는 표 1에 명시되어 있다.

표 1. 시뮬레이션 및 환경 설정

Table 1. Simulation and environment configuration

Category	Type	Value
Environment	Link capacity (Gbps)	{50,100, 200}
	Link availability	[0.9999, 0.999999]
	Link operating cost	[1, 10]
	Computing operating cost	[1, 20]
Service req.	Transmission capacity (Gbps)	[0.1,10]
	Maximum delay (ms)	[10, 100]
	Minimum availability	[0.9, 0.999]
	Required resource - CPU	[1, 10]
	Required resource - RAM	[1, 100]
	Required resource - GPU	[100, 1000]
GA	Population size	10
	Generations	100
	Mutation rate	0.1
	Crossover rate	0.8
Training	Hidden layer units	64
	Mini-batch size	32
	Learning rate	0.001
	Epochs	100
	Optimizer	Adam

5.2. 성능 분석

성능 평가를 위해 라우팅이 필요한 서비스 요청을 50, 75, 100, 200개 생성하여 유전 알고리즘 기반 라우팅을 수행한 후 성능 분석을 수행하였다. 각 개수마다 100회 반복 생성하여 수행한 후 평균값을 계산하여 성능 분석을 하였다. 이 때, 제안하는 유전 알고리즘 기반 라우팅을 통해 경로를 결정하는 과정에서, 무작위로 생성한 초기 해집단을 기반으로

수행한 경우와 초기 해집단 모델로 5회 반복을 통해 가장 높은 품질로 예측된 초기 해집단을 대상으로 수행한 경우를 비교하였다. 각 경우는 GA(Base) 및 GA(QP, Quality Prediction)로 표기된다.

첫 번째 성능 분석 결과로서, 그림 6는 요청 수에 따른 총 운용 비용을 Xspedius 토폴로지에서 비교한 것이다. 이 때, 운용 비용은 라우팅을 통해 결정된 경로에서 소요되는 비용을 의미하며, 수식 (1)을 통해 계산된다. 초기 해집단 품질 모델을 적용한 경우에는 모든 요청 수에서 적용하지 않은 경우 대비 더 낮은 비용을 유지하였으며, 요청 수가 많을수록 비용 차이가 크게 나타난다. 특히, 요청 수가 네트워크에서 충분히 수용할 수 있는 작은 수인 50과 75인 경우 두 방법의 차이는 크지 않지만, 100 또는 200개의 요청에서는 품질 평가 모델을 쓴 방법이 10~15% 낮은 운용 비용을 보인다.

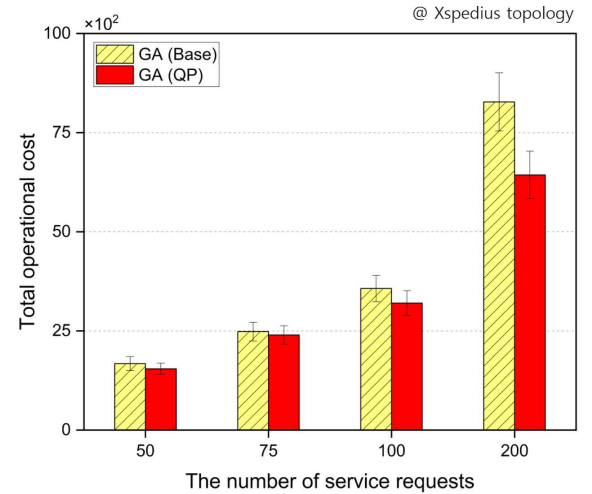


그림 6. 유전 알고리즘 기반 라우팅의 평균 운용 비용

Fig. 6. Average operational cost of GA-based routing

다음 성능 분석 지표인 서비스 요청 수에 따른 성공률은 유전 알고리즘 기반 라우팅에 의해 결정된 경로들이 각 서비스의 요구사항을 만족했는지 여부를 나타내는 값이다. 마찬가지로, Xspedius 토폴로지에서 성능 평가를 수행하였으며, 그림 7은 측정된 성공률의 분포를 상자 수염 그림 (box-and-whisker plot) 형태로 시각화한 것이다. 이에 따르면, 품질 평가 모델을 활용한 경우에는 모든 구간에서 활용하지 않은 경우 대비 안정적이고 높은 성공률을 보인다. 특히, 과부하 상황을 보이는 서비스 요청 개수 200개인 경우에도 품질 모델을 활용하지 않은 라우팅 방법은 중간 값이 30% 이하로 떨어지는 반면, 품질 모델을 사용하면 45~50% 수준을 유지하

였다. 한편, 서비스 요청 개수가 50개인 경우에 GA (QP)에서 최대 값이 100%을 넘는 것처럼 표현되었으나, 이는 최대 값 위치가 제 3분위 값을 기준으로 표현되는 상자 수염 그림의 특성으로 인한 것이며 실제 측정 성공률은 100%을 넘지 않는다.

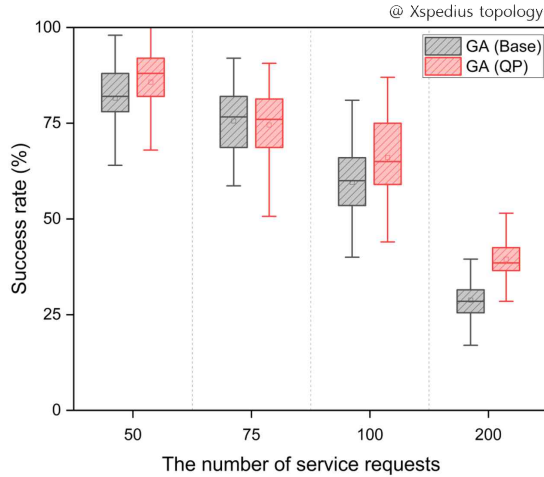


그림 7. 유전 알고리즘 기반 라우팅의 서비스 성공률
Fig. 7. Service success rate of GA-based routing

Xspedius 토폴로지에서 측정된 서비스 요청 수에 따른 평균 가용성은 그림 8을 통해 보이고 있다. 전반적으로 품질 평가 모델 적용 여부와 관계없이 모두 99.97% 이상의 높은 가용성을 유지하지만, 요청 수가 증가할수록 성능 차이가 커지는 것으로 평가되었다. 또한, 서비스 요청 수가 늘어나더라도 품질 평가 모델을 적용한 경우는 비교적 분산이 작은 형태로 안정적인 가용성을 가지는 형태인 반면, 그렇지 않은 경우에는 큰 분산을 가지며 가용성이 떨어지는 경향을 보였다.

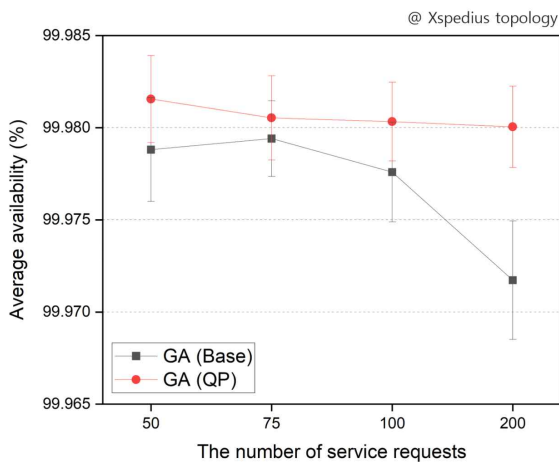


그림 8. 유전 알고리즘 기반 라우팅의 평균 가용성
Fig. 8. Average availability of GA-based routing

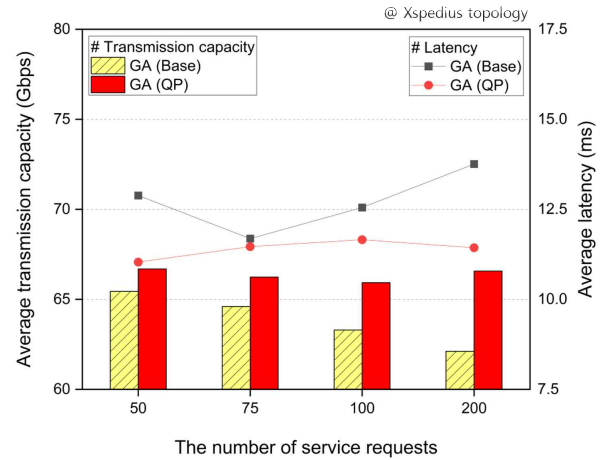


그림 9. 평균 전송량 및 지연 시간
Fig. 9. Average transmission capacity and latency

그 외에도 그림 9는 서비스 요청 수에 따른 평균 전송 용량과 평균 지연 시간을 Xspedius 토폴로지에서 비교한 결과를 나타낸다. 평균 가용성 결과와 유사한 형태로, 평균 전송 용량과 평균 지연 시간 측면에서도 품질 평가 모델을 적용한 경우가 모든 구간에서 높은 전송 용량과 작은 지연 시간을 갖는 것으로 관측되었다. 이는 서비스 요청 수가 늘어남에 따라 과부하 상황에 이르더라도, 품질 평가 모델을 적용한 경우에는 비교적 안정적인 성능을 유지할 수 있다는 것을 입증한다.

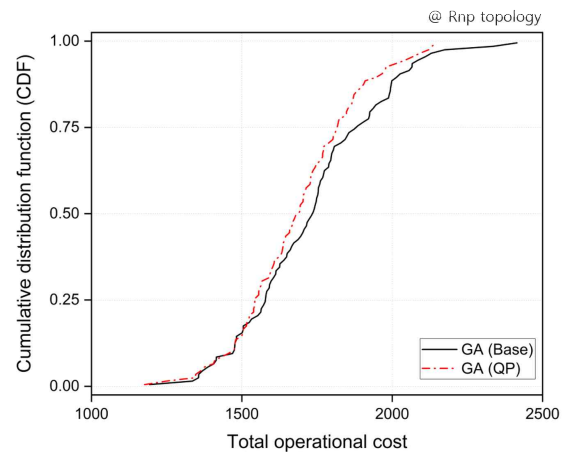


그림 10. 운용 비용 누적 분포 함수
Fig. 10. CDF of total operational cost

마지막으로, Xspedius 토폴로지에서 학습된 모델이 다른 네트워크 토폴로지에서 라우팅 성능 개선 효과를 보이는지 검증하였다. 그림 10은 Rnp 토폴로지에서 측정된 총 운용 비용의 평균값을 누적 분

포 함수 (CDF, Cumulative Distribution Function) 형태로 나타낸 것이다. 이 때, 총 운용 비용은 50개의 서비스 요청을 무작위로 생성하여 라우팅을 수행하는 것을 100회 반복한 후, 각 반복에서 계산된 운용 비용의 평균으로 계산되었다. 실험 결과, Rnp 토폴로지에서도 품질 모델을 사용한 경우가 그렇지 않은 경우 대비 전반적으로 낮은 운용 비용을 달성하고 있음을 확인할 수 있었다. 특히, 운용 비용 1500~2000 사이의 범위에서 품질 모델을 사용한 경우의 CDF 곡선이 더 빠르게 상승하며, 이는 동일한 확률 수준에서 더 낮은 비용을 실현했음을 의미한다.

VI. 결 론

본 논문에서는 동적 네트워크 환경에서 서비스 요구사항을 만족하는 효율적인 라우팅 경로를 결정하기 위해, 그래픽 신경망 (GNN)을 기반으로 초기 해집단의 품질을 평가하고 이를 유전 알고리즘 (GA)과 연동한 라우팅 기법을 제안하였다. 성능 평가 결과, 기존의 무작위 초기 해집단 생성 방법을 활용한 유전 알고리즘 기반 라우팅 방식과 비교했을 때, 제안하는 품질 평가 모델을 연동한 방식은 초기 탐색 공간의 품질을 사전에 예측하여 우수한 해, 즉, 최적 경로를 안정적으로 도출하였다.

향후 연구로는 운용 비용 최소화 뿐만 아니라 라우팅 알고리즘 수행에 필요한 연산 비용도 함께 고려하는 최적화 문제로 확장할 예정이다. 또한, 다중 목표 최적화를 위해 그래픽 신경망과 유전 알고리즘 간 연계 구조를 개선할 예정이다. 성능 평가 방법 측면에서도 대규모 또는 실시간 환경에서 검증을 수행하고, 해당 결과를 바탕으로 제안하는 방법의 일반화 성능과 학습 성능 개선 방안을 모색할 것이다. 그 밖에도, 유전 알고리즘 외 메타 휴리스틱 기반 방법과의 융합을 통한 라우팅 기술 연구와 함께 MEC 환경 또는 위성 통신망 등을 고려한 가변적인 네트워크 토폴로지 시나리오에서 성능 검증을 수행할 예정이다.

References

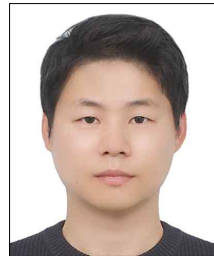
[1] Cerroni, Walter, et al. "Network Softwarization and Management." *IEEE Communications Magazine* 58.10 (2020): 14-15.
[2] Gyu Seon Ki, et al. "Reinforcement Learning-Based

Dynamic Routing for Robust Optimization of Low Earth Orbit (LEO) Satellite Communication Networks," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 48, no. 9, pp. 1123-1134, 2023.

- [3] Sanghoon Lee, et al. "QoS-aware Routing Algorithm in Satellite and Ground Integrated Networks," *The Journal of Korean Institute of Communications and Information Sciences*, vol. 50, no. 5, pp. 781-789, 2025.
[4] Amin, Rashid, et al. "A survey on machine learning techniques for routing optimization in SDN." *IEEE Access* 9 (2021): 104582-104611.
[5] Mirjalili, Seyedali, and Seyedali Mirjalili. "Genetic algorithm." *Evolutionary algorithms and neural networks: Theory and applications* (2019): 43-55.
[6] Maaranen, Heikki, Kaisa Miettinen, and Antti Penttinen. "On initial populations of a genetic algorithm for continuous optimization problems." *Journal of Global Optimization* 37 (2007): 405-436.
[7] Akhter, Farzana, and Joseph P. Near. "Energy Efficient Data Routing for IoT Networks Using Genetic Algorithm." 2024 *International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2024.
[8] Patel, Jatinkumar, and Hosam El-Ocla. "Energy efficient routing protocol in sensor networks using genetic algorithm." *Sensors* 21.21 (2021): 7060.
[9] Wang, Ning, et al. "An overview of routing optimization for internet traffic engineering." *IEEE Communications Surveys & Tutorials* 10.1 (2008): 36-56.
[10] Ramasubramanian, Venugopalan, et al. "On the treeness of internet latency and bandwidth." *Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems*. 2009.
[11] Sangheung Lee, et al. "A Routing Protocol for Wireless SDN in Mission-Critical Unmanned Systems," *The Journal of Korean Institute of Communications and Information Sciences*, vol.

- 47, no. 1, pp. 21-27, 2022.
- [12] Vinayakumar, Ravi, K. P. Soman, and Prabakaran Poornachandran. "Applying deep learning approaches for network traffic prediction." 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2017.
- [13] Yuan, Ruixi, et al. "An SVM-based machine learning method for accurate internet traffic classification." Information Systems Frontiers 12 (2010): 149-156.
- [14] Almasan, Paul, et al. "ENERO: Efficient real-time WAN routing optimization with Deep Reinforcement Learning." Computer Networks 214 (2022): 109166.
- [15] Wei, Wenting, et al. "Grace: Toward Routing in Dynamic Network Environments With Graph Embedding." IEEE Transactions on Networking (2025).
- [16] Ding, Mingjie, et al. "GROM: A generalized routing optimization method with graph neural network and deep reinforcement learning." Journal of Network and Computer Applications 229 (2024): 103927.
- [17] Sun, Geng, et al. "Large language model (llm)-enabled graphs in dynamic networking." IEEE Network (2024).
- [18] Çavdar, Tuğrul, and Şeyma Aymaz. "New approach to dynamic load balancing in software-defined network-based data centers." ETRI Journal 45.3 (2023): 433-447.
- [19] Bhardwaj, Antra, and Hosam El-Ocla. "Multipath routing protocol using genetic algorithm in mobile ad hoc networks." IEEE Access 8 (2020): 177534-177548.
- [20] Riveros-Rojas, G. J., Cespedes-Sanchez, P. P., Pinto-Roa, D. P., & Legal-Ayala, H. (2024). Energy-and-blocking-aware routing and device assignment in software-defined networking—a MILP and genetic algorithm approach. Mathematical and Computational Applications, 29(2), 18.
- [21] Tabaeiaghdaei, Seyedali, et al. "Carbon-aware global routing in path-aware networks." Proceedings of the 14th ACM International Conference on Future Energy Systems. 2023.
- [22] Ding, Hongqing, et al. "Optimal routing and heterogeneous resource allocation for computing-aware networks." ICT Express 10.3 (2024): 614-619.
- [23] Garey, Michael R., and David S. Johnson. Computers and intractability. Vol. 29. New York: wh freeman, 2002
- [24] Abu-Lebdeh, Ghassan, and Rahim F. Benekohal. "Convergence variability and population sizing in micro-genetic algorithms." Computer-Aided Civil and Infrastructure Engineering 14.5 (1999): 321-334.
- [25] Reynolds, Douglas. "Gaussian mixture models." Encyclopedia of biometrics. Springer, Boston, MA, 2009. 659-663.
- [26] Knight, Simon, et al. "The internet topology zoo." IEEE Journal on Selected Areas in Communications 29.9 (2011): 1765-1775.

이 도 영 (Do-Young Lee)



2015 건국대학교 컴퓨터공학과 학사
2021 포항공과대학교 컴퓨터공학과 박사
2021~2023 삼성전자 네트워크 사업부 Staff Engineer
2023~현재 한국전자통신연구원 선임연구원

<관심분야> 6G, 네트워크 최적화, 지능형 네트워크

김 태 연 (Tae-Yeon Kim)



1990 중앙대학교 전자계산학과 학사
1992 중앙대학교 전자계산학과 석사
2008 충북대학교 컴퓨터과학과 박사
1992~현재 한국전자통신연구원 책임연구원

2019 ~ 현재 한국전자통신연구원 지능네트워크 연구실 실장

<관심분야> 6G, Network For/By AI

한국통신학회 통신망운용관리연구회

KNOM Review Vol.28/ No.1

ISSN : 2287-1543(Online)

2025년 8월 31일 인쇄

2025년 8월 31일 발행

발행인/ 김우태

편집인/ 김명섭 편집위원장

발행처/ 한국통신학회 통신망운용관리연구회
서울시 서초구 서초동 1330-8 현대기림오피스텔 1504동 6호
전화 : 02-3453-5555
홈페이지 : www.knom.or.kr/knom-review/