

GNN 기반 사용자 방문 분석을 통한 개인 PoI 추천 시스템 설계 및 구현

권혁찬*¹, 이유진*², 홍윤기*³, 김유진*⁴, 정기현**

Design and Implementation of Personalized PoI Recommendation System with GNN-based User Visiting Analysis

Hyukchan Gwon*¹, Yujin Lee*², Yunki Hong*³, Yujin Kim*⁴, and Kihyun Jung**

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(2021R111A3049788)

요 약

본 연구는 그래프 신경망 (GNN, Graph Neural Network)을 활용하여 사용자 특성, 방문지, 상호작용 데이터를 기반으로 개인화된 관심 지점인 PoI 추천 시스템을 개발하고자 한다. 복잡한 사용자 데이터를 그래프 구조로 모델링함으로써, 사용자-아이템 간 분석을 극대화하고 추천 시스템의 성능을 향상시킨다. 본 연구에서는 GCN, GAT, GraphSAGE, ChevConv Layer, 멀티-헤드 어텐션 기법 등의 방법을 사용하여 M1, M2, NGCF 및 그에 따른 Fine-Tuning된 모델을 개발하여 사용자 특성, 방문지, 상호작용 데이터를 분석하고 PoI를 추천한다. 연구 결과에서는 사용자 데이터를 기반으로 학습된 모델이 유사한 특성의 새로운 지역을 성공적으로 추천함으로써 개인화된 사용자 경험을 제공하고 그래프 데이터와 그래프 신경망을 활용한 다양한 접근 방식을 보여준다.

Abstract

This study utilizes Graph Neural Networks (GNN) to develop a personalized Point of Interest (PoI) recommendation system based on user characteristics, visitation data, and interaction data. Complex user data is modeled into a graph structure, maximizing user-item interaction analysis and enhancing the performance of the recommendation system. This research employs various methods such as GCN, GAT, GraphSAGE, ChevConv Layer, and multi-head attention techniques to develop M1, M2, NGCF models, and their fine-tuned versions to analyze user characteristics, visitation, and interaction data and to recommend PoIs. The results show that the model, trained on user data, successfully recommends new areas with similar characteristics, providing a personalized user experience and proposing a variety of approaches utilizing graph data and graph neural networks.

Keywords

graph neural networks, GNN, point of interest, PoI, data modeling, personalized user experience

* 국립경국대학교(국립안동대학교) 소프트웨어융합학과

- ORCID¹: <https://orcid.org/0009-0007-0060-7872>

- ORCID²: <https://orcid.org/0009-0001-0840-7900>

- ORCID³: <https://orcid.org/0009-0006-4624-773X>

- ORCID⁴: <https://orcid.org/0009-0005-9762-5262>

** 국립경국대학교(국립안동대학교) 소프트웨어융합학과 교수
(교신저자)

- ORCID: <https://orcid.org/0000-0002-0662-8355>

• Received: Jan. 09, 2025, Revised: Apr. 09, 2025, Accepted: Apr. 12, 2025

• Corresponding Author: Ki-Hyun Jung

Dept. of Software Convergence, Gyeongbuk National University
(Andong National University), Korea

Tel.: +82-54-820-7968, Email: kingjung@anu.ac.kr

1. 서 론

현대 사회에서 디지털 정보의 폭발적 증가와 함께, 데이터는 비즈니스 및 의사결정의 핵심 자원으로 자리 잡고 있다. 특히 사용자의 행동 패턴을 분석하여 요구와 선호도를 예측하는 것은 전자상거래, 소셜 미디어, 온라인 광고 등 다양한 분야에서 중요한 가치를 창출하고 있다[1]. 그중에서도 개인화된 맞춤형 추천 시스템은 온라인 플랫폼에서 핵심적인 역할을 하며, 사용자 경험을 극대화하고, 고객 충성도와 상호작용을 증진하는 데 크게 기여하고 있다[2].

추천 시스템을 성공적으로 구현하기 위해서는 복잡한 사용자 데이터와 상호작용 패턴을 효과적으로 분석하고 모델링하는 것이 필수적이다. 기존의 추천 시스템은 주로 협업 필터링(Collaborative filtering)이나 일반적인 딥러닝 모델을 활용해왔다. 협업 필터링은 사용자-아이템 간의 상호작용 매트릭스를 기반으로 하여 유사한 선호도를 가진 사용자들의 패턴을 분석하지만, 데이터의 희소성(Sparsity) 문제와 콜드 스타트(Cold start) 문제에 취약하다는 한계를 지닌다. 또한 일반적인 딥러닝 모델은 사용자와 아이템 간의 복잡한 관계성을 단순한 피쳐(Feature)의 집합으로만 다루어 실제 존재하는 상호연결성을 충분히 반영하지 못한다는 제약이 있다. 그러나, PoI(Point of Interest) 추천 시스템의 경우 위치 정보, 시간적 맥락, 사용자 선호도 등 다양한 특성을 동시에 고려할 수 있다[3].

본 연구에서는 이러한 배경을 바탕으로 그래프 신경망(GNN, Graph Neural Network)을 활용하여 사용자의 방문 이력과 상호작용 데이터를 심층적으로 분석하고, 이를 통해 개인화된 관심 지점(PoI)을 추천하는 시스템을 제안한다.

GNN은 다음과 같은 고유한 장점으로 효과적인 추천 시스템 구현을 가능하게 한다. 첫째, GNN은 사용자-장소-시간 등 다차원적 관계를 하나의 통합된 그래프 구조로 자연스럽게 모델링할 수 있어, 복잡한 상호작용 패턴을 포괄적으로 포착할 수 있다. 둘째, 그래프의 메시지 전달(Message passing) 메커니즘을 통해 직접적인 상호작용이 없는 노드 간에도 정보를 전파할 수 있어, 데이터 희소성 문제를

효과적으로 해결할 수 있다. 셋째, GNN의 계층적 구조는 지역적(Local) 및 전역적(Global) 패턴을 동시에 학습할 수 있어, 사용자의 개별적 선호도뿐만 아니라 전체적인 추세를 파악할 수 있다. 마지막으로, GNN은 시간에 따른 사용자의 방문 순서와 같은 순차적 패턴을 엣지의 특성으로 인코딩할 수 있어, 동적인 사용자 행동을 더욱 정교하게 모델링할 수 있다.

GNN은 그래프 구조 데이터를 처리하는 데 최적화된 심층 학습 모델로, 노드(Node)와 엣지(Edge)로 표현된 데이터 간의 복잡한 연관성을 학습할 수 있는 특징을 지닌다. 이러한 구조는 사용자의 행동 패턴과 상호작용을 자연스럽게 그래프 형태로 표현할 수 있게 하며, GNN은 그래프 구조 내에서 노드의 특성을 학습하고 노드 간의 연결 패턴을 파악하여 네트워크 내 정보를 효과적으로 추출할 수 있다. 이 과정에서 GNN은 사용자 간의 상호 연결성을 분석하고, 각 사용자의 개별적인 성향을 반영한 맞춤형 분석이 가능하다.

본 연구는 GNN의 이론적 배경을 바탕으로 다양한 그래프 기반의 신경망 아키텍처를 활용한 효과적인 PoI 추천 모델 구현 방법을 제시한다. 제안된 PoI 추천 시스템은 GNN의 강점을 활용하여 사용자의 방문 이력과 선호도를 분석하며, 이 과정에서 AI-Hub에서 제공하는 실제 데이터인 국내 여행 로그 데이터를 사용해 시스템의 실질적인 적용 가능성을 검증하였다. 이 시스템은 사용자가 과거에 방문했거나 상호작용을 한 장소를 기반으로 유사한 특성을 가진 새로운 장소를 추천하여 개인화된 경험을 제공하는 것을 목표로 한다. 이를 위해 NGCF(Neural Graph Collaborative Filtering), GCN(Graph Convolutional Network) 등의 다양한 GNN 아키텍처를 실험적으로 탐구하고, 각 아키텍처의 적합성과 성능을 분석하였다.

나아가 본 연구는 GNN 기술이 어떻게 사용자 데이터를 더욱 세밀하게 분석하고, 이를 통해 사용자의 숨겨진 요구와 선호를 예측하는데 기여할 수 있는 방식에 대한 실질적 검토를 다룬다. 궁극적인 목표는 GNN을 활용하여 기존의 정적 추천 시스템을 넘어서는, 보다 정확하고 개인화된 추천 시스템

을 구축하는 것이다. 이를 통해 실제 응용에서 가능성을 모색하고자 한다. 특히, 다양한 GNN 아키텍처의 비교 및 분석과 실험을 통해 추천 시스템의 성능을 정량적으로 평가한다. 이러한 평가를 바탕으로 실제 환경에서의 적용 가능성을 검증하여 GNN의 실용적 가치를 입증하는 것을 목표로 한다.

본 논문의 구성은 다음과 같다. II장에서는 관련 연구로서 GNN과 관련된 다양한 추천 시스템 기술 및 알고리즘을 소개하고, NGCF, GCN 등의 GNN 아키텍처와 그 응용을 설명한다. III장에서는 본 연구에서 제안하는 PoI 추천 시스템의 설계 및 구현 방법을 구체적으로 다루며, 데이터 수집, 전처리, 그래프 생성 과정 및 GNN 아키텍처의 선택과 적용 방법에 관해 논의한다. 또한, 제안된 시스템을 실험적으로 구현하고, 이를 통해 실시한 성능 평가 결과를 분석한다. 마지막으로 IV장에서는 본 연구의 결론을 제시하며, 연구의 한계점과 향후 연구 방향에 대해 논의한다.

II. 관련 연구

2.1 GNN

복잡한 그래프 구조의 데이터를 효과적으로 처리하기 위해 제안된 심층 학습 아키텍처이다. 이 네트워크는 그래프 데이터의 노드와 엣지로 구성된 구조에서 각 노드의 특성을 학습하고 전체 그래프의 복잡한 패턴을 파악하는 것을 목표로 한다.

GNN은 각 노드가 인접한 이웃 노드들로부터 정보를 수집하고 집계하여 자신의 특성 표현을 반복적으로 업데이트하며 동작한다. 이 과정에서 일반적으로 가중치 합, 평균 또는 최댓값 등의 집계 함수를 사용하며, 각 노드는 여러 레이어를 거치며 최종적인 임베딩 벡터를 형성한다. 학습 과정에서는 주로 미분 함수와 역전파 알고리즘을 적용해 최적의 네트워크 파라미터를 추정한다. GNN은 기존 컨볼루션 신경망(CNN)과 달리 복잡한 그래프 데이터를 자연스럽게 처리할 수 있는 장점이 있다. 하지만 대규모 그래프나 동적 그래프 데이터 처리 시 계산 복잡도와 효율성 개선이 주요 과제로 남아있다. 최

근 연구에서는 효율적인 그래프 샘플링, 병렬 처리 기법, 그래프 구조 특성을 반영한 새로운 아키텍처 등을 통하여 이러한 문제를 해결하려는 연구를 진행하고 있다[4].

2.2 NGCF

NGCF는 사용자-항목 간의 복잡한 상호작용을 효과적으로 모델링 하기 위한 기법이다. 사용자-항목 상호작용을 그래프 구조로 모델링하고, 그래프 노드 간 임베딩을 통해 연결 정보를 활용한다. 이를 통해 단순 직접 연결뿐만 아니라 간접적 고차 연결까지 학습에 명시적으로 반영할 수 있다.

NGCF는 임베딩 전과 과정에서 노드의 임베딩이 연결된 이웃 노드들의 임베딩과 결합하도록 함으로써 필터링 효과를 극대화한다. 이 과정은 계층적 구조를 통해 각 레이어마다 노드 임베딩의 특성이 강화되는 방식으로 진행된다. 복수의 그래프 컨볼루션 레이어를 통해 노드 임베딩이 반복적으로 업데이트되며, 각 레이어는 그래프의 구조적 정보를 임베딩에 인코딩한다.

NGCF의 접근법은 상호작용 데이터를 활용하는 기존 행렬 분해 기법과 차별화된다. 사용자-항목 상호작용 그래프에서 전파되는 임베딩을 통해 보다 정밀한 사용자 및 항목 프로파일을 생성할 수 있다. 이는 희소 데이터에서 사용자 선호도 예측에 유리하며 개인화된 추천이 가능해진다[5].

2.3 GCN

그래프 구조 데이터에 대한 심층 학습 접근법으로, 스펙트럴 그래프 이론에 기반한 일차 근사화를 통해 그래프상의 노드 분류 작업에 중점을 둔다. 이 모델은 그래프의 인접행렬과 노드 특성을 입력으로 하여 그래프 노드 간 정보 전파 및 특정 학습을 수행한다.

GCN은 레이어 별 전파 규칙을 통해 복잡한 네트워크 구조를 직접 모델링한다. 각 노드의 이웃 연결성을 반영하여 국소적 그래프 구조와 노드 특성으로부터 새로운 특성을 도출한다.

이를 통해 반지도 학습 설정에서 우수한 성능을 보이며, 인용 네트워크, 지식 그래프 등의 데이터셋에서 기존 방법을 능가하는 결과를 얻을 수 있다. 또한, 엣지 수에 선형적인 계산 복잡도를 가지게 함으로써 대규모 그래프에 효율적으로 확장 가능하다. 하지만 노드 특성의 다양성이 증가할수록 메모리 및 계산 리소스 한계에 부딪힐 수 있다. 따라서 실제 응용 시, 이러한 제약을 고려하여 모델을 적절히 설계하고 최적화해야 한다[6].

2.4 GAT

자기주의 메커니즘을 활용하여 노드 특성을 학습한다. 이를 통해 각 노드가 그래프 내 다른 노드의 정보를 중요도에 따라 통합할 수 있어, 기존 그래프 신경망 대비 유연성과 표현력이 향상된 모델이다. GAT(Graph Attention Network)는 노드가 주변 노드에 대한 주의 가중치를 동적으로 할당하여 작동한다. 이를 통해 노드 간 복잡한 관계를 효과적으로 포착하며, 노드 분류 등의 작업에서 우수한 성능을 나타낸다. 또한 높은 계산 효율성으로 복잡한 그래프 구조 데이터에 대해서도 빠르고 정확한 학습이 가능하다. 주요 장점은 그래프 구조를 사전에 알 필요가 없다는 점이다. 이를 통해 귀납적 학습 문제에 적용할 수 있으며, 학습 시 미 출현 그래프에 대한 일반화 능력을 갖출 수 있다. 실제로 GAT는 전이/귀납 학습 문제에서 최고 수준의 성능을 보이며, 다양한 그래프 벤치마크에서 그 성능이 입증되었다[7].

2.5 GraphSAGE

노드의 이웃으로부터 특성 정보를 샘플링하고 집계하는 방식으로, 본 적 없는 노드들에 대한 임베딩을 효과적으로 생성할 수 있으며, 매우 큰 그래프 데이터셋에서도 높은 확장성과 범용성을 제공한다. GraphSAGE는 노드의 이웃으로부터 정보를 집계하는 학습 가능한 함수를 도입하였다. 각 노드는 자신의 이웃 노드들의 정보를 종합하여 더 넓은 그래프 구조를 반영한 임베딩을 생성하고, 이 과정은 반복적으로 수행된다. 각 반복은 노드가 자신의 지역적 그

래프 구조에서 더 멀리 있는 이웃의 정보를 포함하게 되며, 이를 통해 학습된 임베딩은 노드가 속한 커뮤니티의 구조적 및 기능적 역할을 반영한다. 다양한 그래프 구조와 노드 특성을 다룰 수 있도록 설계된 GraphSAGE는 텍스트 속성이나 사용자 프로필 정보 같은 풍부한 노드 특성을 활용할 수 있다. 이는 GraphSAGE가 소셜 네트워크, 인용 네트워크, 생물학적 상호작용 네트워크 등 다양한 종류의 그래프 데이터에 적용될 수 있음을 의미한다. 또한, 이 방법은 새로운 노드나 전혀 본 적이 없는 그래프에 대해서도 일반화할 수 있는 능력을 가지고 있다[8].

2.6 멀티-헤드 Attention 메커니즘

멀티-헤드 어텐션 메커니즘은 복잡한 순환 신경망(RNN)과 합성곱 신경망(CNN)을 사용하는 기존 시퀀스 변환 모델을 대체하기 위해 Transformer에 도입되었다. 이 메커니즘은 입력과 출력 간의 전역적 의존성을 효과적으로 모델링하며, 각 레이어에서 여러 어텐션 메커니즘을 동시에 처리할 수 있는 능력을 제공한다.

Transformer 모델은 인코더와 디코더로 구성되어 있으며, 이들 각각은 멀티 헤드 셀프-어텐션과 위치별 전방 피드 포워드 네트워크로 이루어진 여러 층을 포함한다. 이 구조는 모델의 병렬 처리 능력을 강화하여 빠른 훈련과 높은 자연어 처리 성능을 가능하게 한다.

멀티-헤드 어텐션 메커니즘은 Transformer의 핵심 구성 요소로서 GNN에 적용되었다. 이 접근법은 그래프 데이터의 복잡한 구조와 관계를 효과적으로 모델링하는 새로운 방법을 제공하며, 입력과 출력 간의 관계를 더 짧은 경로를 통해 포착할 수 있도록 한다. 이는 그래프 기반 자연어 처리 작업에서 뛰어난 성능을 발휘할 가능성을 열어준다[9].

2.7 ChebConv layer

ChebConv Layer는 스펙트럴 그래프 이론을 기반으로 한 그래프 신경망 접근법이며, 그래프 신호 처리 도구를 활용하여 그래프 데이터에 대한 합성곱

신경망(CNN)을 일반화한다. 특히, 체비 셰프 다항식을 이용해 지역화된 스펙트럴 필터를 구현함으로써, 고차원의 불규칙 도메인에서도 효과적인 특성 추출이 가능하다. ChebConv Layer는 중심 정점으로부터 정해진 홉 수(K-홉) 내에서만 연산을 수행하도록 설계된 필터를 사용해 그래프의 복잡성을 크게 감소시킨다.

이 기술에 Transformer의 멀티-헤드 어텐션 메커니즘을 통합함으로써, ChebConv Layer는 입력과 출력 간의 전역적 의존성을 모델링할 수 있는 능력을 향상시킨다. 멀티-헤드 어텐션은 각 어텐션 헤드가 다양한 부분집합의 정보를 포착하도록 하여, 정보의 다양성을 증대시키고, 이를 통해 더 정교한 그래프 기반 특성을 추출할 수 있다. 실제 데이터셋에 대한 실험을 통해, 이 통합된 접근법은 지역적이고 정적인 그래프 데이터의 특성을 효과적으로 학습하며, 높은 정확도와 낮은 계산 비용으로 데이터를 처리할 수 있음을 보여준다[10].

2.8 기존 추천 시스템에 대한 분석

기존 추천 시스템은 사용자 맞춤형 추천을 위해 다양한 알고리즘을 도입 및 연구하고 있다. 추천 시스템은 크게 콘텐츠 기반 필터링, 협업 필터링, 그리고 하이브리드 추천 시스템으로 구분할 수 있다. 하이브리드 추천 시스템은 다양한 알고리즘과 데이터를 결합하여 사용자에게 높은 만족도를 제공하기 위한 중요한 방법론이다.

이 방법론은 콘텐츠 기반 필터링과 협업 필터링

의 장점을 극대화하고 단점을 보완하여 추천의 정확성과 다양성을 높이는 데 중점을 둔다. 예를 들어, 여러 알고리즘의 추천 결과를 가중 평균하는 방식이나, 현재 상황에 가장 적합한 추천 알고리즘을 선택하는 방식 등이 사용될 수 있다. 그러나 이러한 하이브리드 추천 시스템에도 몇 가지 단점이 존재한다. 여러 알고리즘을 결합하는 복잡성과 높은 관리 비용이 발생한다. 또한, 각 알고리즘이나 데이터의 특성에 따라 최적의 결합 방법이 달라질 수 있어서 설계와 유지 관리가 어려울 수 있다. 또한, 사용자에게 최적화된 추천을 제공하기 위해서는 많은 양의 데이터와 계산이 필요하므로 시스템의 성능이 저하될 수 있다. 이러한 단점들을 고려하여 하이브리드 추천 시스템을 구축할 때 신중한 설계와 유지보수가 필요하며, 사용자 경험과 성능을 균형 있게 고려하여야 한다[11]. 그러므로, 본 연구에서는 그래프를 통해 사용자와 아이템을 연결하여 사용자의 특성과 경험을 모두 고려할 수 있는 GNN 모델을 활용하여 사용자에게 PoI를 추천하고자 한다.

III. 설계 및 구현

3.1 아키텍처 설계

본 연구에서 PoI 추천을 위한 인공지능의 설계 및 구현 방법으로 파이프-필터 패턴을 채택하였다. 그림 1은 파이프-필터 패턴을 사용하여 다이어그램을 작성한 것이다.

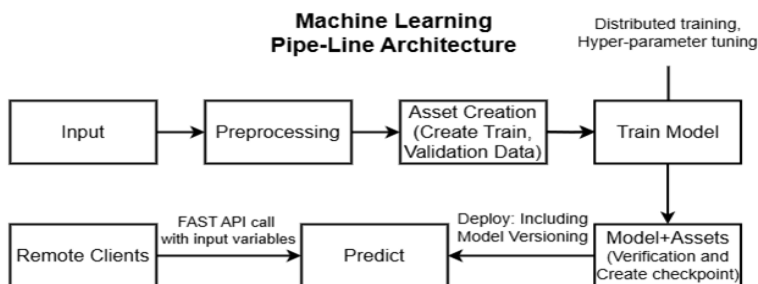


그림 1. 파이프-필터 패턴 다이어그램
Fig. 1. Pipe-filter pattern diagram

파이프-필터 패턴은 아키텍처 패턴으로 소프트웨어 아키텍처를 설계할 때 참조하는 전형적인 해결 방식으로, 데이터 스트림 작업 단계를 일련의 개별적인 요소로써 다루는 방식이다. 각 작업은 개별적인 요소인 필터 컴포넌트로 캡슐화되어 파이프를 통해 데이터를 송신하고 다음 단계에서 수신된 데이터를 처리하는 단계로 진행된다.

본 연구에서는 입력, 전처리, 필요 데이터 생성, 모델 학습, 모델 생성, 예측, 원격 클라이언트 단계로 다이어그램을 설계하고 세부적으로 분산 학습, 하이퍼-파라미터 튜닝 방식, 배포 : 모델 버전 관리, 입력 변수를 포함하는 FAST API 호출을 작성하였다. 다음은 실제 데이터 수집 및 처리, 인공지능 모델 구현에 대하여 자세하게 설명하고자 한다.

3.2 데이터 수집

목적에 맞는 인공지능 학습을 위해서 AI-Hub의 국내 여행 로그데이터를 사용한다. 국내 여행 로그 데이터는 수도권, 동부권, 서부권, 제주도 및 도서 지역으로 분류되어 있으며 총 4개의 데이터셋을 사용한다.

각 국내 여행 로그데이터는 여행자 패널 데이터, GPS 데이터, 여행 기록 데이터, 여행지 사진 데이터, 소비내역 데이터, PoI 데이터로 구성된다. 수도권 데이터 형식은 사진을 제외하고 csv파일이며 사진과 PoI 데이터를 제외하고 4,000 SET으로 이루어져 있다. 동부권, 서부권, 제주도 및 도서 지역도 마찬가지로 각 4,000 SET으로 이루어져 있으며 본 연구에서는 훈련 데이터셋의 구축으로 여행지 패널 데이터, 여행 기록 데이터의 여행객 데이터, 방문지 데이터를 사용한다.

여행객 데이터는 여행객 ID, 거주지시군구코드, 성별, 연령대, 최종학력, 최종학력이수여부, 혼인상태, 가구원수, 직업, 본인소득, 가구 소득, 여행빈도, 선호여행, 여행스타일, 여행현황, 여행동기, 여행동반자수 칼럼을 가진다. 방문지 데이터는 여행 ID, 방문지 ID, 진행순서, 방문지명, 방문일자, 주소, 좌표, 도로명코드, 지번코드, PoI ID, 체류시간, 방문지 유형, 재방문여부, 방문선택이유 코드, 숙소유형코

드, 만족도, 재방문의향, 추천의향, 시군구코드 칼럼들로 구성된다.

데이터셋은 이렇게 수집된 각 지역별 2개의 여행객 데이터와 방문지 데이터로 총 8개의 csv파일로 구성되고 전처리 작업을 수행하여 인공지능 모델 학습을 위한 그래프로 만드는 작업을 진행한다.

3.3 데이터 전처리 및 그래프 생성

데이터 전처리 작업 단계에서는 각 사용자와 방문지의 특성을 가지는 그래프를 구성하는데 필요한 요소들을 삭제, 결측치를 처리하는 방법을 사용하고 방문지 데이터와 여행객 데이터에 각각 진행하였다.

방문지 데이터는 여행 ID, 방문지명, 주소, 체류시간, 방문수단, 만족도, 재방문의향, 추천의향을 제외한 나머지 칼럼들을 삭제하고 집, 숙소, 친구/친지집, 아파트, 오피스텔, 지하철, 펜션, 호텔, 친척집, 사무실, 터미널, 정류장, 공항 등 여행지 추천 항목이 될 수 없는 값들을 삭제한다. 결측치 처리 단계에서는 만족도, 재방문의향, 추천의향, 체류시간의 결측치를 중간값으로 처리하였다. 알고리즘 1은 결측치 처리 코드이다.

알고리즘 1. 데이터셋 결측치 처리

Alg. 1. Handling missing values in dataset

Algorithm 1: Filling Missing Values in a DataFrame

Input: DataFrame with missing values

Output: DataFrame with missing values filled with median values

Function fill_missing_values(df_filtered):

```

1. tdf = df_filtered // Copy filtered DataFrame;
2. median_dgstfn = tdf['DGSTFN'].median();
3. median_revisit_intention =
   tdf['REVISIT_INTENTION'].median();
4. median_rcmdtn_intention =
   tdf['RCMDTN_INTENTION'].median();
5. median_residence_time =
   tdf['RESIDENCE_TIME_MIN'].median();
6. tdf['DGSTFN'].fillna(median_dgstfn, inplace=True)
   // Fill missing values in DGSTFN;
7.
   tdf['REVISIT_INTENTION'].fillna(median_revisit_intention,
   inplace=True);
8. tdf['RCMDTN_INTENTION'].fillna(median_rcmdtn_intention,
   inplace=True);
9. tdf['RESIDENCE_TIME_MIN'].fillna(median_residence_time,
   inplace=True);
10. return tdf // Return the processed DataFrame;
```

여행객 데이터도 마찬가지로 여행객 ID, 성별, 연령대, 가구원 수, 여행 동반자 수를 제외한 나머지를 삭제하고 결측치 확인을 진행하였다. 기본적인 전처리가 완료되면 여행객 데이터와 방문지 데이터를 결합하여 그래프를 생성한다. 본 연구에서는 그래프를 두 가지 방법으로 생성하여 PyTorch의 Geometric Data 라이브러리를 활용하여 인공지능의 학습을 진행한다.

첫 번째 방법은 여행 ID(TRAVEL_ID)를 Label Encoder를 사용하여 인코딩하고 방문지명(VISIT_AREA_NM) 칼럼의 각 방문 지역을 고유한 인덱스로 매핑하여 디서너리에 저장한다. 사용자별 데이터를 여행 ID로 구분하여 그룹화하고 사용자마다 NetworkX 라이브러리를 사용하여 새로운 그래프를 생성한다. 사용자의 방문지명을 사용자별 그래프의 각 노드에 추가하고 각 노드의 특성에는 체류시간, 재방문의향, 추천의향 값을 추가한다. 노드를 연결하는 엣지의 가중치로는 연결된 각 노드 쌍의 만족도(DFSTFN)의 평균값으로 설정하여 그래프에 추가한다.

PyTorch를 사용해서 인공지능 모델을 훈련하기 위해 PyTorch Geometric 데이터로 변환하는 작업을 진행한다. NetworkX로 생성된 그래프로부터 각 노드의 특성, 엣지의 정보를 추출하여 모든 데이터를 PyTorch 텐서로 변환한다. 또한, 재인덱싱 함수를 통해 그래프 처리와 에러를 방지하고 모델의 출력값은 방문지명으로 설정해야 하므로 Torch의 y 데이터로 방문지명을 설정한다.

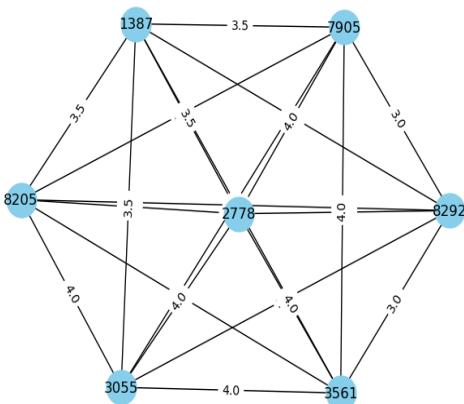


그림 2. 시각화된 생성된 그래프
Fig. 2. Visualization of the generated graph

결과적으로, 그래프는 사용자별로 방문했던 방문지명의 특성만을 고려하여 생성된 그래프 데이터셋으로 구성된다. 그림 2는 시각화된 생성된 그래프 중 하나이다.

두 번째 방법 또한 NetworkX 그래프와 PyTorch Geometric 데이터를 사용해서 생성한다. 첫 번째 방법과 비슷하게 여행 ID를 Label Encoder를 사용하여 인코딩을 진행하며 방문지명, 성별 칼럼 또한 인코딩한다. 여행 ID 별로 데이터를 그룹화하여 각 사용자에게 대한 그래프를 생성한다.

두 번째 방법에서는 각 사용자가 그래프의 노드로 추가되고 해당 사용자의 특성인 성별, 연령대, 가구원 수, 여행 동반자 수를 노드의 속성으로 추가한다. 그래프에 해당 사용자에게 대한 방문지명을 노드에 추가하고 방문지명 노드에는 별도의 속성을 지정하지 않고 사용자 노드와 방문지명 노드 사이에 엣지를 추가한다. 엣지의 속성으로는 체류시간, 재방문의향, 추천의향, 만족도 값을 설정하여 NetworkX 그래프를 생성한다. 사용자별로 생성된 그래프를 PyTorch Geometric 형식의 데이터로 변환하는데 각 노드의 특성, 엣지 인덱스, 엣지 특성, 노드 라벨을 추출한다.

사용자 노드의 경우는 성별, 연령대, 가구원 수, 동행자 수의 특성을 노드 특성에 추가하고 해당 노드의 라벨을 -1로 설정한다. -1은 사용자 노드임을 표시하는 일종의 플래이스 홀더이다.

방문지명 노드의 경우는 PyTorch Geometric 데이터를 처리하는 방식의 특성상 노드 특성의 개수를 같게 만들어야 하므로 4개의 특성을 추가하고 모든 값을 0으로 구성하고 노드 라벨은 해당 노드의 방문지명(VISIT_AREA_NM)으로 설정한다.

엣지 인덱스는 각 엣지의 시작 노드와 종료 노드 인덱스를 추가하고 노드 인덱스 매핑 디서너리를 사용하여 노드 ID를 해당 노드의 인덱스로 변환한다. 엣지 특성은 NetworkX 그래프로부터 체류시간, 재방문의향, 추천의향, 만족도를 추출하여 설정한다.

설정이 완료된 사용자별 그래프로부터 node_features, edge_index, edge_attr, node_label 리스트를 생성하여 PyTorch의 텐서로 변환한다. 변환된 텐서를 사용하여 Data 객체를 생성하고 인공지능의 입력데이터로 사용한다. 그림 3은 그래프 데이터셋 구성 과정을 순서도로 나타낸 것이다.

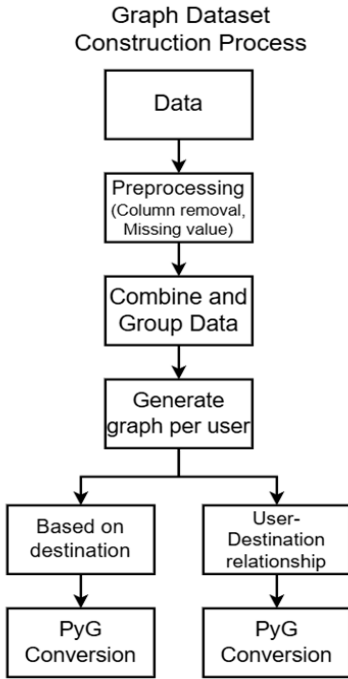


그림 3. 그래프 데이터셋 구성 과정
Fig. 3. Graph dataset construction process

3.4 인공지능 학습

사용자에게 가장 적합한 PoI를 추천하기 위해 인공지능 모델을 학습하는 단계로, 데이터 전처리와 그래프 생성 단계를 거쳐 생성된 그래프 데이터셋을 사용하여 인공지능 모델의 PoI 추천 능력을 학습하는 단계이다.

본 연구에서는 인공지능을 학습하기 이전에 그래프 데이터셋의 특성상 컴퓨터에서 그래프 데이터 리소스를 처리하는 데 많은 시간이 소요되어 Pickle 라이브러리를 활용하여 그래프 데이터셋을 모델에 맞춤화하여 구성하기 위해 CustomDataset 인스턴스와 학습 데이터와 검증데이터를 나누기 위한 DataSplitter 클래스 인스턴스를 파일화하여 미리 저장해서 불러오는 방식을 사용한다.

인공지능 학습은 두 가지 형태의 그래프 데이터셋과 자체적으로 인공지능 구조를 설계한 M1, M2 모델과 NGCF 모델, NGCF 모델을 파인 튜닝 및 하이퍼 파라미터를 조정한 모델로 학습을 진행한다. 학습에서는 첫 번째 그래프 데이터는 수도권의 데이터로만 구성되어 있고 두 번째 그래프 데이터는

수도권, 동부권, 서부권, 제주도 및 도서 지역이 모두 결합한 데이터이다. 첫 번째 그래프 데이터는 M1, M2 모델, 두 번째 그래프 데이터는 M2, NGCF, NGCF 기반 모델에 적용하였다.

3.4.1 M1 모델 구현 및 학습

첫 번째로 사용자에게 PoI 추천을 위해 M1 모델을 설계하였다. M1 모델은 첫 번째 방법의 그래프 데이터셋을 사용하여 학습을 진행하였다.

M1 모델은 입력 특징의 수와 클래스 수를 매개 변수로 받는다. M1 모델은 노드의 정보와 이웃 노드의 중요도를 학습하는 모델로 입력 특징을 16차원으로 변환하는 GCNConv 레이어, 16차원을 32차원으로 변환하여 4개의 Attention head를 사용하여 정보를 수집하는 GATConv 레이어, 다시 32*4차원의 특징을 클래스 수에 맞게 변환하는 GCNConv 레이어로 구조를 설계하고 Relu 활성화 함수를 사용하여 각 레이어의 출력을 비선형적으로 변환한다. 각 층 사이에 Dropout을 적용하여 과적합을 방지하고 마지막 층에는 Softmax 함수를 사용하고 방문지명 추천을 위한 값을 계산한다. 훈련 설정으로는 Epoch는 10, Loss는 CrossEntropyLoss, 배치 사이즈는 32, 옵티마이저로는 Adam을 사용하였다. 표 1은 모델의 훈련 결과를 나타낸다.

표 1. Epoch에 따른 손실 값 : M1 model
Table 1. Loss by epoch : M1 model

Epoch	Best loss
7	252.261
8	253.132
9	253.904
10	254.303

훈련 결과를 보면 Loss 값이 253~255 사이로 기대했던 결과 보다 Loss 값이 굉장히 높아 더 깊고 복잡한 모델인 M2 모델을 구성하였다.

3.4.2 M2 모델 구현 및 학습

M2 모델은 M1 모델을 기반으로 새로운 구조와 층을 추가하여 개발한 모델이다.

M2 모델은 두 가지 그래프 데이터셋을 사용하여 여러 번의 훈련을 진행하였다. M2 모델은 GCN, GAT, GraphSAGE, Unet 구조, Skip-Connection, Residual-Connections, 멀티-헤드 Attention 메커니즘을 활용하여 구조를 설계하였다. 노드 특징을 입력으로 받아, 그래프의 구조를 고려하여 각 노드를 분류하여 사용자에게 맞는 노드로부터 PoI를 추천할 수 있다. 활성화 함수로는 Relu 함수를 사용하고 Softmax로 추천하기 위한 값을 계산한다. Loss 값은 CrossEntropy, 배치 사이즈는 32, 옵티마이저는 Adam을 사용한다. 표 2는 M2 모델의 훈련 결과이다.

표 2는 사용자 방문지명의 특성에 특화된 첫 번째 그래프 데이터셋을 사용하여 훈련을 진행하였고 각 Epoch에 따라 모델의 Best Loss와 Best Accuracy를 나타낸다. 가장 낮은 Loss 값은 2.09이고, 가장 높은 정확도는 33.3 퍼센트의 정확도를 보이고 있다.

표 2. Epoch에 따른 손실과 정확도 : 첫 번째 데이터
Table 2. Loss and accuracy by epoch : first data

Epoch	Best loss	Best accuracy
10	2.13	26.4
100	2.09	31.7
500	2.68	33.3

표 3은 사용자 특성과 방문지명 특성을 모두 포함하는 두 번째 그래프 데이터셋을 사용한다. 가장 낮은 Loss값은 10.1, 가장 높은 정확도는 44.1 퍼센트의 정확도를 보인다.

표 3. Epoch에 따른 손실과 정확도 : 두 번째 데이터
Table 3. Loss and accuracy by epoch : second data

Epoch	Best loss	Best accuracy
6	10.5	40.3
16	10.1	44.1

3.4.3 NGCF 모델 구현 및 학습

NGCF 모델은 협업 필터링 모델로 추천 시스템에서 사용되고 사용자-아이템 간의 상호작용을 모델링할 수 있다. 본 논문에서 사용한 NGCF 모델은 사용자와 아이템을 벡터 공간에 임베딩하여 표현하고 그래프 신경망인 GCNConv 레이어를 사용하여 임베딩을 업데이트하여 학습을 진행한다. 마지막 레

이어에는 선형 레이어를 배치하여 각 사용자-아이템 쌍에 대한 예측 점수를 계산하여 사용자별 PoI를 추천할 수 있다. 훈련 설정은 epoch는 100, 옵티마이저는 Adam, Loss는 MSELoss를 사용한다. 표 4는 NGCF 모델의 훈련 결과이다.

표 4. Epoch에 따른 손실 : NGCF model
Table 4. Loss by epoch : NGCF model

Epoch	Best loss
98	7828188.504
99	7828383.817
100	7828267.7033

결과를 보았을 때 Loss값이 높아 인공지능의 성능이 보장되지 않는다고 보인다. 이를 해결하기 위해서 이전 모델을 기반으로 GCN, GAT 레이어를 추가하여 그래프 구조를 더 깊게 학습할 수 있게 파인튜닝을 진행한 모델을 설계하였다.

파인튜닝을 진행한 Advanced NGCF 모델은 NGCF 모델에 GCN, GAT레이어를 추가하는 작업을 진행하였다. Advanced NGCF 모델에서는 사용자와 아이템 임베딩을 생성하고 GCN레이어를 통과한 후 GAT 레이어를 거쳐 다시 선형 레이어를 통해 아이템 예측 점수를 계산하여 아이템을 추천한다. 또한 정확한 Loss를 계산하기 위해서 MSELoss를 기반으로 CustomLoss를 작성하여 사용한다. CustomLoss는 방문지명에 대해 특화된 계산을 하기 위해서 사용자가 가지는 -1 라벨을 가진 타겟을 손실 계산에서 제외하는 방법으로 Loss 값의 정확도를 향상시킨다. 알고리즘 2는 CustomLoss 클래스이다.

알고리즘 2. CustomLoss 클래스 코드

Alg. 2. CustomLoss class code

Algorithm 2: Custom Loss Function to Handle Invalid Targets

```

Input: Predictions, Targets
Output: Loss value
Class CustomLoss():
1. // Initialize with base loss function (MSELoss)
   Data: base_loss_function = nn.MSELoss();
2. Method forward(predictions, targets):
3. valid_indices = targets != -1 // Find valid targets;
4. if valid_indices.any() then
5. return
   base_loss_function(predictions[valid_indices],
   targets[valid_indices]) // Compute loss on valid
   data
end
6. return torch.tensor(0.0).to(predictions.device)
// Return zero if no valid targets

```

CustomLoss와 모델의 구조를 깊게 설계한 결과 Loss 값이 현저하게 감소한 것을 표 5에서 볼 수 있다.

표 5. Epoch에 따른 손실 : Advanced NGCF model
Table 5. Loss by epoch : Advanced NGCF model

Epoch	Best loss
17	8.1966
18	9.1220

Loss 값은 많이 줄었지만, 모델의 성능이 기대했던 바와는 달라, 성능 개선을 위해 모델의 구조를 수정하였다. 모델 구조가 수정된 모델은 회귀 모델과 분류 모델로 구분하여 학습을 진행하였다.

Final Advanced NGCF 모델은 기존 Advanced NGCF 모델에 GraphSAGE, ChebConv 레이어, MLP 레이어, Dropout 레이어를 추가하여 첫 번째로 회귀 모델을 구축하였다. 분류 모델은 Final Advanced NGCF 모델의 점수 예측 단계 다음에 Softmax 함수를 배치하여 분류모델로 수정하였다. 이 두 모델은 훈련 데이터의 배치 사이즈, epoch에 변화를 주며 훈련을 진행하였다. 동일한 훈련 설정은 Loss는 CustomLoss, 옵티마이저는 Adam을 사용하였다.

표 6. 최종 모델의 Epoch, batchsize에 따른 손실
Table 6. Loss of the final model on epoch and batch size

Model	Epoch	BatchSize	Best loss
Regression	20	1	9.1388
Regression	100	4	9.1051
Regression	100	8	9.1074
Classification	20	4	10.5523
Classification	100	4	10.6296

표 6은 Regression 모델에서는 epoch가 100, Batchsize가 4인 경우, Classification은 epoch가 20, BatchSize가 4인 경우에서 가장 좋은 Loss 값을 볼 수 있다.

3.5 결과 분석 및 테스트

본 연구에서는 두 가지 형태의 데이터셋과 M1, M2, NGCF, Advanced NGCF, Final Advanced NGCF 모델들을 사용해서 인공지능 훈련을 진행하였다.

표 7. 모델 별 아키텍처와 개선점

Table 7. Architecture and improvements for each model

Model	Layer-wise architecture	Improvement
M1 Model	GCN(16) → ReLU+Dropout → GAT(32, 4 heads) → ReLU+Dropout → GCN → Softmax	Baseline; simple, high loss
M2 Model	Composite (GCN, GAT, GraphSAGE, U-Net) + Skip/Residual → Multi-head → ReLU → Softmax	Advanced aggregation
Advanced NGCF Model	Embedding → GCN → extra GCN & GAT → Linear → CustomLoss	Deeper collaborative filtering; noise reduction
Final Advanced NGCF (Regression)	Embedding → NGCF Base → GraphSAGE, ChebConv, MLP, Dropout → Linear	Optimized for regression
Final Advanced NGCF (Classification)	Embedding → NGCF Base → GraphSAGE, ChebConv, MLP, Dropout → Linear → Softmax	Softmax for classification

학습모델 변화에 따른 모델의 계층적 아키텍처와 개선점은 표 7을 통해 확인할 수 있다.

수도권 데이터만을 사용한 그래프 데이터셋으로 훈련을 진행한 모델의 결과들을 제외했을 때 M2(Epoch = 100), Advanced NGCF(Epoch = 100), Regression(Epoch=100, BatchSize=4)의 결과들이 다른 모델들에 비해 좋은 성능을 보였다. 하지만 실제 임의의 입력을 통해 모델을, 테스트를 진행한 결과 M2 모델은 과적합 문제로 하나의 결과만 출력하는 문제가 있다. 나머지 두 모델의 결과는 임의의 값들을 입력할 때마다 다른 결과가 나오는 것을 확인하였다. 표 8과 표 9에서 출력 결과를 볼 수 있다.

표 8. 30대 남성의 그래프 데이터로 학습된 결과
Table 8. Results trained on graph data of men in 30s

Visit Area	Output
디앤드파트먼트 제주	서귀포매일올레시장
제주동문시장	

표 9. 20대 여성의 그래프 데이터로 학습된 결과
Table 9. Results trained on graph data of women in 20s

Visit Area	Output
감성점빵	우도, 해운대블루라인파크, 방화수류정, 제주국제공항
국수먹고가게	
이창명짜장면시키신분	

이에 따라, 그래프 데이터를 본 논문과 같은 방식으로 구성한다면, Loss 계산 시 사용자 노드 부분의 영향을 제거하기 위한 CustomLoss 함수를 사용한다. 또한, 적당한 레이어들의 추가로 모델 성능을 향상시킬 수 있다.

결과적으로, GNN 모델을 활용하여 사용자 특성과 방문지 특성을 결합한 그래프 데이터를 활용하여 사용자에게 적합한 방문지를 추천할 수 있다.

3.6 기존 방문지 입력 후 PoI 도출

GNN을 기반으로 사용자에게 PoI를 추천할 수 있는 서비스를 구현하였다. 그림 4는 서비스를 실행한 결과이다.

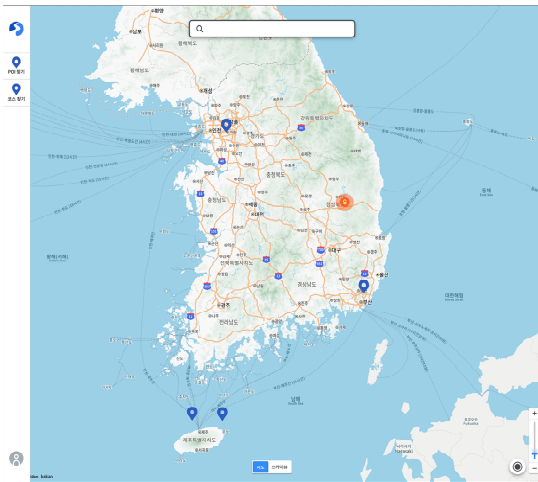


그림 4. 사용자 인터페이스
Fig. 4. User interface

화면에 보이는 결과 폼에서 자신이 다녀온 곳을 검색해 찾아 추가하면, 화면 측에서 카카오맵 API로부터 정보를 백엔드로 위도, 경도, 주소 등을 넘겨 데이터베이스에 저장한다. 그 후 탐색 옵션 설정 창에서 동행자 수를 설정 후 로그인 시 받아왔던 토큰들을 함께 백엔드로 전송하고 백엔드에서 해당 데이터들과 사용자의 과거 방문지들을 데이터베이스에서 조회하여 모두 요청 헤더를 통해 AI 서버로 요청한다. 결과적으로, 사용자는 GNN을 통해 그림 6과 같이 PoI를 받아보게 된다. 이를 통해 사용자는 보다 효율적으로 자신에게 맞는 장소를 찾을 수 있게 된다.

IV. 결론 및 향후 과제

본 연구에서는 장소에 대한 사용자의 요구 및 선호도 예측 방법을 GNN을 활용하여 사용자의 방문 및 상호작용 패턴 데이터를 기반으로 개인화된 관심 지점인 PoI를 추천하는 시스템을 제안하고 구현하였다. 다양한 GNN 아키텍처를 사용하여 효과적인 PoI 추천 모델을 개발하였으며, 실제 국내 여행 로그 데이터를 활용하여 모델의 실제 적용 가능성을 검증하였다. 사용자 패턴을 개인별 그래프 구조로 모델링하여 GCN, GAT, GraphSAGE, ChevConv Layer, 멀티-헤드 어텐션 기법 등의 방법을 그래프 신경망에 적용하여 PoI 추천 성능을 향상시켰다.

M1, M2, NGCF 및 그에 따른 Fine-Tuning 모델을 구현하고, 각각의 성능을 비교 분석을 수행하여 최종적으로 성능 향상을 위한 다양한 접근 방식 제시와 PoI를 추천하는 NGCF 기반 Fine-Tuning 모델을 개발하였다.

그러나, 현재 본 연구는 국내 여행 로그 데이터만을 사용하고 학습 데이터를 구축하는 데 있어서 특정 지역에 대한 데이터의 불균형 문제에 대한 한계가 있다. 한계점을 극복하기 위한 향후 과제로는 다양한 지역 및 국가의 데이터를 추가하여 모델의 일반화 능력을 향상하고 사용자로부터 받은 피드백과 추가적인 정보를 실시간으로 모델에 적용하여 추천 시스템의 적응성과 정확성을 개선하고자 한다.

References

- [1] F. Ricci, L. Rokach, and B. Shapira, "Recommender systems handbook", Recommender Systems Handbook, pp. 1-35, Jan. 2012. https://doi.org/10.1007/978-0-387-85820-3_1.
- [2] D. Jannach, M. Zanker, M. Ge, and M. Groning, "Recommender systems in computer science and information systems—a landscape of research", E-Commerce and Web Technologies, Vol. 123, pp. 76-87, Sep. 2012. https://doi.org/10.1007/978-3-642-32273-0_7.
- [3] Y. J. Lee and J. W. Kim, "A Study on

- Optimization of Machine Learning-based POI Recommendation System", Journal of Korean Institute of Information Technology, Vol. 19, No. 11, pp. 1-10, Nov. 2021. <https://doi.org/10.14801/jkiit.2021.19.11.1>.
- [4] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks", IEEE transactions on neural networks and learning systems, Vol. 32, No. 1, pp. 4-24, Mar. 2020. <https://doi.org/10.1109/TNNLS.2020.2978386>.
- [5] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural graph collaborative filtering", Proc. of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval, Paris France, pp. 165-174, Jul. 2019. <https://doi.org/10.1145/3331184.3331267>.
- [6] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks", arXiv preprint arXiv:1609.02907, Sep. 2016. <https://doi.org/10.48550/arXiv.1609.02907>.
- [7] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks", arXiv preprint arXiv:1710.10903, Oct. 2017. <https://doi.org/10.48550/arXiv.1710.10903>.
- [8] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering", Proc. of the 30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain, pp. 3844-3852, Dec. 2016.
- [9] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive Representation Learning on Large Graphs", Proc. of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, USA, pp. 1024-1034, Dec. 2017.
- [10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need", Proc. of

the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, California, USA, pp. 5998-6008, Dec. 2017.

- [11] J. E. Song, S. B. Kim, H. J. Kim, and S. J. Jo, "Recommender System Techniques: A Review and Analysis", Journal of the Korean Institute of Industrial Engineers, Vol. 41, No. 2, pp. 185-208, Apr. 2015. <https://doi.org/10.7232/JKIIIE.2015.41.2.185>.

저자소개

권혁찬 (Hyeokchan Kwon)



2025년 2월 :

국립경국대학교(국립안동대학교)
멀티미디어공학과(공학사)
관심분야 : 기계학습, 딥러닝,
클라우드 컴퓨팅

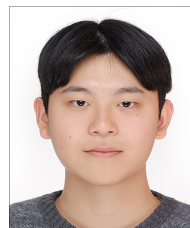
이유진 (Yujin Lee)



2025년 2월 :

국립경국대학교(국립안동대학교)
멀티미디어공학과(공학사)
관심분야 : 웹 개발, 클라우드
컴퓨팅

홍윤기 (Yunki Hong)



2020년 3월 ~ 현재 :

국립경국대학교(국립안동대학교)
멀티미디어공학과 학사과정
관심분야 : 백엔드 서버, 클라우드
컴퓨팅

김 유 진 (Yujin Kim)



2022년 3월 ~ 현재 :
국립경국대학교(국립안동대학교)
멀티미디어공학과 학사과정
관심분야 : 클라우드 컴퓨팅,
취약점 분석, 보안컨설팅

정 기 현 (Kihyun Jung)



2007년 8월 : 경북대학교
컴퓨터공학과(공학박사)
2022년 9월 ~ 현재 :
국립경국대학교(국립안동대학교)
소프트웨어융합학과 교수
관심분야 : 데이터보호,
스태가노그래피, 블록체인,

디지털위변조탐지기술, 보안프로그래밍