

미세먼지와 교통 혼잡도에 따른 교통수단 추천에 관한 연구

박창현, 서종현, 주세훈, 김원종, 장세훈, 조주필 교수

국립군산대학교

cnrrn0411@naver.com, flopssy@naver.com, wntpgns8211@naver.com,
01055130143@gmail.com, sehoon0926@naver.com, stefano@kunsan.ac.kr

A Study on the Recommendation of Transportation by Fine Dust and Traffic Congestion

Park Changhyun, Seo Jonghyun, Joo Sehoon, Kim Wonjong, Jang Sehoon, Jo Joopil

요약

본 논문은 전라북도 여러 지역을 여행하는 사람들을 위한 정보를 제공하는 웹, 앱 개발에 관한 연구를 다루고 있다. 이곳저곳 여행을 다니는 사람들이 늘어남으로써 전라북도에 속한 지역들에 대하여 어떠한 것들이 있는지에 대해 알아보기 쉽게 정보를 알 수 있도록 하는 것을 목표로 한다. 본 연구에서는 전라북도에 속한 각 지역들의 정보를 카테고리별로 찾아보고 정리하였다. 또한, 웹이랑 앱 모두에서 정보를 찾아볼 수 있도록 같은 기능으로 개발하였다.

I. 서 론

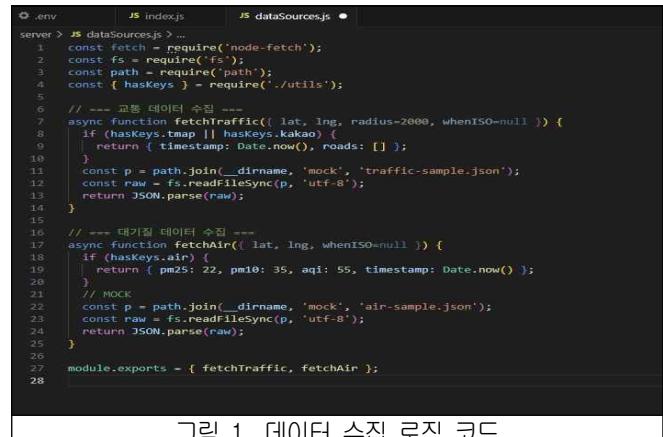
최근 미세먼지로 인한 건강 피해가 사회적 문제로 대두되며, 실시간 공기질 예측의 중요성이 점차 부각되고 있다. 특히 지역 단위의 미세먼지 예측 정보는 시민들의 일상적인 야외 활동, 출퇴근 경로 선택, 건강 관리 등과 밀접하게 연관되어 있어, 보다 정밀하고 신뢰할 수 있는 예측이 요구된다. 한편, 도시 인구의 급격한 증가와 함께 차량 통행량의 증가는 교통 혼잡도를 심화시키고 있으며, 이는 대기 오염의 주요 원인 중 하나로 작용하고 있다. 도로 위 차량 수가 많을수록 교통 흐름이 저해될 뿐만 아니라, 배출 가스의 누적으로 인해 대기질이 악화되는 현상이 반복되고 있다. 따라서, 미세먼지와 교통량이라는 두 요소를 통합적으로 고려한 예측 시스템 구축이 필요하다. 이를 통해 시민의 건강을 보호하고, 효율적인 이동 시간 관리, 도시 환경 개선, 나아가 탄소 배출 절감 등의 다양한 사회적 효과를 기대할 수 있다.

이러한 배경과 필요성에 따라 본 프로젝트는 지역 기반의 실시간 공기질 및 교통 혼잡도에 따른 교통수단 추천 시스템을 개발하고자 한다.

II. 본론

1. 데이터 수집

본 논문에서는 교통 혼잡도와 대기 오염 수준(미세먼지 등)에 따라 교통수단을 추천하기 위해, 위치 기반의 실시간 데이터를 수집하는 데이터 수집 모듈을 구현하였다. 필요한 데이터들을 수집하기 위해 한국환경공단_에어코리아에서 실시간 미세먼지 농도를 API로 받았고, 이 데이터는 교통수단 추천 시 대기 상태를 고려하기 위해 사용하였다. 또 T-MAP에서 실시간 교통 정보를 받아와 차량 속도, 혼잡도, 길 찾기 등을 제공하였다.

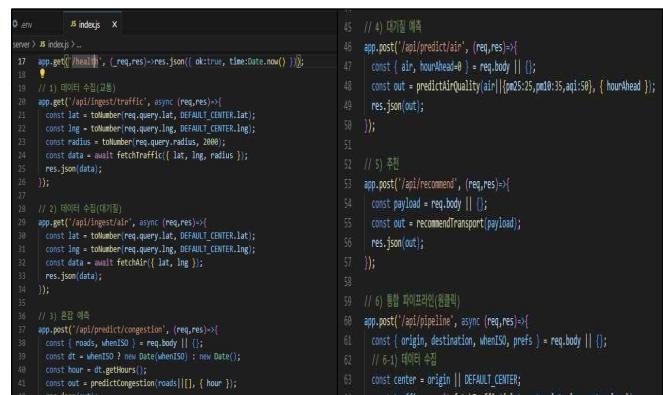


```
server > JS dataSources.js
server > JS dataSources.js > ...
1  const fetch = require('node-fetch');
2  const fs = require('fs');
3  const path = require('path');
4  const { haskeys } = require('../utils');
5
6  // --- 교통 데이터 수집 ---
7  async function fetchTraffic({ lat, lng, radius=2000, whenISO=null }) {
8    if (haskeys.tmap || haskeys.kakao) {
9      return { timestamp: Date.now(), roads: [] };
10   }
11   const p = path.join(__dirname, 'mock', 'traffic-sample.json');
12   const raw = fs.readFileSync(p, 'utf-8');
13   return JSON.parse(raw);
14 }
15
16 // --- 대기질 데이터 수집 ---
17 async function fetchAir({ lat, lng, whenISO=null }) {
18   if (haskeys.tmap) {
19     return { pm25: 22, pm10: 35, aqi: 55, timestamp: Date.now() };
20   }
21   // MOCK
22   const p = path.join(__dirname, 'mock', 'air-sample.json');
23   const raw = fs.readFileSync(p, 'utf-8');
24   return JSON.parse(raw);
25 }
26
27 module.exports = { fetchTraffic, fetchAir };
28
```

그림 1. 데이터 수집 로직 코드

2. 교통, 대기질 데이터 수집 및 예측

수집한 실시간 교통 및 대기질 데이터를 기반으로 혼잡 예측 및 교통수단을 추천하는 코드를 구현하였다.



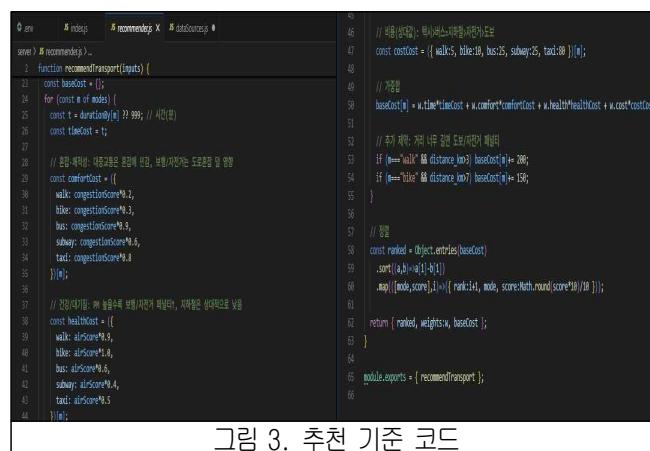
```
server > JS index.js
server > JS index.js > ...
17 app.get('/traffic', (req,res) => res.json({ dictow, timestamp: Date.now() }));
18 // 1) 데이터 수집(대기질)
19 app.get('/api/air', (req,res) => {
20   const lat = toLat(req.query.lat, DEFAULT_CENTER.lat);
21   const lng = toLat(req.query.lng, DEFAULT_CENTER.lng);
22   const radius = toLat(req.query.radius, 2000);
23   const data = await fetchAir({ lat, lng, radius });
24   res.json(data);
25 });
26
27 // 2) 데이터 수집(대기질)
28 app.get('/api/ingest/air', async (req,res) => {
29   const lat = toLat(req.query.lat, DEFAULT_CENTER.lat);
30   const lng = toLat(req.query.lng, DEFAULT_CENTER.lng);
31   const radius = toLat(req.query.radius, 2000);
32   const data = await fetchAir({ lat, lng, radius });
33   res.json(data);
34 });
35
36 // 3) 혼잡 예측
37 app.post('/api/predict/congestion', (req,res) => {
38   const { roads, whenISO } = req.body || {};
39   const dt = whenISO ? new Date(whenISO) : new Date();
40   const hour = dt.gethours();
41   const out = predictCongestion(roads || [], hour);
42   res.json(out);
43 });
44
45 // 4) 대기질 예측
46 app.post('/api/predict/air', (req,res) => {
47   const { lat, hourAhead=0 } = req.body || {};
48   const out = predictAirQuality(air || [pm25:25,pm10:35,aqi:50], { hourhead });
49   res.json(out);
50 });
51
52 // 5) 추천
53 app.post('/api/recommend', (req,res) => {
54   const payload = req.body || {};
55   const out = recommendTransport(payload);
56   res.json(out);
57 });
58
59 // 6) 통합 파이프라인(원래는)
60 app.post('/api/pipeLine', async (req,res) => {
61   const { origin, destination, whenISO, prefs } = req.body || {};
62   // 6-1) 대기 수준
63   const center = origin || DEFAULT_CENTER;
64   const traffic = await fetchTraffic({ lat:center.lat, lng:center.lng });
65 });
66
```

그림 2. 교통, 대기질 데이터 수집 후 예측 코드

3. 추천 기준 설정

사용자에게 적절한 교통수단을 추천하기 위해, 시간, 교통 혼잡도, 대기 오염도, 경제성 등 다양한 요소를 통합적으로 고려한 교통수단 추천 알고리즘을 구현하였다. 해당 알고리즘은 각 교통수단의 총 비용(baseCost)을 계산한 후, 이를 기준으로 우선순위를 정하여 추천 결과를 제공한다.

이동 시간(Time)은 교통수단별 예상 이동 시간이 길수록 비용 증가하도록 설계하였고, 혼잡도(Comfort)는 교통수단별 혼잡도 점수를 반영하여 쾌적성 저하에 따른 불편을 계량화하여 버스 및 지하철은 혼잡도 영향을 크게 받는다. 건강(Health)은 미세먼지(PM2.5, PM10) 농도 및 대기 질지수(AQI)를 기반으로 실외 활동(도보, 자전거)의 건강 영향도를 반영하였다. 경제성(Cost)은 교통수단의 상대적 비용을 반영. 도보는 비용이 없고, 택시는 가장 높은 비용이 적용된다. 이러한 요소들을 반영하여 산출된 총 비용(baseCost)를 바탕으로 사용자에게 적절한 교통수단을 추천하게 된다.

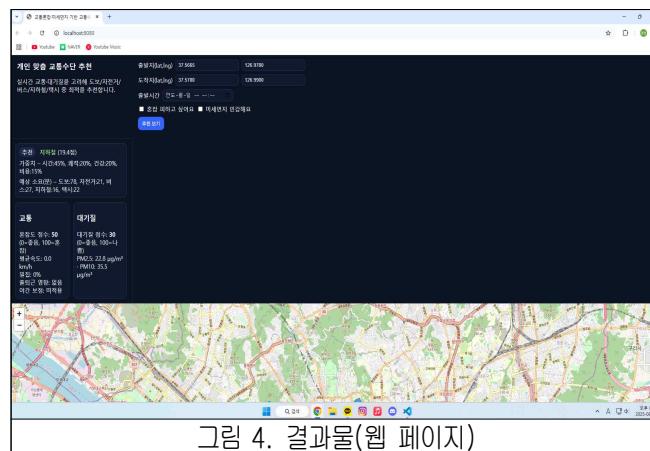


```
1 // 사용자에게 적절한 교통수단을 추천하기 위해, 시간, 교통 혼잡도, 대기
2 // 오염도, 경제성 등 다양한 요소를 통합적으로 고려한 교통수단 추천 알고
3 // 리즘을 구현하였다. 해당 알고리즘은 각 교통수단의 총 비용(baseCost)
4 // 을 계산한 후, 이를 기준으로 우선순위를 정하여 추천 결과를 제공한다.
5
6 // 이동 시간(Time)은 교통수단별 예상 이동 시간이 길수록 비용 증가하
7 // 도록 설계하였고, 혼잡도(Comfort)는 교통수단별 혼잡도 점수를 반영하
8 // 여 쾌적성 저하에 따른 불편을 계량화하여 버스 및 지하철은 혼잡도 영향
9 // 을 크게 받는다. 건강(Health)은 미세먼지(PM2.5, PM10) 농도 및 대기
10 // 질지수(AQI)를 기반으로 실외 활동(도보, 자전거)의 건강 영향도를 반영
11 // 하였다. 경제성(Cost)은 교통수단의 상대적 비용을 반영. 도보는 비용이
12 // 없고, 택시는 가장 높은 비용이 적용된다. 이러한 요소들을 반영하여 산출
13 // 된 총 비용(baseCost)를 바탕으로 사용자에게 적절한 교통수단을 추천
14 // 하게 된다.
15
16 // 추천 기준: 대중교통은 혼잡에 반길, 평화기반에는 도로환경 절 영향
17 const baseCost = {
18   for (const mode of modes) {
19     const t = duration[mode] ?? 999; // 시간(0)
20     const cost0 = t;
21
22     // 혼잡: 평화: 대중교통은 혼잡에 반길, 평화기반에는 도로환경 절 영향
23     const comfortCost = {
24       walk: congestionScore*2,
25       bike: congestionScore*3,
26       bus: congestionScore*5,
27       subway: congestionScore*5,
28       taxi: congestionScore*8
29     }[mode];
30
31     // 건강: 대기, 미세먼지, 대중교통의 보행(자전거)은 평화기반, 지하철은 상대적으로 낫음
32     const healthCost = {
33       walk: airScore*3,
34       bike: airScore*1.4,
35       bus: airScore*0.5,
36       subway: airScore*0.4,
37       taxi: airScore*0.5
38     }[mode];
39
40     // 경제성: 대중교통은 저렴한 편, 개인차는 비싸다.
41     const cost1 = {
42       walk: 0.0,
43       bike: 0.0,
44       bus: 0.0,
45       subway: 0.0,
46       taxi: 100.0
47     }[mode];
48
49     // 총 비용: 편의(시간)+편안함(혼잡)+건강(환경)+경제성(비용)
50     const cost2 = [t, comfortCost, healthCost, cost1][0];
51
52     // 추가 제한: 평화: 나중 점수 더해지면 제외
53     if (mode === 'walk' && distance[mode] > 200) {
54       if (t === 0) distance[mode] += 100;
55     }
56
57     // 정렬
58     const ranked = (obj, entries) => entries.map((entry) => {
59       const [mode, score] = entry;
60       return {rank: 1, mode, score: Math.round(score*10)/10};
61     });
62
63     return {ranked, weights, baseCost};
64   }
65 }
66
67 module.exports = {recommendTransport};
```

그림 3. 추천 기준 코드

4. 결과물(웹 페이지 구현)

카카오맵 API를 이용하여 출발지와 도착지 주소를 좌표로 변환하는데 사용하여 시각화하였다. 이렇게 사용자의 위치 정보를 기반으로 미세먼지 농도 및 교통 혼잡도를 종합적으로 고려하여 최적의 교통수단을 추천하는 웹 애플리케이션을 개발하였다.



III. 결론

본 논문에서는 교통 혼잡도와 대기질(미세먼지) 데이터를 통합하여, 사용자에게 적절한 교통수단을 실시간으로 추천하는 시스템을 개발하였다. 실제 위치 기반의 데이터 수집, 가중치 기반의 추천 알고리즘, 지도 시각화를 구현함으로써, 사용자가 보다 쾌적하고 안전한 이동수단을 선택할 수 있도록 지원하였다. 이 웹 페이지를 개발해서 얻을 수 있는 기대효과는 총 4가지 정도가 있다. 처음으로는 교통과 환경요인을 통합하여 고려한 복합적 추천 알고리즘을 제공하여 최근 심각한 문제로 대두되고 있는 대기질 오염에 대해 사람들에게 경각심을 줄 수 있다. 이로써 개인 건강 보호 효과도 기대해 볼 수 있다. 두 번째로 교통 혼잡 완화에 기여할 수 있다. 실시간 혼잡도 데이터를 반영하여 덜 혼잡한 수단이나 경로를 유도, 도시 내 교통 분산 효과 기대해 볼 수 있다. 세 번째로는 사용자 선택 옵션 구현이다. 사용자 맞춤형 옵션(예: 미세먼지 민감도 반영, 혼잡 피하기 등)을 통한 개인화된 추천 기능 제공하여 사용자의 높은 만족도를 기대해 볼 수 있다. 마지막 네 번째로는 스마트 모빌리티 서비스 확장 기반 마련이다. 향후 AI 기반 경로 추천, 실시간 대중교통 통합 정보 서비스 등으로 확장 가능성을 기대해 볼 수 있다. 하지만 한계점도 분명히 있다.

실시간 교통 및 대기질 데이터는 한정된 API나 지역(예: 서울) 기반으로 수집되어, 전국적 확장성에 제약이 있다. 또한 추천 알고리즘은 비교적 단순한 가중치 기반이며, 딥러닝/머신러닝 기반의 고도화된 예측 모델은 도입되지 않았다. 그러므로 향후 머신러닝 기반으로 교통 및 환경 요인을 예측하여 더 정교한 경로 추천 모델을 구현하고 카카오맵, Tmap 등 다양한 플랫폼과의 API 연동으로 서비스 범위를 확장하는 방향으로 다양한 연구가 필요하다.

ACKNOWLEDGMENT

본 논문은 (재)전북테크노파크 재원을 지원 받아 수행된 지역특성화 산업전문인력양성사업 연구 결과입니다.

참 고 문 현

<https://apis.map.kakao.com/> – 카카오맵

<https://www.data.go.kr/data/15073861/openapi.do> – 한국환경공단 에어코리아

<https://openapi.sk.com/> – 티맵