KNU

GAT 기반 웨어러블 라이프로그 데이터 분석을 통한 치매 고위험군 이상 경고 시스템

2024. 10. 09

목마른 감자들 - 강소연, 안성겸

목차

KNU

1. 연구 배경 및 목표 — 연구의 필요성 및 목표에 대한 설명

2. 데이터 소개 및 EDA — 데이터 셋 소개와 간략한 EDA 과정 설명

3. GAT 모델 설계 — 데이터 전처리 및 GAT 모델 설계

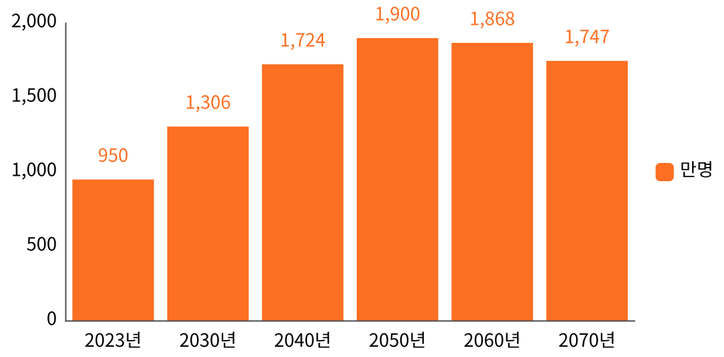
4. 분석 및 결론 — 결과 분석 및 결론

연구 배경

KNU

지속되는 고령화

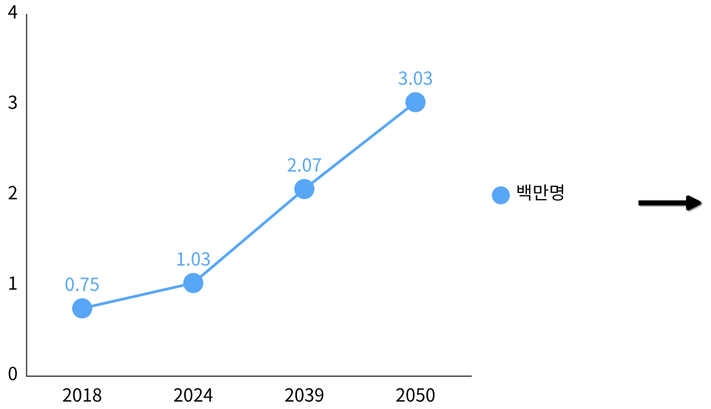
대한민국은 고령화 사회로 진입하고 있으며, 통계에 따르면 2025년에는 65세 이상의 인구 비율이 20.6%에 도달할 것으로 예상되고, 2050년에는 이 비율이 40%를 넘길 것으로 전망.



65세 이상 고령인구 전망

늘어나는 치매환자

전국 65세 이상 고령 인구 중 치매환자는 70만 5,473명으로 추정되며, 이는 10명 중 1명은 치매를 앓고 있는 것을 의미함. 이러한 국가 치매관리비용은 2019년 GDP의 약 0.8%, 1인당 연간 진료는 약 344만원에 달함.



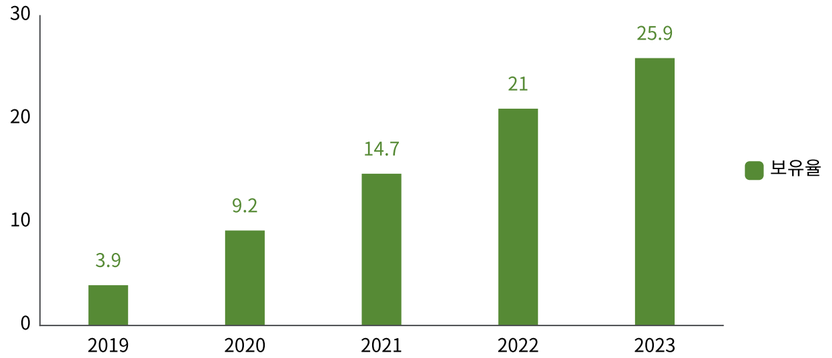
치매 환자 장례 추계

목표

KNU

웨어러블 기기 현황

웨어러블 기기는 손목 시계, 반지 등의 형태로 사용자의 신체에 착용하는 전자 장치로 웨어러블 기기 보유율은 2019년 3.9%에서 25.9%로 빠르게 증가함.

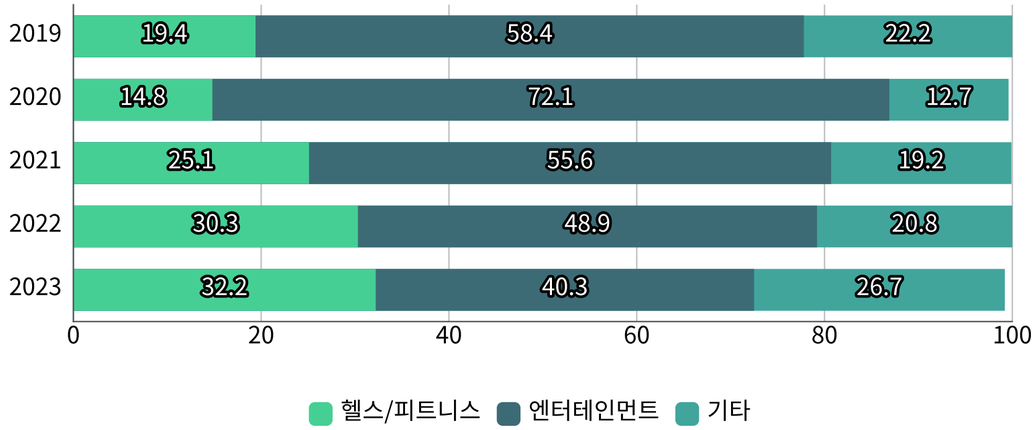


웨어러블 기기 보유율

기기 이용목적

초기에는 주로 엔터테인먼트 용도로 사용되던 웨어러블 기기가 시간이 지나며 헬스/피트니스 용도로 더 많이 활용되고 있으며, 이는 건강 관리에 대한 관심과 수요가 증가하고 있음을 보여줌.

따라서, 웨어러블 기기를 통해 고령자들의 데이터를 수집하고, 이를 분석하여 치매 고위험군을 예측하는 모델을 구축해 치매 고위험자 이상 경고 시스템을 개발하고자 함.



기기 이용 목적

데이터 소개

KNU

걸음걸이

웨어러블 디바이스를 통해
수집한 착용자의 활동 정보 데이터

수면

웨어러블 디바이스를 통해
수집한 착용자의 수면 정보 데이터

인지기능

MMSE 검사 도구로 평가된 연구 대상자의
인지 기능 데이터

EDA

KNU

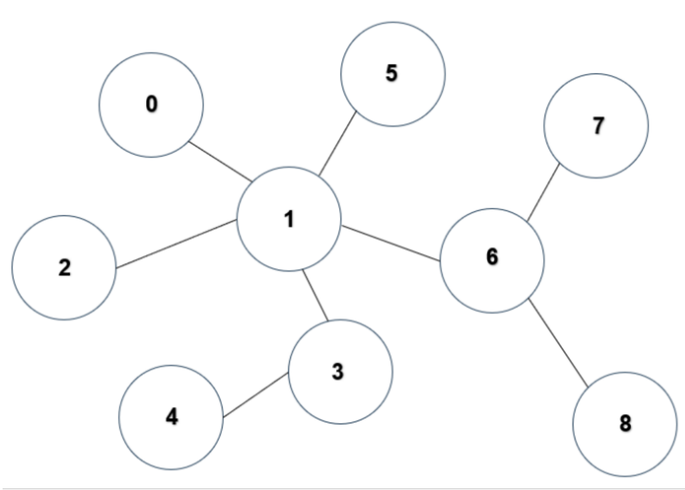
걸음걸이, 수면 데이터와 인지 기능 데이터의 수가 상이

걸음걸이, 수면 데이터의 경우 일 별로 수집된 데이터이지만 인지 기능의 경우 디바이스가 아닌 지필식 검사로 수집된 데이터이기 때문

걸음걸이와 수면 데이터를 시계열 모델(LSTM)이 아닌 각 시간대의 데이터를 각각의 새로운 데이터라고 정의해 모델을 만드는 것이 좋다고 판단

웨어러블 디바이스로 일정시간 동안 측정한 데이터들을 하나의 노드로 가정

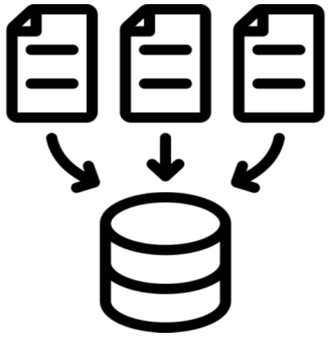
EMAIL	activity_av	activity_ca	activity_ca	activity_cl	activity_da	activity_da	activity_da	
nia+279@	1.28125	196	2251	...	3353	2020-10-2	2020-10-1	--- node0
nia+279@	1.25	145	2159	...	2516	2020-10-2	2020-10-2	--- node1
nia+279@	1.21875	118	2140	...	1716	2020-10-2	2020-10-2	--- node2
nia+279@	1.28125	180	2240	...	2791	2020-10-2	2020-10-2	
nia+279@	1.46875	374	2559	...	5393	2020-10-2	2020-10-2	
nia+279@	1.53125	436	2630	...	6860	2020-10-2	2020-10-2	
nia+279@	1.28125	159	2212	...	2433	2020-10-2	2020-10-2	
nia+279@	1.28125	159	2212	...	2365	2020-10-2	2020-10-2	
nia+279@	1.4375	407	2508	...	7288	2020-10-2	2020-10-2	
nia+279@	1.5	408	2587	...	6368	2020-10-2	2020-10-2	
nia+279@	1.25	191	2224	...	2950	2020-10-3	2020-10-2	
nia+279@	1.28125	154	2233	...	2138	2020-10-3	2020-10-3	
nia+279@	1.1875	74	2075	...	934	2020-11-0	2020-10-3	
nia+279@	1.15625	63	2039	...	776	2020-11-0	2020-11-0	
nia+279@	1.28125	143	2213	...	1947	2020-11-0	2020-11-0	
nia+279@	1.34375	204	2336	...	2767	2020-11-0	2020-11-0	
nia+279@	1.40625	329	2454	...	5611	2020-11-0	2020-11-0	
nia+279@	1.3125	207	2295	...	3362	2020-11-0	2020-11-0	
nia+279@	1.40625	392	2463	...	7154	2020-11-0	2020-11-0	
nia+279@	1.46875	381	2554	...	5810	2020-11-0	2020-11-0	
nia+279@	1.5625	431	2663	...	6410	2020-11-0	2020-11-0	
nia+279@	1.4375	392	2523	...	6306	2020-11-1	2020-11-0	
nia+279@	1.5	442	2616	...	7621	2020-11-1	2020-11-1	
nia+279@	1.46875	372	2531	...	5880	2020-11-1	2020-11-1	
nia+279@	1.25	162	2181	...	2216	2020-11-1	2020-11-1	
nia+279@	1.28125	201	2250	...	3408	2020-11-1	2020-11-1	--- node n



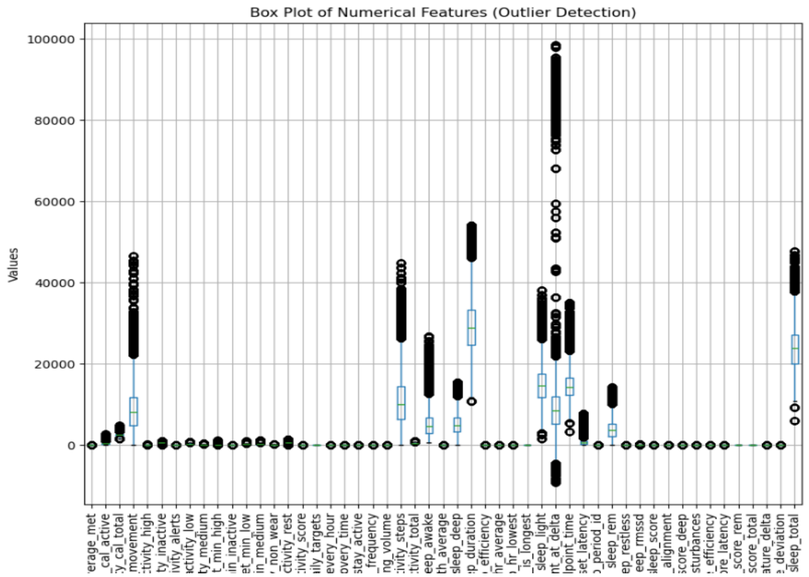
데이터 전처리 (1/3)

KNU

- 데이터는 웨어러블 기기로 측정한 걸음걸이, 수면 데이터와 MMSE 검사 도구로 평가된 인지기능 총 3가지 데이터로 구성
- 운동 시작 및 종료 시간과 같은 모델 학습에 영향을 미치지 않는 feature 제거
- 결측치와 데이터 중복값들이 포함된 feature 제거 후, 3가지 데이터를 하나의 데이터로 병합



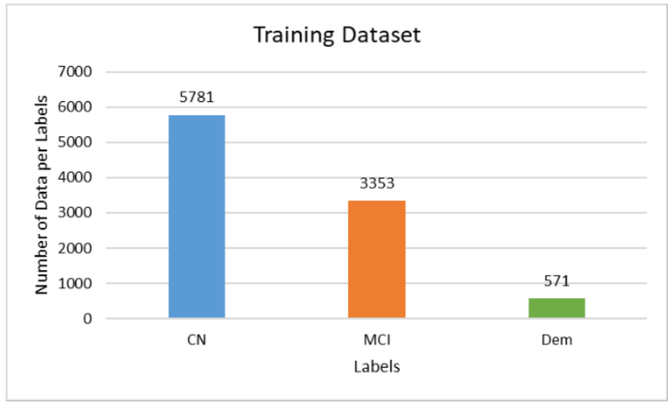
- 이후, 데이터를 시각화 한 결과 값들의 편차가 크고 넓게 분포함을 인지
- IQR을 사용해 Q1과 Q3를 기준으로 1.5배 넘는 값들을 이상치로 판별한 결과
  ->많은 이상치(좌측값과 우측값의 편차)가 있어 Z-score로 정규화 수행

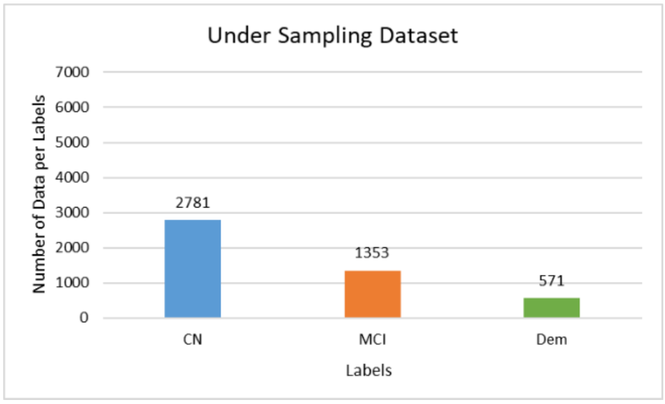


$$Z = \frac{x - \mu}{\sigma}$$

데이터 전처리 (2/3)

- Train 데이터의 클래스별 분포를 분석한 결과, CN 클래스의 데이터 수가 다른 클래스에 비해 최대 약 10배 많음을 확인
- 데이터의 편차가 심하면 모델의 과적합 및 일반화 성능 저하 발생으로 편향된 예측 수행
- 편차가 심한 문제를 해결하기 위해 **언더샘플링** 사용





CN: 인지 정상, MCI:경도 인지 장애, DEM: 치매

데이터 전처리 (3/3)

KNU

- 앞에서 전처리된 데이터에서 각 행은 노드를, 각 열은 해당 노드의 feature로 정의하여 Feature Matrix를 생성
- 노드 간의 관계를 정의하기 위해 **피어슨 상관 계수**를 사용해 노드 간의 유사성 측정 -> 상위 0.5%에 해당하는 노드 쌍을 선택하여 Adjacency Matrix 생성

Features

Nodes

	activity_average_met	activity_cal_active	activity_cal_total	activity_daily_movement
0	-1.088143	-1.145147	-0.931540	-1.137273
1	-0.928780	-1.018678	-0.707893	-1.086964
2	0.027402	-0.317677	0.172888	-0.610942
3	0.346129	-0.093646	0.368923	-0.342562
4	0.186765	-0.194821	0.250197	-0.432571
...	...	...	...	...
4700	0.664856	0.730211	0.705774	0.882252
4701	-0.450689	-0.682631	-0.337910	-0.764249
4702	-0.610053	-0.848848	-0.498052	-0.890847
4703	0.346129	0.390551	0.512499	0.425256
4704	-0.610053	-0.603136	-0.456636	-0.658142

Feature Matrix

→

$$r_{XY} = \frac{\sum_i^n \left(X_i - \overline{X}\right)\left(Y_i - \overline{Y}\right)}{\sqrt{\sum_i^n \left(X_i - \overline{X}\right)^2}\sqrt{\sum_i^n \left(Y_i - \overline{Y}\right)^2}}$$

피어슨 상관 계수 적용

→

	0	1	2	3	4	5	6	
0	1.000000	0.745040	0.132299	0.039592	0.354180	0.337087	0.448324	\
1	0.745040	1.000000	0.274094	0.369639	0.328238	0.648028	0.553701	
2	0.132299	0.274094	1.000000	0.549814	0.568611	0.265719	0.048596	
3	0.039592	0.369639	0.549814	1.000000	0.682991	0.590814	0.370141	
4	0.354180	0.328238	0.568611	0.682991	1.000000	0.267367	0.364055	
...	...	...	...	...	...	...	...	
4700	-0.458350	-0.613355	-0.103647	-0.373386	-0.213057	-0.622357	-0.390584	
4701	0.147964	0.057548	-0.270694	-0.338493	-0.314217	-0.180590	0.069466	
4702	0.144246	0.047683	-0.290797	-0.312624	-0.278211	-0.192876	0.234297	
4703	-0.394254	-0.445535	-0.139542	-0.178917	-0.277869	-0.445995	-0.323815	
4704	-0.044326	-0.060789	-0.478279	-0.175263	-0.247178	-0.134136	0.241188	

	7	8	9	...	4695	4696	4697	
0	0.440225	0.387539	0.198317	...	-0.644858	0.402671	-0.514844	\
1	0.768057	0.748454	0.331818	...	-0.503966	0.216510	-0.357477	
2	0.531923	0.384844	0.227503	...	-0.000414	-0.281179	-0.247392	
3	0.634397	0.652711	0.495081	...	-0.044968	-0.341473	-0.095775	
4	0.460076	0.462152	0.464006	...	-0.229769	-0.273678	-0.210430	
...	...	...	...	...	...	...	...	
4700	-0.593148	-0.579806	-0.070139	...	0.386289	0.056219	0.524241	
4701	0.009616	-0.101359	-0.156767	...	-0.148619	0.469789	0.053423	
4702	0.053442	-0.015958	0.227538	...	-0.298943	0.676858	0.163491	
4703	-0.360857	-0.407053	-0.140356	...	0.316794	0.030165	0.469330	
4704	-0.095945	-0.054650	0.037036	...	-0.135932	0.192944	0.076127	

노드 간의 유사도 행렬

→

Nodes

Nodes

1	1	1	0	1
1	1	1	0	0
1	1	1	1	1
0	0	1	1	0
1	0	1	0	1

Adjacency Matrix

GAT (Graph Attention Network)

KNU

- GAT는 각 노드가 이웃 노드들의 feature를 집계하여 노드의 새로운 feature를 계산함
- 이 과정에서 Attention Mechanism을 활용하여 각 이웃 노드와의 연관성을 바탕으로 중요도를 계산하고 가중치 부여

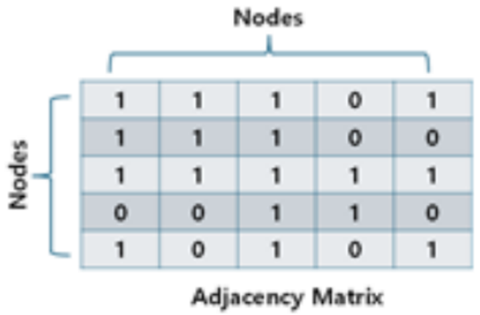
- Attention Coefficient: 노드 i에 대해 j가 갖는 중요도

$$e_{ij} = a(\mathbf{W}\vec{h}_i, \mathbf{W}\vec{h}_j)$$

- Masked Attention: attention coefficient에 softmax 적용

$$\alpha_{ij} = \mathrm{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} \exp(e_{ik})}$$

최종적으로, 가중치가 부여된 Adjacency Matrix 생성



e_{11}	e_{12}		e_{14}	
e_{21}	e_{22}	e_{23}		e_{25}
	e_{32}	e_{33}	e_{34}	e_{35}
e_{41}		e_{43}	e_{44}	e_{45}
	e_{25}	e_{53}	e_{54}	e_{55}

Attention Coefficient

a_{11}	a_{12}	0	a_{14}	0
a_{21}	a_{22}	a_{23}	0	a_{25}
0	a_{32}	a_{33}	a_{34}	a_{35}
a_{41}	0	a_{43}	a_{44}	a_{45}
0	a_{52}	a_{53}	a_{54}	a_{55}

Masked Attention

GAT 학습 및 테스트

- 데이터 전처리 단계에서 생성한 Feature matrix와 Adjacency matrix로 Graph Data 객체 생성

```
print('-----Graph Data 객체로 변환-----')
data = Data(x=feature_data, edge_index=edge_tensor, y=y, train_mask=train_mask)
print(data)
```

```
-----Graph Data 객체로 변환-----
Data(x=[4705, 51], edge_index=[2, 216666], y=[4705], train_mask=[4705])
```

- 생성된 Graph Data 객체를 GAT 모델의 입력으로 사용하여 학습

```
Epoch: 010, Train Loss: 0.882
Epoch: 020, Train Loss: 0.805
Epoch: 030, Train Loss: 0.768
Epoch: 040, Train Loss: 0.739
Epoch: 050, Train Loss: 0.711
...
Epoch: 470, Train Loss: 0.610
Epoch: 480, Train Loss: 0.609
Epoch: 490, Train Loss: 0.612
Epoch: 500, Train Loss: 0.594
```

모델 테스트 및 결과 비교

KNU

- 학습된 모델에 validation dataset을 적용해 node classification 수행

```
Node 2001: 정상입니다.
Node 1997: 정상입니다.
Node 1022: 치매위험입니다.
Node 1843: 치매입니다.
Node 1115: 치매입니다.
Validation Accuracy: 0.823
```

- 기존의 LSTM 모델은 클래스를 정상/이상 2가지로만 분류 -> 데이터 편향 때문이라고 추정
- Accuracy 결과는 80.85%

번호	측정항목	AI TASK	학습모델	지표명	기준값 점수	측정값 점수
1	치매 질환 분류 성능 (정상/이상)	Text Classification	Hierarchical Bi-directional LSTM	Accuracy	80 %	80.85 %

- 분석 결과, 3가지 클래스로 분류한 GAT 모델의 정확도가 82.3%로, 2가지 클래스로 분류한 LSTM 모델보다 **약 1.4% 상승**

기대효과 및 향후연구

KNU

치매 고위험자 이상 경고 시스템의 기대효과

- **치매 진행 예방 및 관리**: 고위험군으로 식별된 사용자에게 조기에 경고를 제공하여 적절한 조취를 취할 수 있도록 지원
- **의료 비용 절감**: 치매 치료에 소요되는 장기적인 의료 비용 절감 기대

향후 연구

- **정규화 방법 개선**
  - 데이터의 균형을 맞춰 안정적인 학습 가능
  - 기존에는 z-score

- **Feature 선택 방법 개선**
  - Random Forset 모델을 활용하여 feature의 중요도 계산 후 feature들을 상위, 중간, 하위 그룹으로 나눔
  - 그 후, 하위 그룹에 PCA를 적용해 주요 성분을 추출하고 차원 축소

- **데이터 오버 샘플링**
  - 기존에는 언더샘플링 수행
  - SMOTE 방법을 적용하여 적은 수의 클래스의 데이터 증가 -> 데이터의 불균형을 줄인 후 학습 수행
  - 언더샘플링과 오버샘플링 결과 비교 예정

- **미니 배치 학습 도입**
  - 실생활에서 지속적으로 생성되는 대규모 그래프 데이터를 효과적으로 학습하기 위한 방법
  - 대규모 그래프에서 일정 크기의 서브그래프를 생성해 모델 학습
  - 실제 환경에서 웨어러블 기기로부터 지속적으로 수집되는 데이터를 효과적으로 처리 가능

감사합니다.