A Graph Representation Learning Approach for Imbalanced Ship Type Recognition Using AIS Trajectory Data

(IEEE Transactions on Intelligent Transportation Systems, 2025, Pan et al.)

Jaeyoung JUNG

wjdjjdud19@gmail.com

Contents

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

About

- Pan, Jiale, Rui Xin, Jian Yang, Fanlin Yang, Tingting Li, Bingchao Xu, and Fenli Jia. "A Graph Representation Learning Approach for Imbalanced Ship Type Recognition Using AIS Trajectory Data." *IEEE Transactions on Intelligent Transportation Systems* 26 (2025): 12049.

IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, VOL. 26, NO. 8, AUGUST 2025 12049

A Graph Representation Learning Approach for Imbalanced Ship Type Recognition Using AIS Trajectory Data

Jiale Pan, Rui Xin, Jian Yang, Fanlin Yang, Tingting Li, Bingchao Xu, and Fenli Jia

CATEGORY
ENGINEERING, CIVIL

5/184

JCR YEAR	JIF RANK	JIF QUARTILE	JIF PERCENTILE
2024	5/184	Q1	97.6
2023	5/182	Q1	97.5

CATEGORY
ENGINEERING, ELECTRICAL & ELECTRONIC

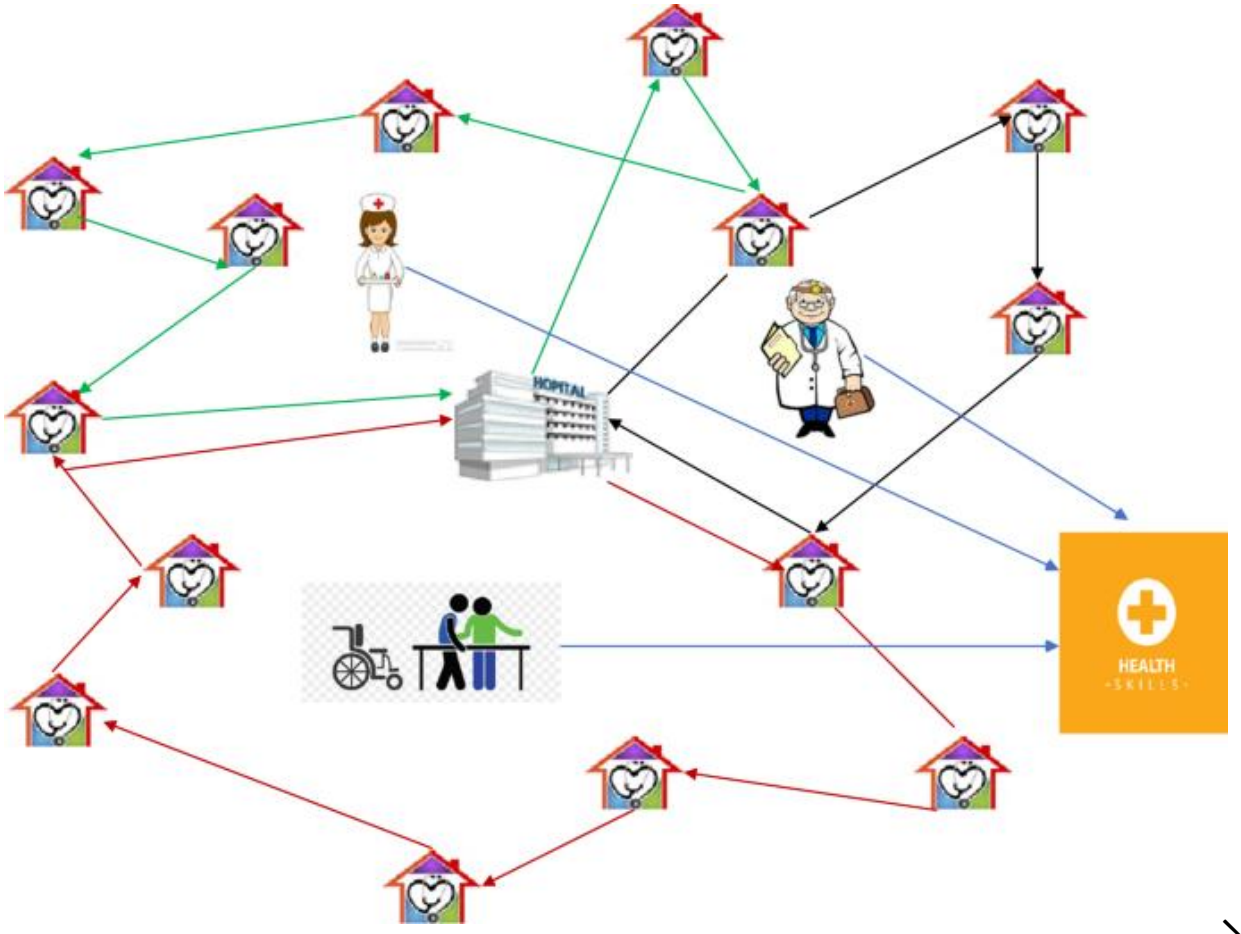
26/368

JCR YEAR	JIF RANK	JIF QUARTILE	JIF PERCENTILE
2024	26/368	Q1	93.1
2023	23/353	Q1	93.6

JCR 2024 **Top 2.72% / Q1**

Scopus 2024 **Top 1.50% / Q1**

About



변형된 VRP
HHCRSP

Time window

Skill matching

Synchronization

⋮

NP-Hard

→ **GNN**으로 구조적 관계 학습

J. Euchi, S. Zidi, and L. Laouamer, "A hybrid approach to solve the vehicle routing problem with time windows and synchronized visits in-home health care," *Arabian Journal for Science and Engineering*, vol. 45, pp. 10637–10652, 2020.

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Keyword

AIS (Automatic Identification System)

- **선박자동식별시스템**으로 선박의 항해안전 및 보안강화를 위하여 **선박의 이름, 제원, 속력, 위치 등의 정보**를 무선통신을 통하여 선박-선박, 선박-육상간 자동 송수신할 수 있는 항해 장비
- 선박의 충돌을 방지하기 위한 도구로 개발되었으며, **해양사고 발생 시 수색, 구조 등을 지원하는 시스템**

구분	정보	비고
정적정보 (static)	- IMO 번호 (MMSI 번호) - 호출부호 및 선명 - 선박의 종류, 길이, 폭, 너비 - 안테나의 위치	최초 또는 변경 사항 발생 시 수동 입력
동적정보 (dynamic)	- 선박의 위치 - UTC 시간 - 대지 침로 - 대지 속력 - 항해 상태(항해, 정박 등) - 선회율, 경사 각도	선박의 항해 상태에 따라 자동 입력
항해정보	- 선박의 흘수(draft, sea gauge) - 목적지 및 도착예정시간 - 위험화물	주기적 수동입력

- **DATA의 오류 발생**
  - **중복 오류, 인적 오류**
    - ✓ 사람의 실수로 발생
  - **위치 오류**
    - ✓ GPS 장비의 고장으로 발생

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Keyword

- **GNN (Graph Neural Network) : 그래프 신경망**
  - **노드**(Node)와 **엣지**(Edge)로 구성된 **그래프 구조**의 데이터를 직접 **처리**하는 딥러닝 모델
  - **Messeage passing** 과정을 통해 주변 노드의 정보를 **집계**하여 **자신의 표현을 업데이트**
    - **GCN, GraphSAGE, GAT** 등이 있음

$$h_v^0 = x_v$$

초기 임베딩 값 (노드 피처)

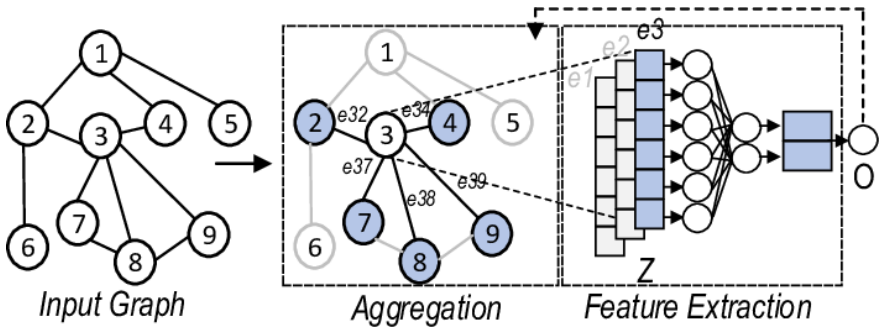
$$h_v^k = \sigma\left(W_k\left(\sum_{u \in N(v)} \frac{h_u^{k-1}}{|N(v)|} + B_k h_v^{k-1}\right)\right), \quad \forall k > 0$$

비선형 함수 적용

Average로 집계

자기 Feature 정보

Propagation step으로 **Neighbor Node의 정보를 받아**
자기의 **Node 정보 업데이트**



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Keyword

- **GIN(Graph Isomorphic Network)**
  - **GNN 중에서 노드 구조를 강하게 구분할 수 있도록 만든 모델**
    - "How Powerful are Graph Neural Networks?" (Xu et al., ICLR 2019) 에서 발표
  - **이웃정보를 aggregation하는 법이 SUM**
    - **GCN**은 **Mean**으로 집계 → {1,2} 와 {1,1,2,2} 를 구분 못함 → 단사성 만족 X 1.5
    - **GraphSAGE**는 **Mean, Max**로 집계 → {1,2,3} 과 {1,3} 을 구분 못함 → 단사성 만족 X 3
    - **GIN**은 **SUM**으로 집계 → 두 경우 모두 구분 가능 → 단사성 만족 O
  - **WL 테스트 (Weisfeiler–Lehman test)**
    - **그래프 두 개가 같은 구조(동형) 인지 확인하는 Test**
    - **기존 GNN의 집계 (Mean)** → 노드 A: 자기 피처=1, 이웃={2, 3}; 노드 B: 자기 피처=1, 이웃={1, 4} → 2.5로 같아짐
    - **WL의 해시 함수** → A → hash(1, {2,3}) = "X"; B → hash(1, {1,4}) = "Y" → 다름
      해시 함수 (단사성 보장)

집계 함수가 단사 함수여야 WL TEST와 동등한 표현 가짐

집계 함수는 SUM으로 구현하고 변환 함수는 MLP로 근사하자

$$h_v^{(k)} = \mathrm{MLP}^{(k)}\left((1+\epsilon^{(k)})h_v^{(k-1)} + \sum_{u\in N(v)} h_u^{(k-1)}\right)$$

단사성을 달성

자기 Feature 정보

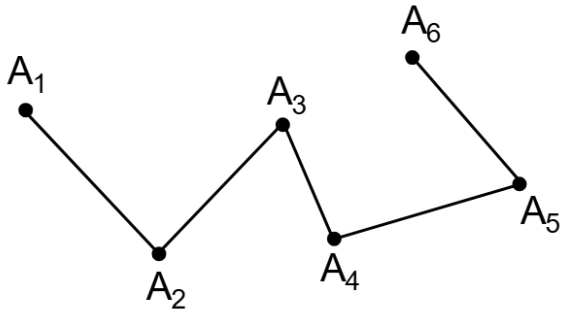
Sum으로 집계

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

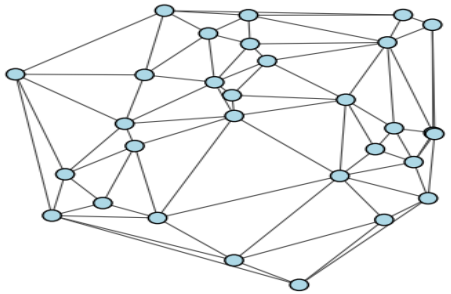
Keyword

- **궤적 그래프 모델링**

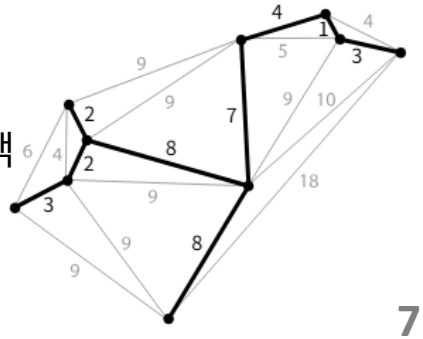
- **TPG** – (Trajectory Polyline Graph)
  - 시간 순서대로 앞뒤 포인트하고만 연결
  - 공간적 근접성은 고려 X



- **CDTG** – (Constrained Delaunay Triangulation Graph)
  - Delaunay 삼각분할로 만든 그래프로 **공간 포인트들을 삼각형으로 연결**
  - 3KM 이상 떨어진 노드 간 엣지는 제거



- **MSTG** – (Minimum Spanning Tree Graph)
  - CDTG에 Minimum Spanning Tree 를 적용하여 꼭 필요한 최소한의 엣지만 선택
  - 선택기준이 시간 순서가 아닌 최단 거리



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Keyword

- **Ship type recognition**
  - 인식 대상 선박 6종
    - 화물선(Cargo)
    - 유조선(Tanker)
    - 레저선(Pleasure)
    - 여객선(Passenger)
    - 어선(Fishing)
    - 예인선(Tug)

화물선 (Cargo)

?

유조선(Tanker)

✓ 동일 항구에서 같은 항로 공유
✓ 직선 항로, 느린 속도라는 유사한 패턴
✓ 정박/앵커링 행동 유사
✓ 선박 크기(정적 정보)도 비슷

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Introduction

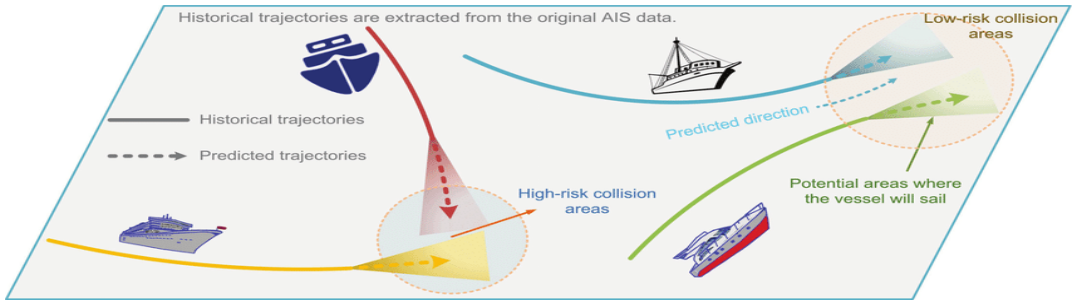
▪ 선박 유형 파악의 필요성

- **해상 운송**은 **전 세계 해외 무역의 90%를 차지**하는 물류 핵심 인프라
- 최근 항만 교통수단의 종류와 수가 급증하면서, 교통 관리 조율에 어려움 발생
- **선박 유형 정보는 항만 관리, 항행 안전, 해양 감시** 등에서 중요한 역할

▪ 기존 방법의 한계

- **기존 ML 방법**은 수작업 feature engineering에 의존 하여 궤적 내 잠재적 구조 패턴을 놓침
- **CNN 기반 방법**은 raster image로 변환하는 과정에서 위치 정밀도와 위상구조 정보가 손실
- **RNN 기반 방법**은 궤적의 시간 순서는 보지만, topology 관계를 충분히 반영 X
- **기존 GNN 방법**은 비단사성 문제와 Oversmooting 현상으로 인해 노드 표현이 비슷해지는 문제 발생
- 클래스 별 분균형 문제를 Over/UnderSampling으로 해결하여 원본 데이터 분포 왜곡

» **Node의 구조적 관계를 보존하면서 AIS 데이터의 불균형 문제까지 고려할 수 있게 GIN을 사용하면서 선박 유형 인식**



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Introduction

- **Q. raw AIS 데이터에 Ship type이 있는데 왜 예측할까?**
  - Raw AIS 데이터의 Ship Type 정보에 많은 오류가 있음을 확인

신뢰 가능한 label로 보기 어려움

예시 사례 : 북한 선박의 AIS 조작

  - → Ship Type 정보를 그대로 정답으로 사용하지 않고, MarineTraffic·VesselFinder와 대조하여 label을 재정의

- 운항 패턴과 궤적 구조를 학습하여 선박 유형 분류가 해당 연구의 CONTRIBUTION

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Related Works

- **Machine Learning** (Random Forest, Logistic Regression, Decision Tree)
  - 지리/행동/정적 속성을 **수작업으로 feature engineering**하여 인식 정확도 달성
  - 복잡한 피처 엔지니어링 필요, 단일 데이터셋에만 실험 → **일반화 능력 미검증, 과적합 위험**

- **Deep Learning** (MLP, **CNN**)
  - Machine Learning 보다 인식 정확도 향상
  - 궤적을 이미지로 변환 + CNN 적용 → **이미지 변환 과정에서 위상 구조 및 공간 정밀도 손실**

- **GNN의 ITS** (Intelligent Transportation Systems)(지능형 교통 시스템) **적용 및 한계**
  - GNN은 비유클리드 공간 데이터 처리에 강점 → 교통 예측, 패턴 인식에서 활발히 활용
  - 단, **불필요한 연산과 Oversmoothing** 으로 인해 **최종 예측 정확도** ↓

- **GIN 등장**
  - 기존 GNN 방식들의 핵심 문제를 해결
    - Over-smoothing: 레이어를 깊게 쌓을수록 모든 노드 표현이 평균으로 수렴 → **노드 간 구별 불가**
    - Over-smoothing 막으려고 복잡한 어텐션/정규화 추가 → 계산 비용 증가
  - (MLP)로 서로 다른 이웃 구성을 다른 표현으로 구별 → 표현력 이론적 보장

    단사함수
    집계방식

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Related Works

저자(연도)	방법론	그래프 구조	손실함수 가중치	불균형 처리	다중 데이터셋	주요 한계
Kraus et al. (2018)	Random Forest	X	X	X	X	수작업 피처, 일반화 능력 검증 X
Sheng et al. (2018)	Logistic Regression	X	X	X	X	수작업 피처, 일반화 능력 검증 X
Zhang et al. (2019)	EasyEnsemble + SMOTE	X	X	O	X	데이터 분포 왜곡
Ichimura et al. (2019)	MLP	X	X	X	X	공간 정보 손실
Liang et al. (2021)	CNN (래스터 이미지)	X	O	O	X	이미지 변환 시 공간 정밀도 손실
Li et al. (2021)	GNN	O	X	X	X	그래프 구조 설계 미흡, 표현력 한계
Eljabu et al. (2021)	GNN (이상 탐지)	O	X	X	X	표현력 한계, Over smoothing 위험
Pan et al. (2025)	**GIN + (TPG)**	**O**	**O**	**O**	**O**	-

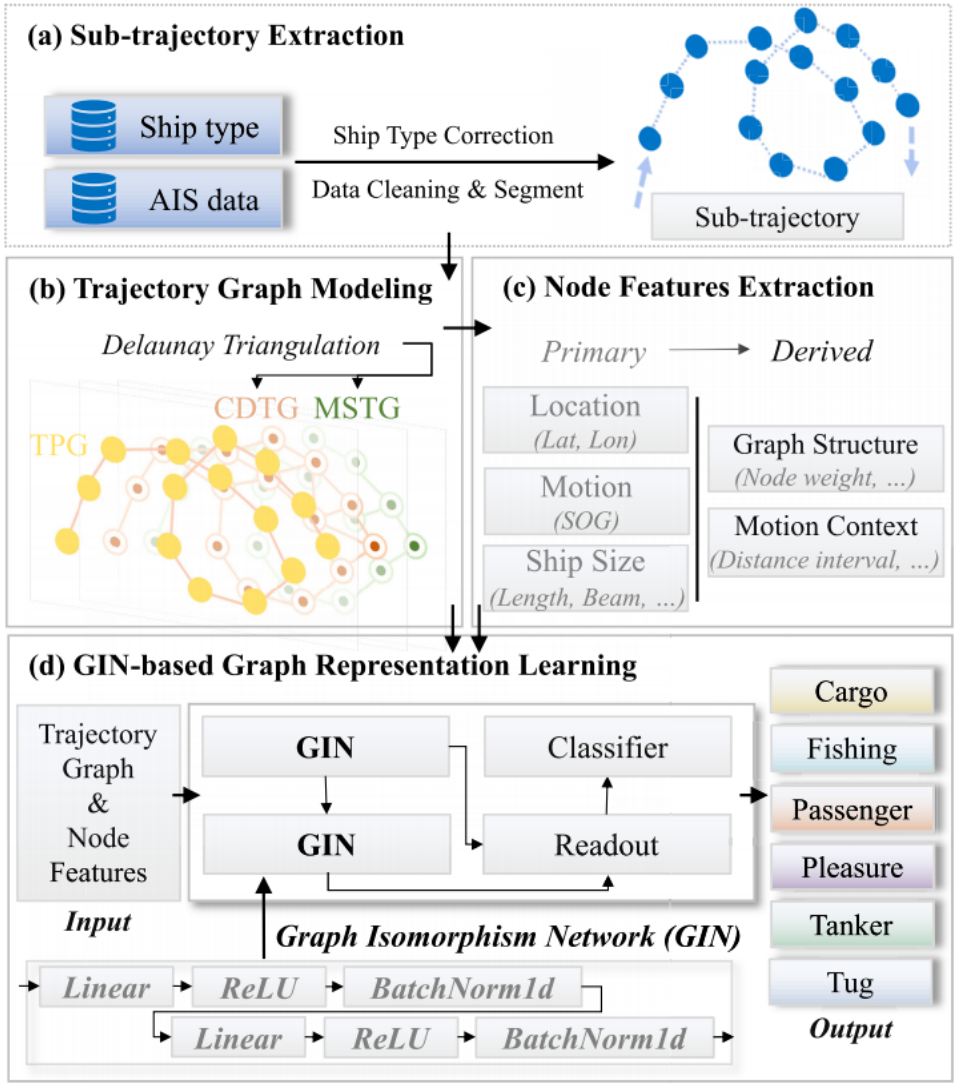
- 기존 연구들은 **그래프 구조, 불균형 처리, 일반화** 중 하나씩만 해결
- GIN + TPG 조합으로 **세 가지를 동시에** 해결

- GIN의 단사 집계 함수로 유사한 궤적 구별
- 손실 함수 가중치로 데이터 분포 왜곡 없이 불균형 문제를 해결
- 두 개의 독립적인 데이터셋에서 성능을 검증

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Methodology

▪ 제안 방법론



1. 서브 궤적 추출

(Sub-trajectory Extraction)

- 이상치와 결측 정보를 처리

2. 궤적 그래프 모델링

(Trajectory Graph Modeling)

- 궤적 그래프 구조를 구성

3. 노드 피처 추출

(Node Features Extraction)

- 기본 피처, 파생 피처 추출하여 계산

4. GIN 기반 그래프 표현 학습

(GIN-based Graph Representation Learning)

- 그래프와 노드를 입력받아 예측한 선박 유형 출력

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Methodology

▪ 서브 궤적 추출 (Sub-trajectory Extraction)

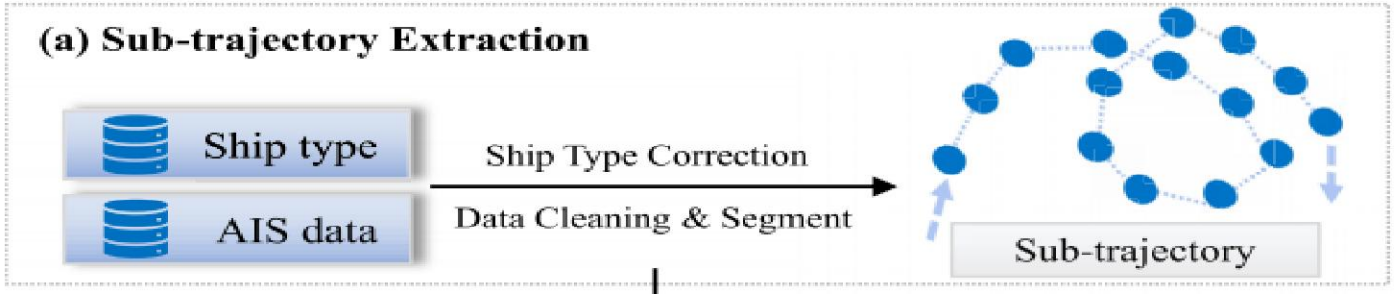
- 목적 : 고품질 선박 이동 궤적 데이터를 원시 AIS 데이터로부터 추출

- Ship Type Correction :
  - 샘플 수가 너무 적은 선박들은 학습 대상에서 제외
  - 최종 6개 유형 선정 : 화물선, 유조선, 레저선, 여객선, 어선, 예인선

- AIS data 포함 정보 : MMSI (선박 식별번호) / Basedatetime (시각) / 위도·경도 / SOG (속도) / COG (각도) / 선박 길이·폭

- 처리 절차

  1. 이상 궤적 포인트 제거 : 시공간 정보의 정확성을 보장
     - ✓ SOG > 35knots 포인트 제거
     - ✓ 전후 포인트와의 거리가 모두 3km 이상인 포인트 제거
  2. 이상 서브 궤적 제거 : 정보량의 불균형을 예방
     - ✓ 일(day) 단위로 궤적 분할
     - ✓ 포인트 수 100개 미만(항해 5분 미만) 서브궤적 제거
     - ✓ SOG > 2knots 인 포인트 비율이 30% 미만이면 서브궤적 제거



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Methodology

▪ 궤적 그래프 모델링 (Trajectory Graph Modeling)

- 목적 : 그래프 내에서 유사한 노드들 사이의 메시지 전달을 강화하는 것
  - **Message passing 할 때** 유사한 노드들 한테서만 정보를 받아오도록 그래프 구조를 설계

- 그래프 구조 설계 조건

1. 순차적 의존성 보존
   - ✓ 궤적의 **시간 순서 관계를 유지**하여 실제 선박 활동 복원
2. 불필요한 엣지 제거
   - ✓ 노드간 차이가 크면 그 사이 엣지 제거
   - ✓ 계산 효율 향상 + 연결 구조의 영향 분석 용이
3. 유사 노드 간 연결 강화
   - ✓ 노드 자신과 이웃 노드 간 유사할 경우 노드 간 연결 엣지를 더 많이 생성

- 설계 조건을 만족하며 궤적의 순차적 의존성을 고려하여 3가지 그래프 구조 비교

구조	설명
TPG	궤적 포인트를 순서대로 연결 (시간 순 체인)
CDTG	DTG에서 거리 임계값(3km) 초과 엣지 제거
MSTG	Delaunay 삼각분할 기반 최소 신장 트리

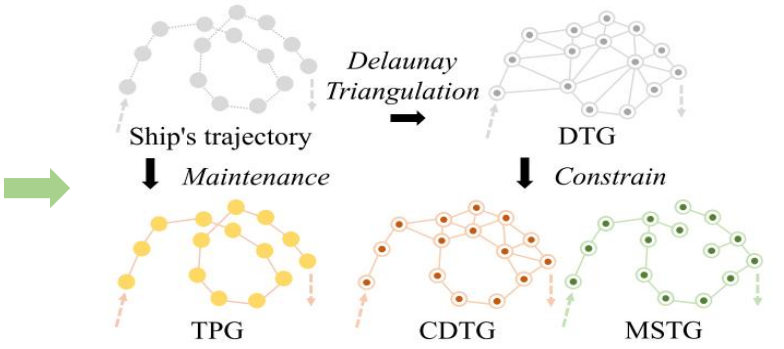


Fig. 2. DTG, TPG, CDTG and MSTG graph structures of a ship trajectory.

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Methodology

▪ 노드 피처 추출 (Node Features Extraction)

- 목적 : 궤적 포인트(노드)에 선박 유형 인식에 적합한 피처를 부여 → GNN 학습 품질 향상

- **Primary features** – (원시 AIS 데이터에서 추출)

카테고리	피처	설명
Location	위도, 경도	선박 활동의 공간적 패턴 표현
Motion	SOG, COG	선박 유형별 속도·방향 차이 반영
Size	길이, 폭, 종횡비	대형/소형 선박 구분

- **Derived features**

카테고리	피처	설명
Graph structure	노드 차수, 노드 가중치	동일 위치 반복 출현 횟수
Motion Context	시간 간격, 거리 간격, 방위 차이	포인트 간 미세 행동 패턴

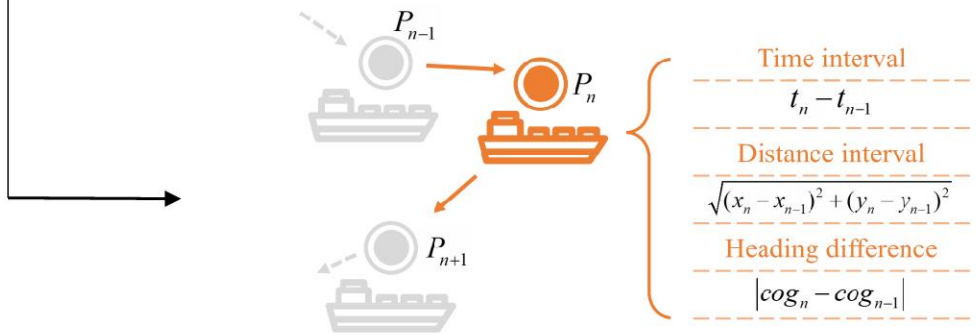
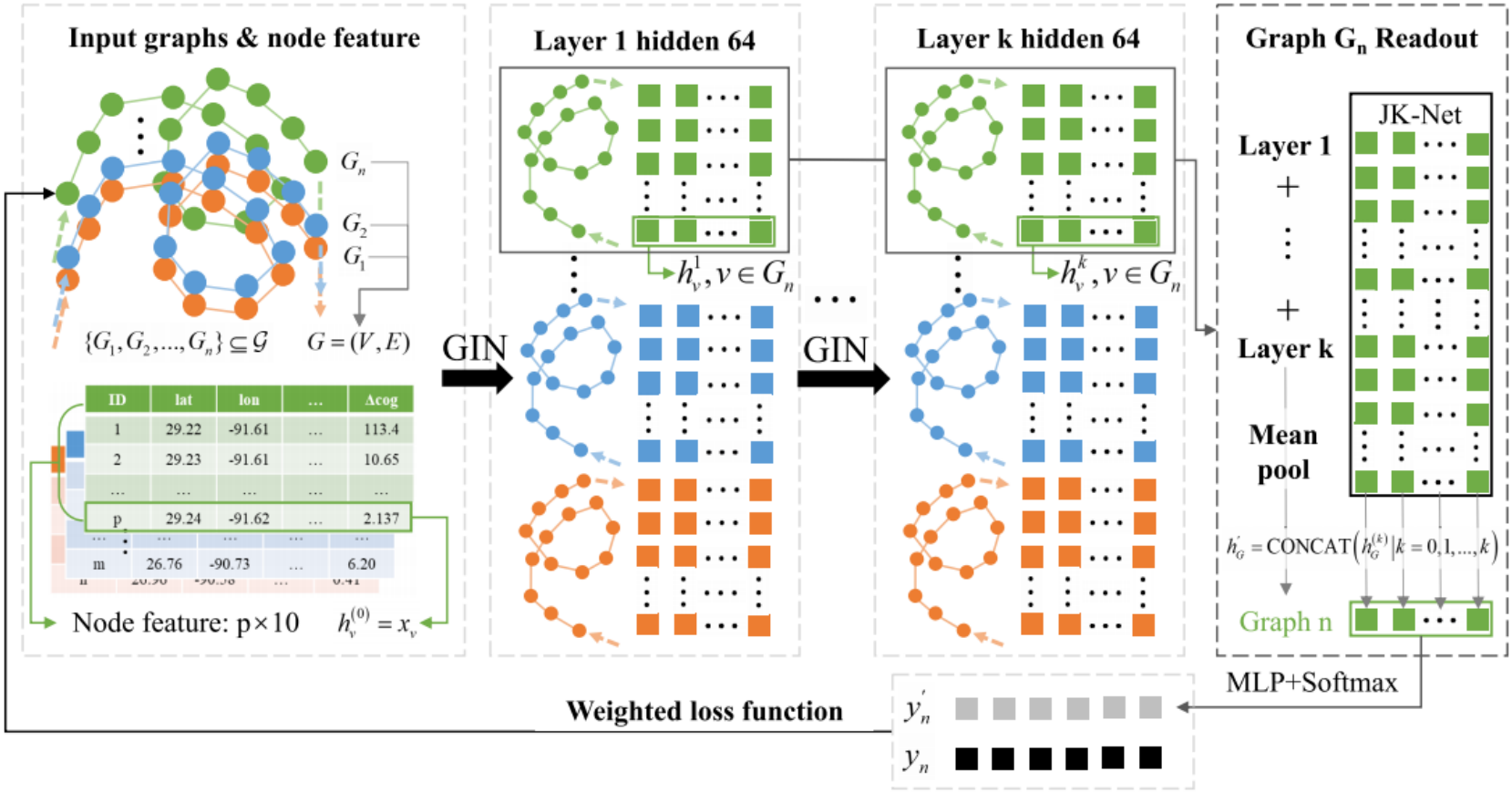


Fig. 3. Motion context features of network nodes.

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Methodology

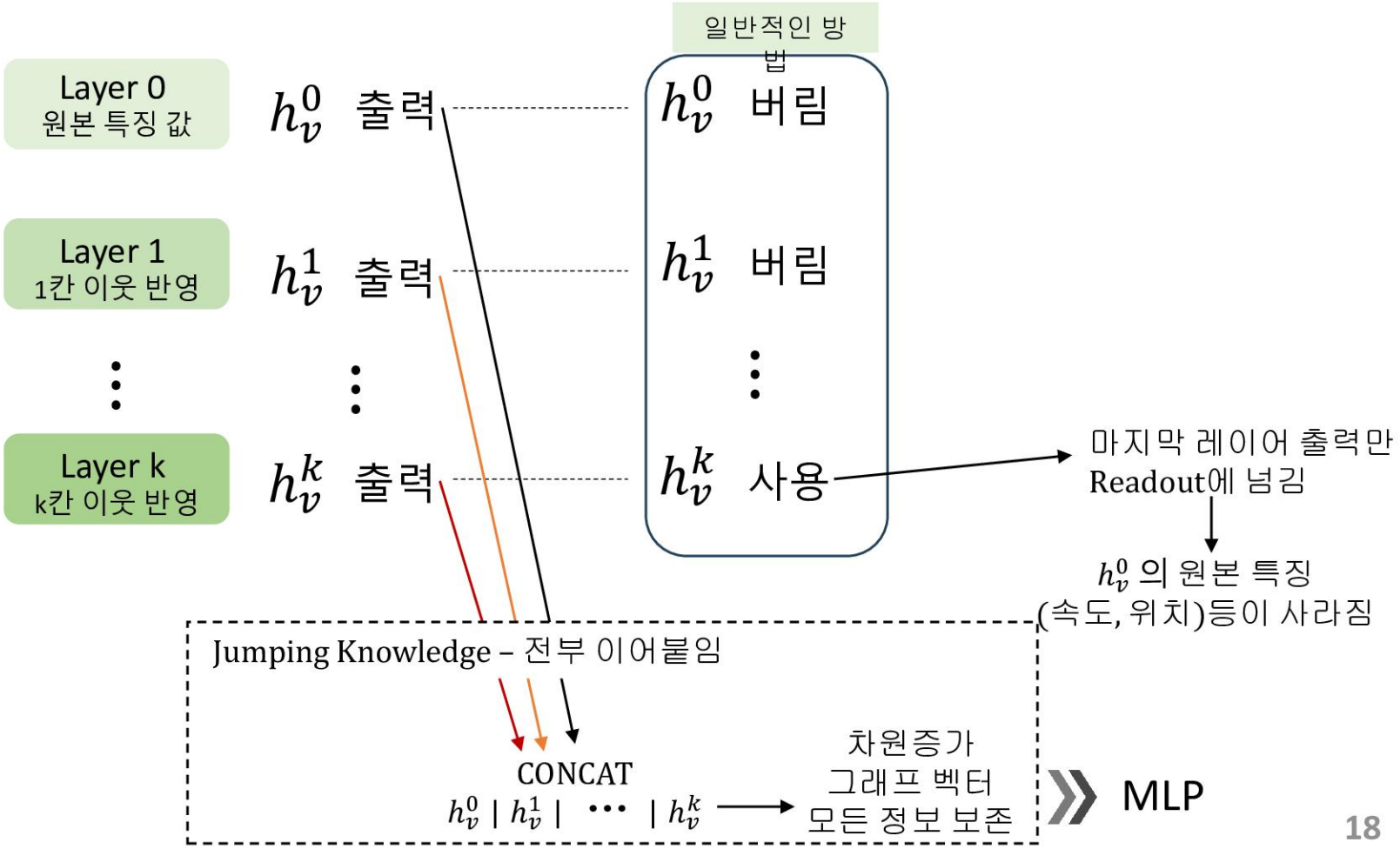
- **GIN 기반 그래프 표현 학습 (GIN-based Graph Representation Learning)**
  - **전체 흐름** : 궤적 그래프 입력 → GNN 노드 표현 학습 → Jumping Knowledge로 전 레이어 집계 → MLP + Softmax → 선박 유형 출력



Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Methodology

- **모든 레이어에서 Readout**
  - Jumping Knowledge로 모든 레이어에서 출력 정보를 활용 → Over-smoothing 해결
  - 전체 그래프 표현 h_G 를 생성



Methodology

▪ Weighted Loss Function

- 클래스 불균형 데이터
  - 예인선 47.%
  - 여객선 3.6%
- 예인선만 잘 맞추는 방향으로 편향 가능성 ↑
- 가중치 공식 : $W_i = \frac{N}{ni}$ → 예인선 (47.5%) → Wi = 2.1 ← 낮은 가중치
여객선 (3.6%) → Wi = 27.5 ← 높은 가중치
- 기존 Cross-Entropy 손실
  - $L = -(1/N)\,\Sigma\, y * log(y)$
- **가중치를 고려한 Cross-Entropy 손실**
  - $L = -(1/N)\,\Sigma\, y * log(y) \times Wi$ → 여객선 하나 틀리면 27.5가 곱해짐

✓ 데이터 분포를 그대로 유지
✓ 샘플 손실 X
✓ 일부 클래스에 편향 학습 방지

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Experiments

실험 데이터셋

- 인식 대상 6종 : 화물선, 유조선, 레저선, 여객선, 어선, 예인선

데이터셋	지역	서브궤적 수	클래스 분포
Gulf of Mexico	북부	27,270개	예인선 47.5% / 여객선 3.6%
New Jersey Bight	동부	~13,000개	상대적으로 균등

실험 설정

- 5-fold 교차검증 (학습:검증:테스트 = 3:1:1)
- 학습률 : 0.002
- 배치 크기 / Epoch : 128/400

그래프 구조 비교

- **학습시간, 추론속도 TPG가 가장 좋음**
- **정확도와 손실 TPG가 가장 우수**

TABLE III

COMPARISON OF PERFORMANCE AND EFFECTIVENESS OF DIFFERENT GRAPH STRUCTURES

	Graph Structure	Avg nodes	Avg edges	Train time (s/epoch)	Inf speed (graph/s)	Avg loss	Avg accuracy
Gulf of Mexico	CDTG	580.96	785.51	3.75	8,623	0.1793	0.942±0.006
	MSTG		289.14	3.66	9,278	0.1704	0.943±0.004
	TPG		289.98	3.39	10,036	0.1620	0.948±0.004

TABLE III

COMPARISON OF PERFORMANCE AND EFFECTIVENESS OF DIFFERENT GRAPH STRUCTURES

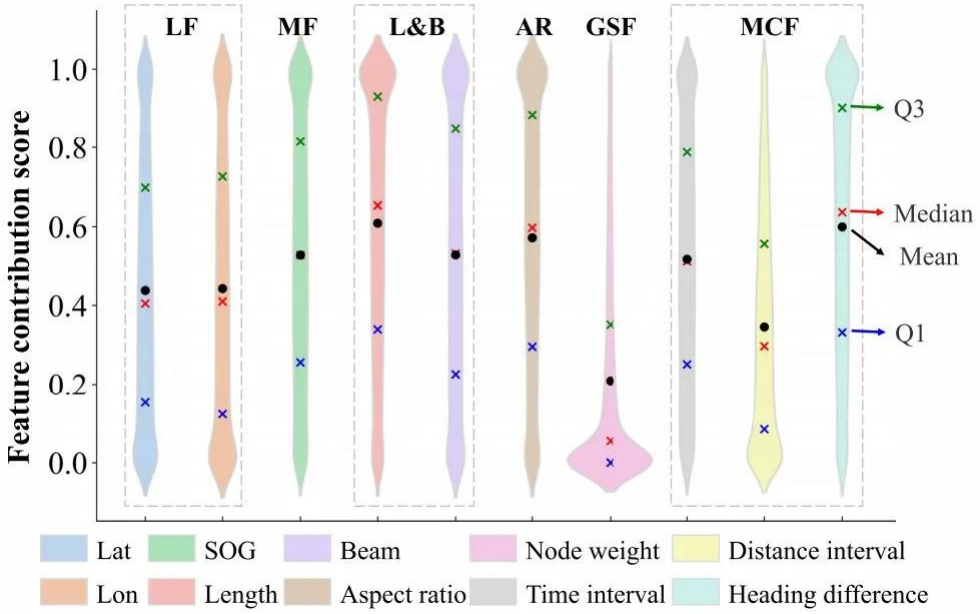
	Graph Structure	Avg nodes	Avg edges	Train time (s/epoch)	Inf speed (graph/s)	Avg loss	Avg accuracy
New Jersey Bight	CDTG	479.39	656.04	1.71	9,815	0.1688	0.944±0.003
	MSTG		239.05	1.57	11,427	0.1732	0.944±0.006
	TPG		239.19	1.49	11,726	0.1655	0.947±0.003

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Experiments

- **피처 기여도 (GNN Explainer)**

피처	기여도
Heading difference	높음 – 선박 유형별 조향 패턴 차이 학습, **가장 높은 기여도**
선박 크기 (길이·폭), 종횡비, SOG	높음 – 대형/소형 선박을 구분하는 1차적인 피처
위도·경도	변동성 큼 – 예측에 불안정
Node weight, Distance interval	가장 낮음



Experiments

▪ 모델 성능 비교

- 비교 실험 설계
  - 그래프 구조 → TPG로 동일
  - Readout 구조 → Mean aggregation으로 동일
  - 손실 함수 → 가중 Cross-Entropy로 동일
  - 하이퍼파라미터 → 학습률, 배치 크기 등 모두 동일
  - Step decay 적용 → 빠르게 학습해서 정밀하게 수렴 (두 테스트셋 모두 95% 이상 정확도 달성)

- GCN, GAT, GraphSAGE, GIN 비교 실험
  - GCN/GAT/GraphSAGE보다 높음
  - 포인트를 절반이나 제거해도 100% 대비 1.5%밖에 안떨어짐. → 실제 AIS 데이터는 신호 누락이 빈번 이런 환경에서의 강건성 보장

TABLE VII

COMPARISON OF MODEL PERFORMANCE AND FRAMEWORK ROBUSTNESS (%)

Methods	Gulf of Mexico			New Jersey Bight		
	Accuracy	Precision	Recall	Accuracy	Precision	Recall
GCN	89.68	89.36	89.68	90.54	90.76	90.54
GAT	90.28	90.05	90.28	90.55	90.83	90.55
GraphSAGE	90.65	90.41	90.65	91.84	91.99	91.84
GIN (50%)	93.37	93.49	93.37	94.21	94.25	94.21
GIN (70%)	93.41	93.54	93.41	94.09	94.14	94.09
GIN (100%)	**95.02**	**95.10**	**95.02**	**95.21**	**95.24**	**95.21**

서브궤적 단위 정확도

멕시코만: 95.02%| 뉴저지 바이트: 95.21%

선박 단위 최종 정확도

멕시코만: 93.95%| 뉴저지 바이트: 92.33%

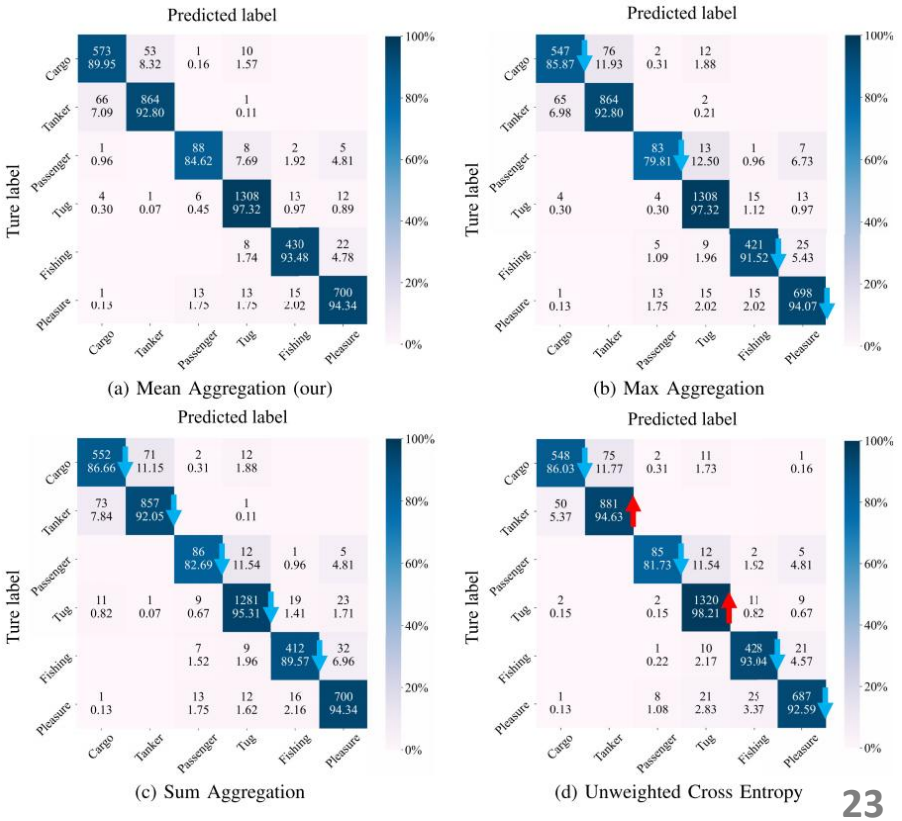
Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Experiments

- **Readout 함수 및 손실 함수 비교**
  - Mean > Max > Sum 순으로 성능 우수
  - Max: 여객선 79.81%, 화물선 85.87% → 극단값 강조로 미세 패턴 손실
  - Sum: 어선 89.57%, 예인선 95.31% → 노드 수 (궤적길이)에 따라 다른 궤적이 비슷한 표현으로 뭉개짐
  - Mean: 전체적으로 가장 균형잡힌 성능 → 최종 선택

- **Weighted Loss Function 효과**
  - 가중치 없으면 다수 클래스에 편향
  - 가중치 적용하면 소수 클래스 recall 향상

- **주의**
  - Sum 집계 – GIN 컨볼루션 내부
  - Mean 집계 – Readout 함수



(a) Mean Aggregation (our)

(b) Max Aggregation

(c) Sum Aggregation

(d) Unweighted Cross Entropy

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Code Review

```python
class GINWithJK(torch.nn.Module):
    def __init__(self, dataset, num_layers, hidden=64, mode='cat'):
        super().__init__()
        self.conv1 = GINConv(
            Sequential(
                Linear(dataset.num_features, hidden),
                ReLU(),
                BN(hidden),
                Linear(hidden, hidden),
                ReLU(),
                BN(hidden),
            ), train_eps=True)
        self.convs = torch.nn.ModuleList()
        for i in range(num_layers - 1):
            self.convs.append(
                GINConv(
                    Sequential(
                        Linear(hidden, hidden),
                        ReLU(),
                        BN(hidden),
                        Linear(hidden, hidden),
                        ReLU(),
                        BN(hidden),
                    ), train_eps=True))
        self.jump = JumpingKnowledge(mode)
        # self.jump = JumpingKnowledge(mode, channels=hidden, num_layers=num_layers)
        if mode == 'cat':
            self.lin1 = Linear(num_layers * hidden, hidden)
        else:
            self.lin1 = Linear(hidden, hidden)
        self.lin2 = Linear(hidden, dataset.num_classes)

    def reset_parameters(self):
        self.conv1.reset_parameters()
        for conv in self.convs:
            conv.reset_parameters()
        self.jump.reset_parameters()
        self.lin1.reset_parameters()
        self.lin2.reset_parameters()
```

첫 번째 GIN 레이어 정의

노드 특징(10차원)을 hidden(64차원)으로 변환
Linear → ReLU → BN 2번 반복하여
개별 특징 재표현 + 특징 간 조합 패턴까지 동시에 학습
train_eps=True를 통해 자기 자신 표현에 가중치를 줌

나머지 레이어 반복 생성

2번째~마지막 레이어 반복 생성
Hidden 64차원은 64->64 차원 그대로 유지
레이어 마다 이웃 범위 1hop씩 증가

mode='cat': **모든 레이어 출력 CONCAT**
lin1: 레이어 갯수× 64 → 64 (**CONCAT으로 커진 차원 축소**)
lin2: 64→6 (**선박 유형 수로 축소**)

레이어 가중치 초기화

모든 레이어의 학습된 가중치를 초기값으로 리셋
5-fold Cross Validation에서fold마다 새로 학습할 때 호출

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Code Review

```python
def forward(self, data):
    x, edge_index, batch = data.x, data.edge_index, data.batch
    x = self.conv1(x, edge_index)
    xs = [x]
    for conv in self.convs:
        x = conv(x, edge_index)
        xs += [x]
    x = self.jump(xs)

    # x = global_add_pool(x, batch)
    x = global_mean_pool(x, batch)

    x = F.relu(self.lin1(x))
    x = F.dropout(x, p=0.5, training=self.training)
    x = self.lin2(x)
    return F.log_softmax(x, dim=-1)

def __repr__(self):
    return self.__class__.__name__
```

데이터 입력
x: **노드 특징 행렬** (노드수×10) ,
edge_index: TPG **엣지 연결 정보**,
batch: **노드가 어느 그래프 소속인지**

jump(xs)로 h^1, h^2, h^3, h^4 저장, 전부 CONCAT → 256차원
over-smoothing 해결
Mean Pool을 통해 모든 노드 표현을 평균내어 벡터로 압축

lin1()함수: 256 → 64 차원 축소
ReLU함수: 음수 제거 + 비선형 패턴 학습

lin2함수: 256 → 64 → 6차원 (선박 유형 수)
log_softmax를 통해 6개 점수(logit)를 확률로 변환
최대값 인덱스 = 최종 선박 유형

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Limitation & Future work

- **데이터 셋 범위의 한계**
  - 2023년 5월 단 한 달 데이터만 사용
    - 계절별 선박 운항 패턴 차이 반영 못함
    - 특정 지역(멕시코만, 뉴저지 바이트)에 국한
    - 다른 해역(아시아, 유럽 항로 등)에서의 일반화 능력 검증 안 됨
  - 데이터 셋의 부족
    - 뉴저지 바이트의 경우 데이터 수가 적어 100 epoch 이후 과적합 발생

- **Future work**
  - **Domain Adaptation 적용**
  - 멕시코만(Source) → 다른 해역(Target)
    - GIN 인코더로 특징 추출 후 MMD Loss로 두 도메인 분포 차이 축소
    - 해역에 관계없이 동작하는 공통 특징 공간 생성
  - 기대효과
    - 아시아·유럽 해역으로 일반화 가능

Prof. Hyerim Bae (hrbae@pusan.ac.kr)

Q & A

감사합니다.

정재영
wjdjjdud19@gmail.com