

學士學位論文

Wi-fi 기계학습을 통한 공간의 혼잡도 분석 및 AR 서비스 개발

전남대학교

전자컴퓨터공학부
소프트웨어공학전공

곽 선 우

指導教授 최 덕 재

2020年 7月

Wi-fi 기계학습을 통한 공간의 혼잡도 분석 및 AR 서비스 개발

전남대학교
전자컴퓨터공학부
소프트웨어공학전공

곽 선 우

指導教授(지도교수) 최 덕 재 (인)

主任教授(주임교수) 박 혁 로 (인)

2020年 7月

목 차

1. 서론.....	4
2. 배경 지식.....	5
2.1 자료 소개.....	5
2.2 관련 연구	7
3. 시스템 설계 및 구현.....	9
3.1 사용자 요구사항	9
3.2 시스템 설계의 목표	10
3.3 시스템 설계	11
3.4 구현.....	12
3.4.1 구현 환경.....	12
3.4.2 구현 결과.....	14
4. 실험 및 분석.....	22
4.1 실험 환경.....	22
4.2 실험 결과.....	23
4.3 실험 결과 분석	25
5. 결론 및 향후과제.....	26
6. 참고문헌.....	27

1. 서론

전남대학교 대학만족도 조사 결과(2018)에 따르면, 전남대에서 활용 가능한 무선랜 단말기의 수는 <표 1>과 같다. 2018년 기준 설치한 무선랜 단말기까지 고려하면 현재 전남대학교 내에는 4,000대 이상의 무선랜 단말기가 존재한다. 전남대학교 정보전산원에서 공개한 자료에 따르면 전남대학교 전체 무선랜의 인증 자료는 하루 평균 10만 건의 접속자료가 생성되고 한 달을 기준으로 약 300만 건이 넘는 빅데이터가 생성된다. 최근 모바일과 웹 사용의 증가로 일상생활에서 기록되는 무선랜 자료를 다양하게 분석하여 의미 있는 정보와 지식을 얻어내고자 하는 연구가 증가하고 있다(조나혜·강영옥 2016). 하지만 전남대학교 내에서는 이런 빅데이터를 활용하지 못하고 버려지고 있다.

<표 1> 전남대학교에서 활용 가능한 무선랜 단말기 수(2018. 12. 기준)

구분	2018년 이전 보유AP	2018년 추가설치	합계
JNU	783	2,500	3,283
통신3사	1,094		1,094
합계	1,877	2,500	4,377

무선랜 자료는 시공간적인 정보를 담고 있으며, 다차원적으로 탐색하고 시각화하여 기존에 분석하지 못했던 다양한 의미를 찾을 수 있다. 이에 따라 무선랜 데이터의 잠재력을 반영하듯 관광, 교통, 측위 등 여러 분야에서 무선랜 빅데이터의 활용 사례들이 보고되고 있다. 그 외에도 공공 와이파이 데이터가 개방되면서 이런 자료에 접근하기 쉬워졌고 이를 활용하여 공공기관이나 정부의 정책제안에도 활용할 수 있다.

무선랜이 발달하면서 센서 역할을 하게 됨에 따라 이를 활용한 위치기반 서비스에 관한 관심이 증가하고 있고 사용자의 실내 위치 추적이나 측위에 관한 연구가 활발히 진행되었다(정용국 2015). 전남대학교 정보전산원에서 공개한 혼잡도 실증 결과 보고서(2019)에서 설문조사에 따르면 학생들이 혼잡도 조회 기능을 바라는 장소에서 도서관이 가장 높은 비율을 차지했다. 혼잡정보는 위치추적과 달리, 공간 내 정확한 위치를 알지 않고도 충분히 구해낼 수 있다는 장점이 있고 도서관을 이용하려는 학생들에게 필요한 정보이므로 전남대학교에서 관련 시스템이 필요하다.

2. 배경 지식

2.1 자료 소개

본 연구를 위한 자료는 보기 쉽게 표로 정리하였다. 자료는 무선랜 로그 데이터를 변형한 접속데이터, 검증을 위한 실제 인원수 데이터, 예측을 위한 공기질 및 기상 관측 데이터를 포함한다. 이는 자료 출처, 측정 기간, 측정 단위, 측정 장소, 공개 대상 항목과 함께 제시하였으며 이 외에도 데이터들을 모은 최종데이터를 포함하였다.

<표 2>는 도서관 인원수 자료이다. 전남대학교 도서관 열람실에는 각각 2개에서 4개의 무선 AP가 설치되어 있고 무선랜 단말기와 인증 서버로부터 해당 단말기의 MAC 주소, 접속한 단말기의 MAC 주소, 접속 시작시각과 종료시각 등의 자료를 얻을 수 있다. 이 자료를 기반으로 무선랜 접속인원수를 측정하였다. 자료는 2019년 9월 20일 00시부터 2019년 10월 13일 24:00까지 수집하였다.

<표 2> 도서관 무선랜 접속 데이터

자료명	도서관 무선랜 접속 데이터
자료 출처	전남대학교 정보전산원
측정 기간	2019년 9월 20일 - 10월 13일
측정 단위	사용자가 접속 시마다 측정
공개 대상 항목	무선랜 단말기 MAC 주소, 휴대용 단말기 MAC 주소, 접속 시작시각, 접속 종료시각, 네트워크 주소
측정 장소	전남대학교 백도

<표 3>은 9월 20일 00시부터 2019년 10월 13일 24:00까지 도서관 열람실을 방문하여 1시간 단위로 재실 인원수를 측정하였다. 재실 인원수는 혼잡도를 구하기 위한 회귀모델의 종속변수로 이용되었다. 도서관 무선랜 접속데이터와 동일기간 직접 측정하였다.

<표 3> 실제 도서관 이용자 수 자료

자료명	실제 도서관 이용자 수 자료
자료 출처	직접 측정
측정 기간	2019년 9월 20일 - 10월 13일
측정 단위	1시간 단위
공개 대상 항목	재실 인원수
측정 장소	전남대학교 백도

<표 4>는 도서관 혼잡도 예측을 위한 도서관 공기질 및 기상관측 자료이다. 이 자료는 2019년 9월 20일 00시부터 2019년 10월 13일 24:00까지의 자료를 수집하였다. 실내 공기질 자료는 전남대학교 열람실의 공기질 측정 센서에서 수집하였다. 수집한 자료는 온도(°C), 습도(%), 이산화탄소 농도(ppm), 휘발성 유기화합물 농도(ppb), 미세먼지농도(ug/m3), 빛의 조명도(lx), 소음 크기(dB)이다.

기상관측 자료는 기상자료개방포털의 종관규모 기상관측 자료를 수집하였다. 종관 기상관측이란 종관규모의 날씨를 파악하기 위해 정해진 시각에 모든 관측소에서 같은 시각에 실시하는 지상관측을 뜻한다. 종관규모는 일기도에 표현된 보통의 고기압이나 저기압의 공간적 크기 및 수명을 말하며, 주로 매일의 날씨 현상을 뜻한다. 수집한 자료는 기온(°C), 강수량(mm), 풍속(m/s), 습도(%)이다.

<표 4> 도서관 혼잡도 예측을 위한 공기질 및 기상관측 자료

자료명	도서관 혼잡도 예측을 위한 공기질 및 기상관측 자료
자료 출처	전남대학교 정보전산원, 기상청 기상자료개방포털
측정 기간	2019년 9월 20일 - 10월 13일
측정 단위	1시간 단위
공개 대상 항목	내부 온도, 습도, 이산화탄소 농도, 휘발성 유기화합물 농도, 소음 세기, 빛 세기, 미세먼지농도, 외부기온, 강수량, 풍속, 습도
측정 장소	전남대학교 백도, 광주지방기상청

<표 5>는 자료를 수합하여 생성된 변수와 자료에 대한 설명을 보여준다. 추후 모든 자료에 대한 상관관계를 분석하고 이를 통해 인공지능 학습에 사용할 변수를 선택한다.

<표 5> 최종 수합 자료

번호	생성된 변수	설명
1	real_number	재실 인원수
2	wifi_number	무선랜 접속인원수
3	temperature(in)	도서관 내부 온도
4	humidity	도서관 내부 습도
5	co2	도서관 내부 co2 농도
6	voc	도서관 내부 voc 농도
7	light	도서관 내부 빛세기
8	rain	외부 강수량
9	rain?	외부 비 유무
10	temperature(out)	외부 온도
11	windpower	외부 풍속
12	month	월
13	day	일
14	hour	시간

2.2 관련 연구

본 연구에서는 무선랜 데이터를 활용한 선행연구에 대한 분석방법 및 데이터 활용도를 비교하여 이를 활용하여 혼잡도를 구할 수 있는지 확인한다. 또한, 혼잡도에 대한 개념을 정의하고 혼잡 정보제공 관련 선행연구를 검토하였다.

1) 무선랜 데이터 분석 관련 연구

무선랜 자료는 시공간적인 정보를 담고 있으며 기존에 분석하지 못했던 다양한 의미를 찾을 수 있다. 무선랜 데이터를 활용한 연구는 다음과 같다. 첫째, 무선랜을 실내·외에서 측위를 하는 방법으로 사용했다. 무선랜을 기반으로 하는 위치 시스템에서 구축된 통신 네트워크를 사용하여 추가적인 설치 없이, 넓은 영역에 적용할 수 있으며 안정적인 시스템이지만, 실내 환경의 다양한 요소에 영향을 받아 위치 정확도가 떨어지는 단점이 있다(Yanying Gu, 2009).

둘째, 제주도에서 빅데이터 기반 관광지 자동 추천 플랫폼(문준환 외, 2019)을 개발하였으며 공공 와이파이 접속정보 등을 활용하여 관광객의 이동 경로의 수가 떨어지는 지역에 정책적 지원이 필요하다는 정책제안의 자료로 활용하였다.

셋째, 김포시의 경우 전기차 충전소를 설치하기 위한 장소를 선정하기 위해 공공 와이파이 접속정보 등을 사용하였고 분석을 통해 유동인구가 많은 지역을 우선으로 설치하도록 하였다(김포타임즈, 2019년 12월 6일).

이외에도 전국적으로 공공 와이파이를 확대하여 빅데이터를 수집하고 이를 활용하여 유용한 정보를 얻으려는 시도가 진행 중이다. 하지만 무선랜 데이터의 중요성이나 활용 가능성은 알려졌지만 활용된 사례가 적다. 선행연구 분석을 통해 분석방법은 다르지만 이와 같은 무선랜 데이터를 분석하여 도서관에서도 유의미한 정보를 얻어낼 수 있다고 판단한다.

2) 혼잡도 정의 및 활용 연구

도서관 혼잡도를 측정하기 위해 혼잡도가 무엇인지 명확하게 정의하였다. 혼잡도 측정을 시도한 다른 논문에서는 혼잡도를 공간적 밀도로 보는 경우와 더불어 공간적 밀도에 따른 심리적인 스트레스라고 보는 경우, 공간적 밀도를 제외한 심리적, 주관적 평가의 결과로 보는 경우 등으로 정의하였지만(전태준, 2011; Stokols, 1972), 학생들에게 혼잡한 상황을 피하도록 혼잡정보를 제공한다는 본 연구의 특성에서는 혼잡도를 측정하고자 하는 공간에 대한 인원수의 밀도를 ‘혼잡도’라고 정의하는 것이 합리적으로 판단하였다. 따라서 본 연구에서 혼잡도란 특정 열람실의 혼잡 정도를 표시하는 지표로써 공간의 정원에 따른 인원수 밀도로 정의하였다. 혼잡도는 단계별로 5단계로 나누었다. 이를 통해 혼잡도에 따른 혼잡레벨을 나타내어 사용자에게 친숙하게 표현하였다.

홍보현(2019)의 연구에서는 도서관의 공간적 특성이 공간 이미지 형성에 물리적인 특성이 작용함을 파악하였다. 혼잡한 공간 이미지는 고정된 시설이나 장비보다는 온도, 소음, 청결과 같은 주변 환경의 변화에 영향을 받는 것으로 나타났다(홍보현, 2019). 따라서 본 연구에서 혼잡도 정보를 제공할 때 단순히 공간의 혼잡정보뿐만 아니라 공기질 정보까지 제공함으로써 사용자가 혼잡한 공간 이미지에 관한 판단을 스스로 하도록 한다.

위에서 정의했던 혼잡정보를 활용한 연구는 다음과 같다. 첫째, 도시철도 객실혼잡 정보 제공에 따른 연구(김성민, 2018)에서는 혼잡정보를 제공하기 위해 차량의 무게에 따른 혼잡도를 정의하였고, 고객들에게 차량별 혼잡정보가 제공된다면 특정 차량의 선택 여부와 관계없이 혼잡뿐만 아니라 신체적 불편이나 범죄 불안 등 여러 요소를 해소하고자 해당 정보를 이용하려는 것으로 분석하였다.

둘째, 교통정보 빅데이터 기반의 시공간 혼잡지표 개발에 대한 연구(한여희, 2018)에서 평균 통행속도라는 빅데이터를 분석하여 혼잡지표를 개발하여 교통관리용 혼잡지표로의 활용 가능성을 도출하였다.

이러한 연구들을 보아 혼잡정보는 교통관리 분야에서 많은 연구가 이루어졌다. 연구마다 혼잡도를 산정하는 방식이 다르고 이 정보를 활용하는 목적이 사람마다 달랐지만, 혼잡도가 사람들에게 유용한 정보가 됨을 알 수 있고 교통관리분야 이외에도 사용할 수 있다고 판단한다. 따라서 본 연구에서는 도서관에서 발생하는 무선랜 접속 빅데이터를 활용하여 혼잡정보를 제공한다.

3. 시스템 설계 및 구현

3.1 사용자 요구사항

01. 추가적인 장비 설치 없이 기존 장비로 수집되는 Wi-fi 데이터를 활용하여 공간별 혼잡도를 분석 및 예측하도록 한다.
02. Wi-fi Raw 데이터에서 한 사람을 특정하여 식별할 수 없도록 학번 등의 정보는 해시 처리하여 보안성을 유지한다.
03. 한사람이 노트북, 핸드폰, 태블릿 pc등 여러 기기를 사용하여 Wi-fi에 접속 시 발생하는 중복 케이스를 배제할 수 있도록 한다.
04. 사용자가 예측 값을 신뢰할 수 있도록 공간별 혼잡도 예측 정확도가 최소 70% 이상 나오도록 한다.
05. 공간 내 인원이 아닌 외부로 지나가던 사람이 Wi-fi에 접속되는 경우를 배제할 수 있도록 한다.
06. 열람실별 재실인원에 따른 혼잡도를 바탕으로 건물의 전체적인 혼잡 상태도 알 수 있도록 한다.
07. 혼잡도 외에도 공기질 등 부가적인 다른 정보들도 제공하도록 한다.
08. 전남대학교 도서관, 정보전산원, 공대7호관 3가지 건물들을 3D Object 형태로 GPS 위치정보에 맞게 나타나도록 한다.
09. 사용자들이 조작하기 간편하고 재미를 느낄 수 있도록 GUI를 구성한다.
10. 도서관이 매우 혼잡할 시 그 상황을 바로 알 수 있도록 알림기능을 구현하도록 한다.

3.2 시스템 설계의 목표

본 연구의 목적은 무선랜 접속자료를 활용하여 전남대학교 도서관 혼잡정보를 제공 시스템 개발을 목표로 한다. 이를 위한 연구 질문은 세 가지이다. 첫째, 무선랜 접속자료를 통해 도서관 혼잡도 정보를 계산할 수 있는가? 둘째, 학생들에게 실시간 정보제공 시스템이 구축 가능한가? 셋째, 실시간 자료를 통해 향후 혼잡도를 예측 가능한가?

1) 머신러닝을 통한 데이터 분석 및 예측

- 현재와 미래의 시간별 예측 혼잡도를 구하기 위해서 머신러닝을 위한 오픈소스 라이브러리를 이용한다.
- AP의 특정 칼럼 값을 이용하여 개인당 여러 기기를 사용하는 경우, 내부적으로 데이터가 중복된 경우를 제거한다.
- 시간에 따른 AP 접속 수와 공기질 데이터, 날씨 데이터를 시계열 데이터에 따라 분석하고 기계학습으로 1시간 단위로 예측한다.
- 예측된 AP 접속 수를 기준에 따라 혼잡도 5개의 클래스로 분류한다.
- 머신러닝한 결과를 JSON 파일로 저장하고 FireBase로 전송하여 실시간 알림서비스를 제공한다.

2) AR 어플리케이션

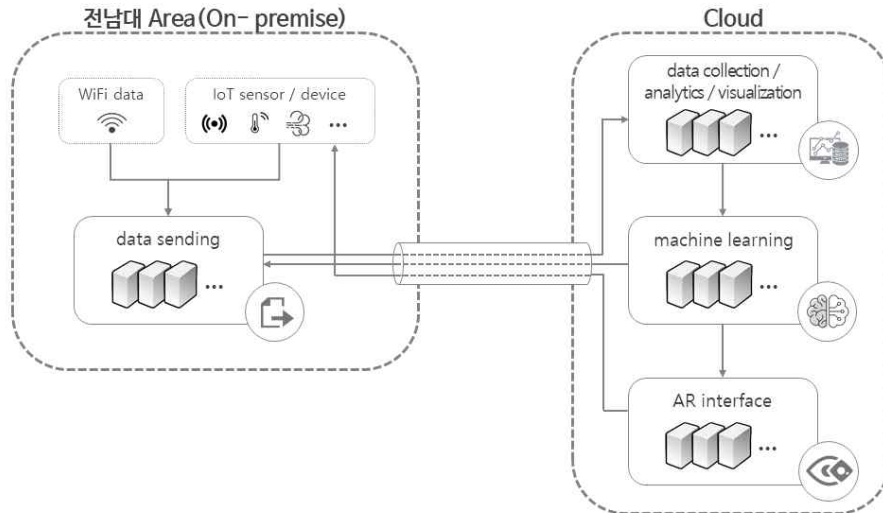
- 증강현실을 구현하기 위해서 Unity의 Vuforia SDK를 사용한다.
- GPS 위치 정보(위도, 경도, 고도)를 가진 3가지 3D Building Object를 제작한다.
- Object 클릭 시 열람실 정보를 제공할 AR 메뉴를 개발한다.
- 5개 클래스로 분류된 혼잡도를 표현하기 위한 Emoticon Animation 및 다양한 이펙트를 개발한다.



< 그림 1. AR 서비스 예시 >

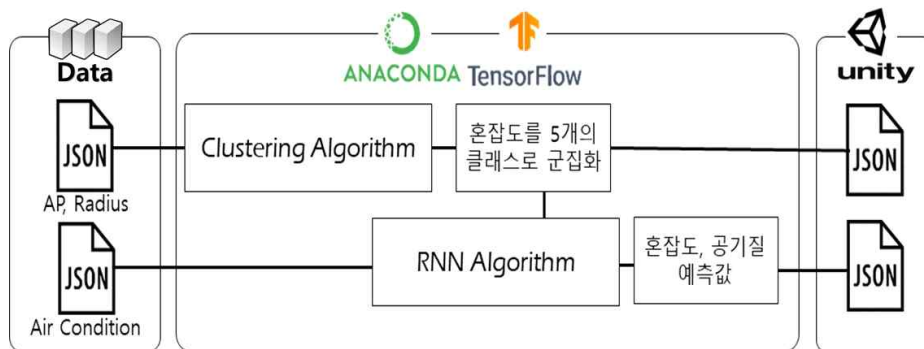
3.3 시스템 설계

전체 시스템은 크게 데이터 수집 및 전송, 기계학습, 증강현실로 구성된다.



< 그림 2. 전체 구성도 >

대량의 데이터를 가공, 분석하는데 발생하는 자원 부족 현상을 해결하기 위해 KOREN SDI의 오픈스택에서 3대의 인스턴스를 할당받았다. AP 데이터를 분석 및 처리하기 위한 인스턴스에 ELK Stack을 설치하고, 데이터를 저장, 검색 및 처리하여 머신러닝 서버로 전송한다. 기계학습을 위한 인스턴스에 Anaconda와 오픈소스 라이브러리를 설치하고, 1시간 단위로 재실인원을 예측하는 모델을 만들고 얻은 결과값을 혼잡도 5단계로 분류한다. 남은 인스턴스에 DB와 Firebase 서버를 연동해 머신러닝 인스턴스에서 분류한 결과를 전송받는다.



< 그림 3. 머신러닝 구성도 >

전체적인 데이터 흐름은 다음과 같다. DB 서버로부터 json형태의 Wi-fi Raw 데이터와 공기질 데이터 파일을 가져와 여러 분석 라이브러리를 통해 데이터를 가공 및 처리하고 그 결과를 머신러닝 서버에서 시계열 예측 알고리즘인 RNN을 사용하여 혼잡도와 공기질을 예측한다. 그 이후, 임의로 정한 혼잡도 5단계의 기준에 따라 혼잡도를 구분하고 이 값을 JSON 파일로 변환하여 AR 서버로 전송한다.

3.4 구현

3.4.1 구현 환경

무선랜 관련 빅데이터 정보(전남대 기준, 2019년 4월 한 달간 수집되는 인증 약 780만건, Accounting 약 260만건, AP에서 수집되는 무선랜 핑거프린트 정보 등)와 공간(건물, 층별, 호수, 면적 등), 이에 설치된 AP위치, RADIUS 사용자 인증 정보(개인정보 비식별 조치한 소속, 성별, 연령, 신분(외부인 여부) 등의 정보)를 결합하여 처리하기 위해 기존의 온프리머스 자원만으로는 한계가 있기 때문에 클라우드 인스턴스 4대를 같이 활용했다.

1) 대용량 데이터 처리를 위한 클라우드

Usage Summary

Select a period of time to query its usage:

From: 2019-11-07 To: 2019-11-08 [재설정](#) The date should be in YYYY-MM-DD format.

활성화된 인스턴스: 3 Active RAM: 24GB This Period's VCPU-Hours: 349.35 This Period's GB-Hours: 873.38 This Period's RAM-Hours: 715471.40

사용 [CSV 요약 다운로드](#) [Download Juju Environment File](#)

인스턴스 이름	VCPU	디스크	RAM	생성된 이후 시간
kafka	4	10GB	8GB	2개월, 4주
elastic	4	10GB	8GB	3개월
ML	4	10GB	8GB	3개월, 1주

Displaying 3 items

< 그림 4. http://sdi.koren.kr:8088/ 인스턴스 >

Usage Summary

Select a period of time to query its usage:

The date should be in YYYY-MM-DD format.

2019-11-07 to 2019-11-08 [재설정](#)

활성화된 인스턴스: 1 Active RAM: 16GB
This Period's VCPU-Hours: 232.88
This Period's GB-Hours: 1455.53
This Period's RAM-Hours: 475900.17

사용 [CSV 요약 다운로드](#) [Download Juju Environment File](#)

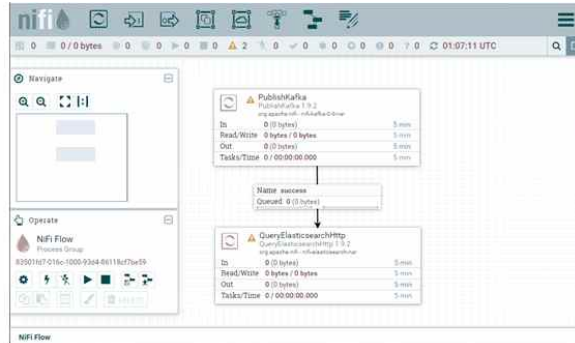
인스턴스 이름	VCPU	디스크	RAM	생성된 이후 시간
JAMAR_VMI	8	50GB	16GB	3개월

Displaying 1 item

< 그림 5. http://scloud.koren.kr:8087/ 인스턴스 >

KOREN SDI의 오픈스택에서 4대의 인스턴스(CPU : 4, RAM : 8GB 3대, CPU : 8, RAM : 16GB 1대)를 할당하여 Ubuntu 16.04.6 LTS 설치 후 사용하였다.

2) 데이터 분석을 위한 오픈소스 사용



< 그림 6. ELK Stack과 Kafka 연결 >

- elasticsearch-7.3.0, logstash-7.3.0, kibana-7.3.0 설치
- kafka 2.11, nifi-1.9.2 설치

대규모 메시지 데이터를 빠르게 처리하도록 개발된 오픈소스인 Kafka와 빅데이터 색인, 저장, 모니터링 오픈소스인 ELK Stack을 사용했다. Kafka 내에 데이터의 분산처리를 위해 3개의 broker를 생성하고 데이터를 kafka에서 ELK Stack으로 보내기 위해 Nifi로 연결했다.

3) 머신러닝을 통한 데이터 분석 및 예측

```
Anaconda Prompt

(base) C:\Users\User>conda --version
conda 4.6.14

(base) C:\Users\User>python --version
Python 3.6.5 :: Anaconda, Inc.

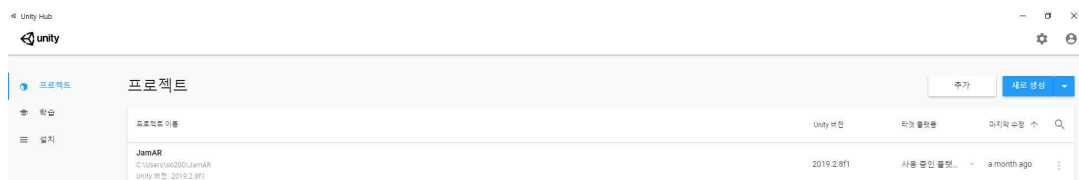
(base) C:\Users\User>
```

< 그림 7. Anaconda 버전 >

- 아나콘다 4.6.14 버전 설치
- Python 3.6.5 버전 설치

아나콘다에서 제공되는 프로그래밍 작업을 보여줄 수 있는 웹 어플리케이션인 Jupyter Notebook을 사용하여 작업했다.

4) AR 어플리케이션



< 그림 8. Unity 버전 >

- Unity 2019.2.8f1 버전 설치

AR 콘텐츠를 만들기 위해서 Unity의 Vuforia Engine을 사용하였다. 하였다. 또한 알림 서비스를 사용하기 위해 Firebase 서버와 연동했다.

3.4.2 구현 결과

1) AI 이용 혼잡도 알고리즘 생성

```

for idx1, row1 in date_data.iterrows():
    queue1.queue = [] #username
    queue2.queue = [] #acctstarttime
    queue3.queue = [] #acctsessiontime
    queue4.queue = [] #place
    queue5.queue = [] #idx

    for idx, row in ap_data.iterrows():
        timestamp = row.acctstarttime
        if row1.date.hour+1 < row.acctstarttime.hour:
            break

        while(date_data.iat[idx1,0] <= timestamp) and (timestamp < date_data.iat[idx1+1,0]):
            if row.username == np.nan:
                break

            queue1.push(row.username)
            queue2.push(row.acctstarttime)
            queue3.push(row.acctsessiontime)
            queue4.push(row.place)
            queue5.push(idx)

            if queue1.size() > 1: #큐만여 1개만 있을때를 위해 2부터 돌아가게
                listt = queue1.queue
                if listt.count(ap_data.iat[idx,0]) == 1:
                    break
                else:
                    i = 0
                    for data in listt:
                        if (ap_data.iat[idx,0] == data) and (queue5.queue[i] == idx):
                            i = i + 1
                            continue
                        elif (ap_data.iat[idx,0] == data) and (queue5.queue[i] != idx):
                            if (ap_data.iat[idx,4] < queue3.queue[i]) and (ap_data.iat[idx,0] == queue1.queue[i]):
                                ap_data.iat[idx,0] = np.nan
                                continue
                            elif ap_data.iat[idx,4] >= queue3.queue[i] and ap_data.iat[idx,0] == queue1.queue[i]:
                                ap_data.iat[queue5.queue[i],0] = np.nan
                                i = i + 1
                    break

```

< 그림 9. 중복 제거 알고리즘 >

열람실마다 AP의 IP 주소를 기준으로 접속자를 나누고 개인당 여러 기기를 사용하는 경우, 내부적으로 데이터가 쌓이는 DB의 설정 오류로 인해 발생한 데이터의 중복을 배제하기 위한 알고리즘을 만들었다. 큐를 사용하여 데이터들을 큐에 넣고 레코드에서 기본키 역할을 하는 사용자ID 칼럼 값이 겹치면 시간 데이터를 비교하여 최초 접속시각이 비슷하면 접속시간이 더 짧은 데이터를 삭제했다.

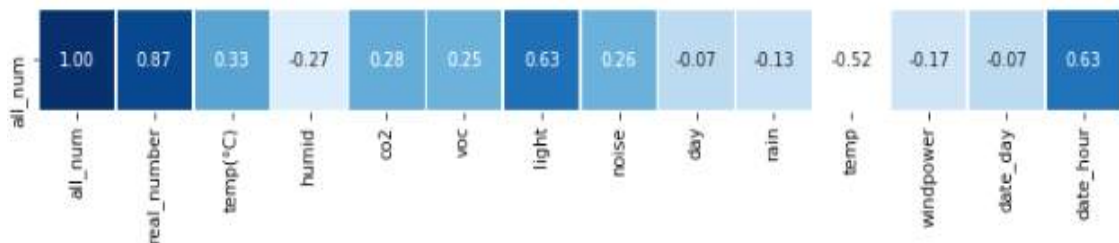
```

import sys
for idx1, row1 in ap_data.iterrows():
    for i in range(int(row1.acctsessiontime//3600 + 1)):
        if(row1.acctstarttime_hour+i<24):
            date_data.loc[(date_data.index.hour==row1.acctstarttime_hour+i) & (date_data.index.day==row1.acctstarttime_day) &
                          (date_data.index.month==row1.acctstarttime_month), 'all_num']+=1
        elif(row1.acctstarttime_hour+i>=24):
            date_data.loc[(date_data.index.hour==(row1.acctstarttime_hour+i)%24) & (date_data.index.day==row1.acctstarttime_day+1) &
                          (date_data.index.month==row1.acctstarttime_month), 'all_num']+=1

```

< 그림 10. 접속자수 도출 알고리즘 >

가공한 Wi-fi 접속 데이터로부터 1시간마다의 접속자수를 도출하기 위해 ‘접속시간’, ‘접속시작시각’ 값 등 여러 칼럼 값을 이용하여 알고리즘을 생성했다.



< 그림 11. 데이터 간 상관관계 >

데이터 분석을 위해서 열람실별 ap 접속자 수, 재실 인원수, 공기질 데이터와 실제 외부 날씨 데이터를 시각화 모듈을 사용하여 상관관계를 확인하였다. ap 접속자 수를 기준으로 시간, light, 외부 기온, 내부 기온 순으로 상관관계가 높았다.

```
# 총 571개의 데이터 중 train 개수 384개 test 개수 187개
n_train = 384

#주어진 특성을 사용하여 평가하고 그래프 그리기
def eval_on_features(features, target, regressor):
    #훈련 /테스트로 나누기
    X_train, X_test = features[:n_train], features[n_train:]
    #타깃값도 나누기
    y_train, y_test = target[:n_train], target[n_train:]
    regressor.fit(X_train, y_train)

    y_pred = regressor.predict(X_test)
    y_pred_train = regressor.predict(X_train)
    plt.figure(figsize=(10, 3))
    print("훈련 세트 R^2: {:.2f}".format(regressor.score(X_train, y_train)))
    print("테스트 세트 R^2: {:.2f}".format(regressor.score(X_test, y_test)))
    plt.xticks(range(0, len(X), 8), x_ticks_name, rotation=90, ha="left")

    plt.plot(range(n_train), y_train, label="훈련")
    plt.plot(range(n_train, len(y_test) + n_train), y_test, '-', label="test")
    plt.plot(range(n_train), y_pred_train, '--', label="훈련 예측")

    plt.plot(range(n_train, len(y_test) + n_train), y_pred, '--', label="테스트 예측")
    plt.legend(loc=(1.01, 0))
    plt.xlabel("날짜")
    plt.ylabel("실제 인원수")

    score = rmsle(y_train, y_pred_train)
    print("Train RMSLE Value: ",score)
    score = rmsle(y_test,y_pred)
    print("Test RMSLE Value: ",score)
```

< 그림 12. 데이터 훈련 및 테스트 >

여러 모델에 학습을 적용하기 위해 그래프와 평가를 섞은 함수를 생성하였다. 평가요소로는 R^2과 RMSLE를 사용하였다. RMSLE값이 0에 가까울수록 정밀도가 높은 값입니다. 입력 값으로 상관관계가 높은 공기질 데이터, ap 접속자 수, 시간 데이터를 사용하였고 출력 값으로 실제 재실 인원수를 사용하여 모델링을 하였다. 그 결과, regression model의 성능을 나타내는 결정계수(R^2) 값이 0.89로 가장

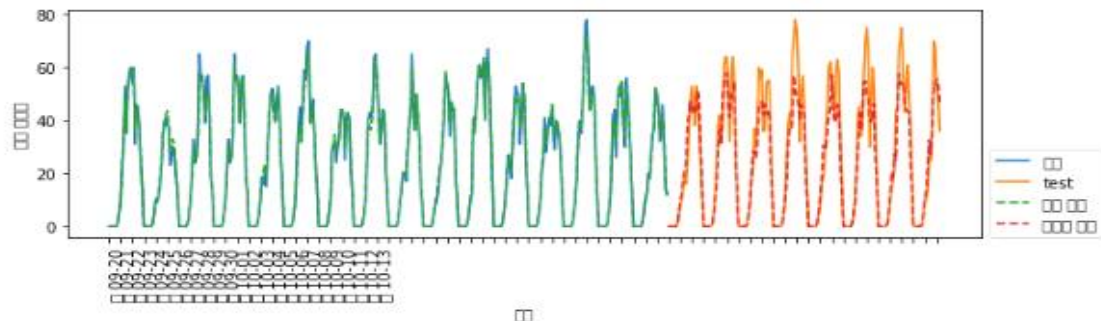
높게 나온 랜덤 포레스트 알고리즘을 선택했다. 결정계수 값이 1로 갈수록 좋은 모델이라고 할 수 있다. 생성한 예측 모델을 통해 1시간 단위로 재실 인원수를 예측했다.

```

1 #다른 feature 추가
2 from sklearn.ensemble import RandomForestRegressor
3 regressor = RandomForestRegressor(n_estimators=100, random_state=0)
4 X_hour_week = np.hstack([result.index.dayofweek.values.reshape(-1,1),
5                           result.index.hour.values.reshape(-1,1),
6                           result.all_num.values.reshape(-1,1),
7                           result.noise.values.reshape(-1,1),
8                           result.light.values.reshape(-1,1),
9                           ])
10 eval_on_features(X_hour_week,y.values, regressor)

```

훈련 세트 R^2 : 0.99
테스트 세트 R^2 : 0.89
Train RMSLE Value: 0.06856035896135955
Test RMSLE Value: 0.19788637538601733



< 그림 13. 랜덤 포레스트 모델 >

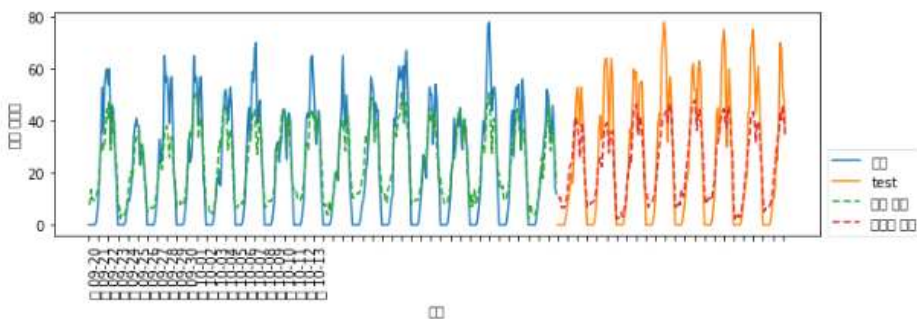
```

ridge_m_ = Ridge()
ridge_params_ = {'max_iter':[3000], 'alpha':[0.01, 0.1, 1, 2, 3, 4, 10, 30, 100, 200, 300, 400, 800, 900, 1000]}
rmsle_scorer = metrics.make_scorer(rmsle, greater_is_better=False)
grid_ridge_m = GridSearchCV(ridge_m_,
                             ridge_params_,
                             scoring = rmsle_scorer,
                             cv=5)

eval_on_features(X_hour_week_onehot,y.values, grid_ridge_m)

```

훈련 세트 R^2 : -1.20
테스트 세트 R^2 : -1.14
Train RMSLE Value: 1.1962380996451991
Test RMSLE Value: 1.1439388762954799



< 그림 14. 릿지 모델 >


```

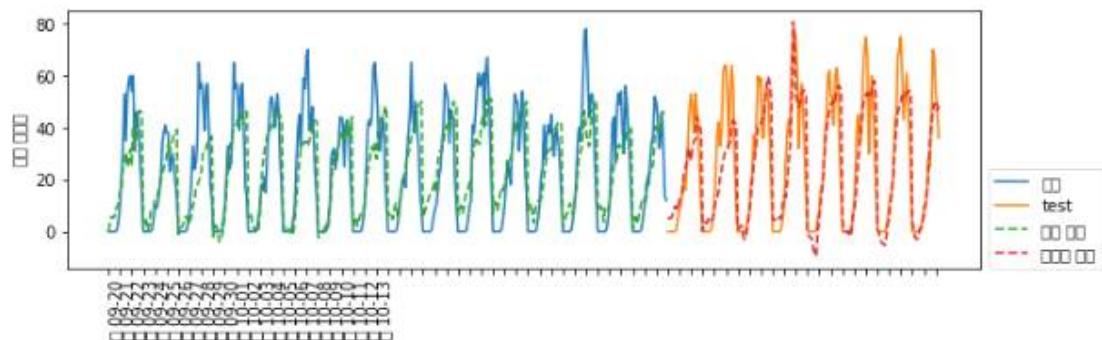
from sklearn.linear_model import LinearRegression, Ridge, Lasso
from sklearn.model_selection import GridSearchCV
from sklearn import metrics
import warnings
pd.options.mode.chained_assignment = None
warnings.filterwarnings("ignore", category=DeprecationWarning)

# 선형회귀 모델을 초기화
lModel = LinearRegression()

eval_on_features(X_hour_week,y.values, lModel)

```

훈련 세트 R^2 : 0.57
 테스트 세트 R^2 : 0.72
 Train RMSLE Value: nan
 Test RMSLE Value: nan



< 그림 15. 선형회귀 모델 >

```

lasso_m_ = Lasso()

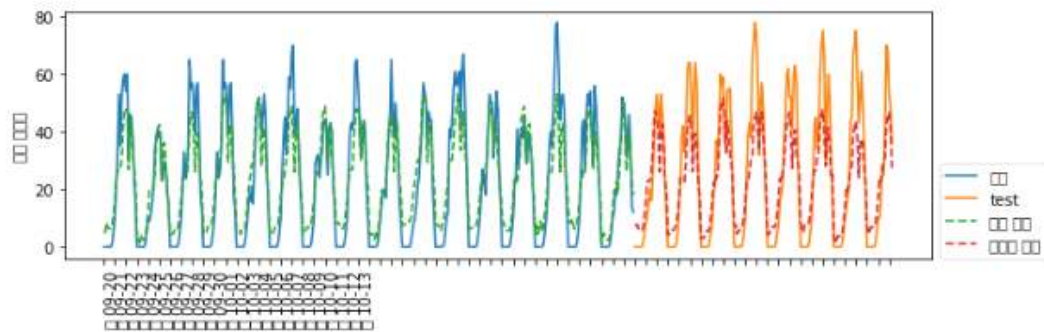
alpha = 1/np.array([0.1, 1, 2, 3, 4, 10, 30,100,200,300,400,800,900,1000])
lasso_params_ = { 'max_iter':[3000], 'alpha':alpha}

grid_lasso_m = GridSearchCV( lasso_m_,lasso_params_,scoring = rmsle_scorer,cv=5)

eval_on_features(X_hour_week_onehot,y.values,grid_lasso_m )

```

훈련 세트 R^2 : -1.04
 테스트 세트 R^2 : -1.04
 Train RMSLE Value: 1.0401475261524313
 Test RMSLE Value: 1.0429366202866293

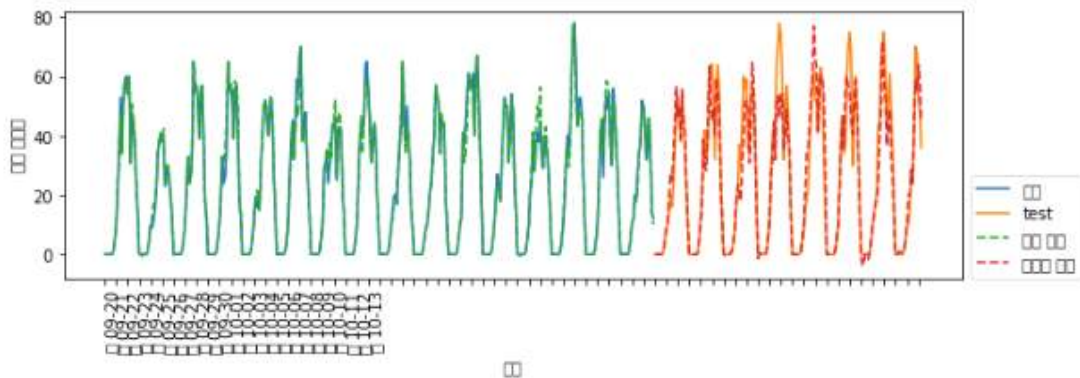


< 그림 16. 라쏘 모델 >

```
from sklearn.ensemble import GradientBoostingRegressor
gbm = GradientBoostingRegressor(n_estimators=4000, alpha=0.01);

eval_on_features(X_hour_week_onehot,y.values, gbm)
```

훈련 세트 R²: 0.98
테스트 세트 R²: 0.90
Train RMSLE Value: 0.10559772816545177
Test RMSLE Value: nan



< 그림 17. 그라디언트 부스트 모델 >

```
predict.loc[predict.predict_number<15, 'predict_congest'] = '1'
predict.loc[(predict.predict_number>=15) & (predict.predict_number<40), 'predict_congest'] = '2'
predict.loc[(predict.predict_number>=40) & (predict.predict_number<60), 'predict_congest'] = '3'
predict.loc[(predict.predict_number>=60) & (predict.predict_number<100), 'predict_congest'] = '4'
predict.loc[predict.predict_number>=100, 'predict_congest'] = '5'
```

< 그림 18. 혼잡도 분류 알고리즘 >

예측된 재실 인원수를 혼잡도 5단계로 나누기 위해서 군집화 알고리즘을 사용하였다. 이 그래프를 바탕으로 재실 인원수가 0 ~ 15이면 혼잡도 1, 15 ~ 40이면 혼잡도 2, 40 ~ 60이면 혼잡도 3, 60 ~ 100이면 혼잡도 4, 100 이상이면 혼잡도 5로 기준을 정하였다. 실제 재실 인원수와 예측된 재실 인원수를 특정 기준으로 혼잡도를 5단계로 분류한 후 비교해 본 결과 일주일의 테스트 데이터 168개 중 131개가 같았고 약 78%의 정확도가 나왔다.

2) AR 어플리케이션

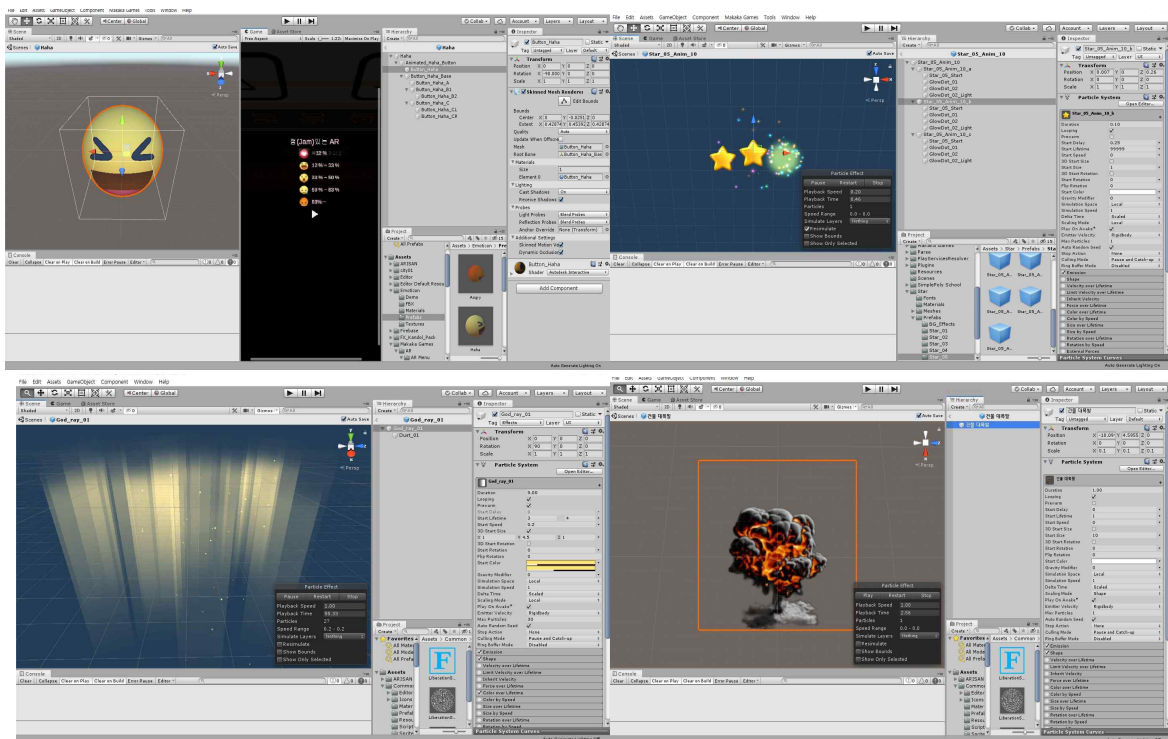
열람실 혼잡도 분류

	5단계	약 100명 ~ (열람실 좌석수 120석 기준) 83% ~
	4단계	약 60명 ~ 99명 (열람실 좌석수 120석 기준) 50% ~ 83%
	3단계	약 40명 ~ 59명 (열람실 좌석수 120석 기준) 33.3% ~ 50%
	2단계	약 15명 ~ 39명 (열람실 좌석수 120석 기준) 12.5% ~ 33.3%
	1단계	~ 약 14명 (열람실 좌석수 120석 기준) ~ 12.5%

건물 상태도 분류

	Jam	도서관이 매우 혼잡하여 추가적인 공간확보 요구 상태
	One Star	도서관이 혼잡한 상태 (건물 전체 포화도 80% ~ 95%)
	Two Star	도서관의 일반적인 상태 (건물 전체 포화도 40% ~ 80%)
	Three Star	도서관이 쾌적한 상태 (건물 전체 포화도 ~ 40%)
	Open	추가적으로 개방된 공간에 나타나는 효과

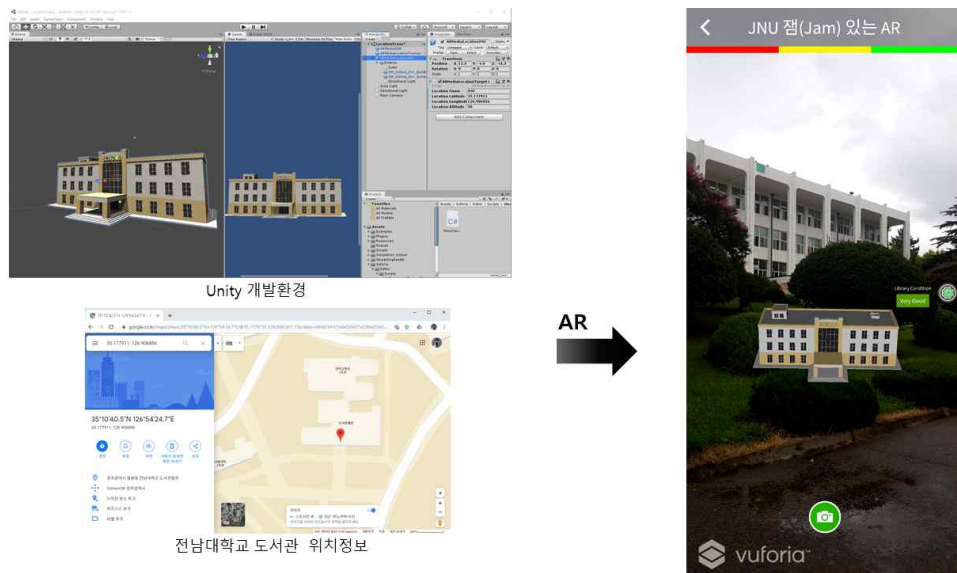
< 그림 19. AR 콘텐츠로 혼잡도 분류 표현 >



< 그림 20. AR 콘텐츠 개발 모습 >

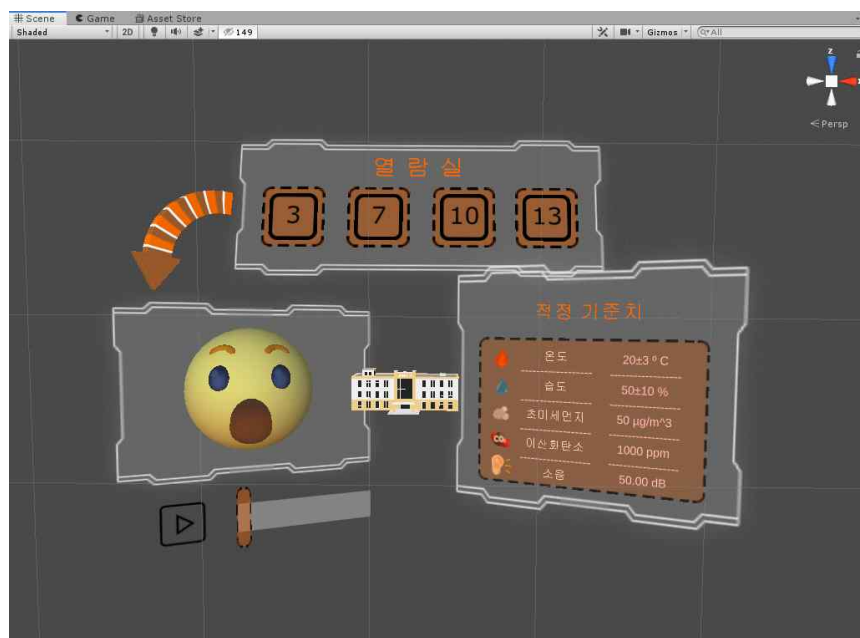
열람실별 재실인원에 따른 혼잡도를 바탕으로 건물의 전체적인 상태를 나타내기 위해 건물 전체 포화도를 기준으로 4단계로 분류했다. 포화도가 0 ~ 40%이면 별 3개, 40% ~ 80%이면 별 2개, 80% ~ 95%이면 별 1개, 95% ~ 100%면 Jam 단계로 기준을 정했다.

재실인원에 따른 5단계의 혼잡도를 나타내는 하트, 웃음, 놀람, 울음, 화냄의 5가지 3D Emoticon Animation과 건물의 전체적인 혼잡도를 표현하기 위해 Unity의 Particle System을 이용하여 건물위에 별 모양 이펙트가 나타나게 하는 Prefab들을 만들었다. 건물이 추가적으로 개방된 상황을 나타내기 위한 별빛이 내리는 이펙트, 건물이 매우 혼잡하여 추가적인 공간 확보를 요구하는 상황에서 나타나는 폭발 이펙트를 개발하였다.



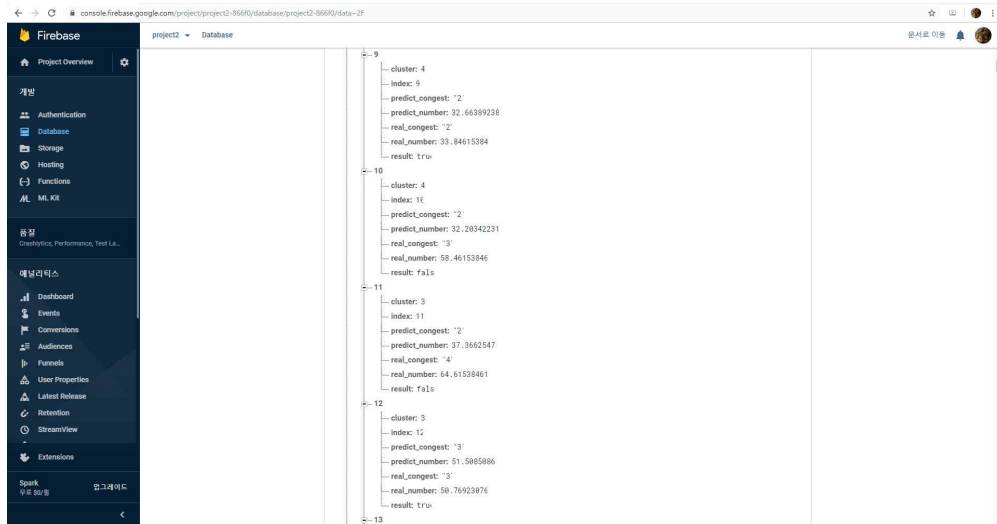
< 그림 21. 3D Building Object 생성 및 GPS 위치 정보 삽입 >

전남대학교 도서관, 정보전산원, 공대7호관 3가지 건물들을 3D Object로 제작하고 Google Map API를 이용하여 GPS 위치 정보(위도, 경도, 고도)를 삽입했다.



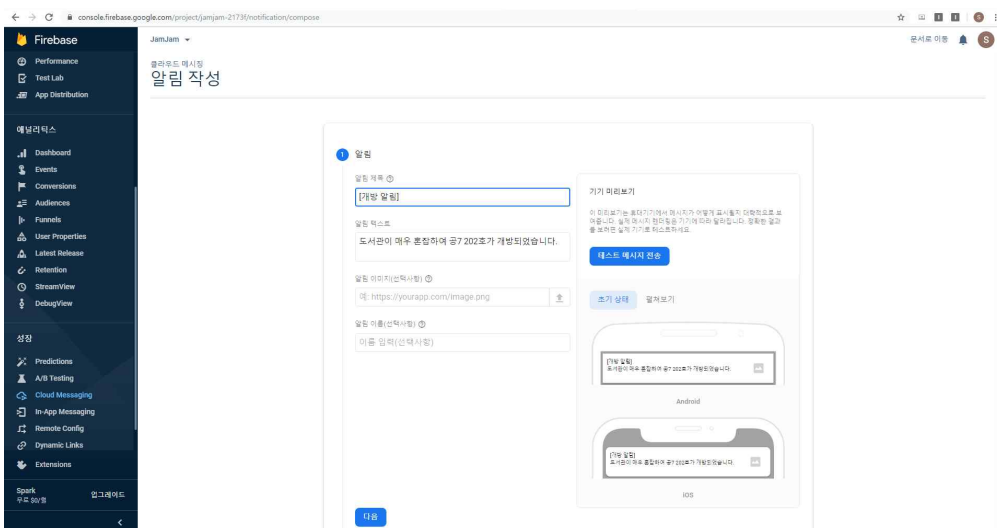
< 그림 22. AR GUI 모습>

오브젝트 클릭시 나타나게 될 전체 콘텐츠를 담은 메뉴를 GUI 형태로 제작했다. 환경부가 정한 도서관 등의 실내오염물질 유지기준 및 권고기준(2018.12월 기준)에 따른 초미세먼지, 이산화탄소의 적정기준치($50\mu\text{g}/\text{m}^3$, 1000ppm)를 제시하고 소음·진동 관리법에 따른 도서관 소음규제 기준치인 50dB를, 겨울철 실내 온·습도의 표준범위(온도 $20\pm 3^\circ\text{C}$, 습도 $50\pm 10\%$)을 제공하여 사용자가 데이터를 판단하는데 도움이 되도록 했다.



< 그림 23. Firebase DB >

머신러닝을 통해 분류한 혼잡도 값과 공기질 데이터 값은 Json파일로 Firebase에 구축한 Database에 저장된다. Firebase와 Unity와 연동하여 혼잡도와 공기질을 불러올 수 있게 했다.



< 그림 24. Firebase Cloud Messaging 시스템 >

Firebase의 cloud Messaging 시스템을 이용하여 도서관이 매우 혼잡해 다른 단과대학의 강의실을 개방한 경우 사용자에게 알림이 가도록 했다.

4. 실험 및 분석

4.1 실험 환경

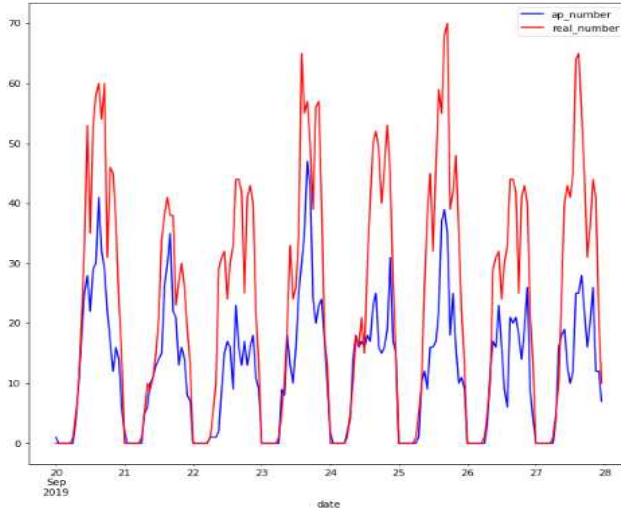
	A	B	C	D	E	F	G	H
17		09/16 월	09/17 화	09/18 수	09/19 목	09/20 금	09/21 토	09/22 일
18	9:30	19	-	16	12	15	10	-
19	10:30	18	12	23	17	21	10	-
20	11:30	22	18	16	6	13	11	-
21	12:30	18	15	15	14	18	11	-
22	13:30	20	-	24	-	22	14	-
23	14:30	27	-	29	-	22	12	-
24	15:30	37	27	27	25	26	14	-
25	16:30	32	26	27	22	24	14	-
26	17:30	17	17	19	22	21	12	-
27	18:30	17	13	9	17	21	11	-
28	19:30	23	18	26	16	20	13	-
29	20:30	26	12	18	22	20	15	-
30	21:30	21	17	16	14	16	10	-
31	22:30	12	13	11	10	12	5	-
32								
33		09/23 월	09/24 화	09/25 수	09/26 목	09/27 금	09/28 토	09/29 일
34	9:30	11	-	15	12	15	9	-
35	10:30	15	6	21	13	22	12	7
36	11:30	11	8	22	13	23	9	11
37	12:30	12	12	17	11	11	10	9
38	13:30	15	-	22	-	13	14	14
39	14:30	27	-	26	-	18	18	14
40	15:30	23	27	25	22	23	18	14
41	16:30	23	27	28	22	24	25	17
42	17:30	15	12	19	17	28	18	22
43	18:30	13	13	10	12	21	16	8
44	19:30	21	22	17	17	20	17	17
45	20:30	28	21	14	22	19	16	15
46	21:30	17	16	13	11	14	13	14
47	22:30	10	8	10	7	5	10	8
48								
49		09/30 월	10/01 화	10/02 수	10/03 목	10/04 금	10/05 토	10/06 일
50	9:30	17	-	9	14	14	7	-
51	10:30	23	24	10	20	18	19	10
52	11:30	17	27	14	19	15	20	15
53	12:30	11	18	12	-	11	8	8
54	13:30	26	-	18	-	16	14	13
55	14:30	19	-	22	38	15	16	15
56	15:30	33	22	29	38	16	18	18
57	16:30	31	28	30	39	15	21	24
58	17:30	27	17	18	25	13	22	29
59	18:30	21	21	24	20	19	16	21
60	19:30	18	19	7	35	25	14	13
61	20:30	29	16	24	33	23	21	14
62	21:30	20	16	17	21	17	17	14
63	22:30	20	9		16	13	14	21
64								
65		10/07 월	10/08 화	10/09 수	10/10 목	10/11 금	10/12 토	10/13 일
66	9:30	14	-	15	16	15	9	-
67	10:30	13	18	21	14	20	13	10
68	11:30	14	13	25	15	25	15	14
69	12:30	13	22	20	16	17	18	11
70	13:30	24	-	27	-	20	19	17
71	14:30	24	-	28	-	37	26	20
72	15:30	29	25	41	38	42	25	25
73	16:30	24	27	37	30	40	27	24
74	17:30	27	18	34	28	35	27	23
75	18:30	11	12	27	21	19	16	15
76	19:30	29	21	33	31	22	27	19
77	20:30	35	25	38	34	27	35	21
78	21:30	28	23	32	27	21	33	18
79	22:30	10	17	17	12	16	15	11

< 그림 25. 실제 재실인원 카운트 데이터 >

예측 알고리즘에 대한 정확도를 높이기 위해 약 4주 동안 1시간마다 직접 도서관 열람실의 재실 인원수를 측정하여 비교 값을 생성하였다.

4.2 실험 결과

1) 자료 간 상관관계 분석



< 그림 26. AP 접속자 수, 재실 인원수 비교 >

위 그래프는 일주일 동안 열람실의 AP 접속자 수와 실제 재실 인원수의 그래프이다. 빨간색 그래프는 재실 인원수를 나타내고 파란색 그래프는 무선랜 접속자 수를 나타낸다. 두 그래프 간 값의 차이가 보이지만, 시간대별 재실 인원수와 무선랜 접속자 수가 서로 비슷한 흐름을 갖는다. 따라서 이 결과를 바탕으로 무선랜 접속자 수와 재실 인원수는 상관관계가 있을 것으로 추정하였다.

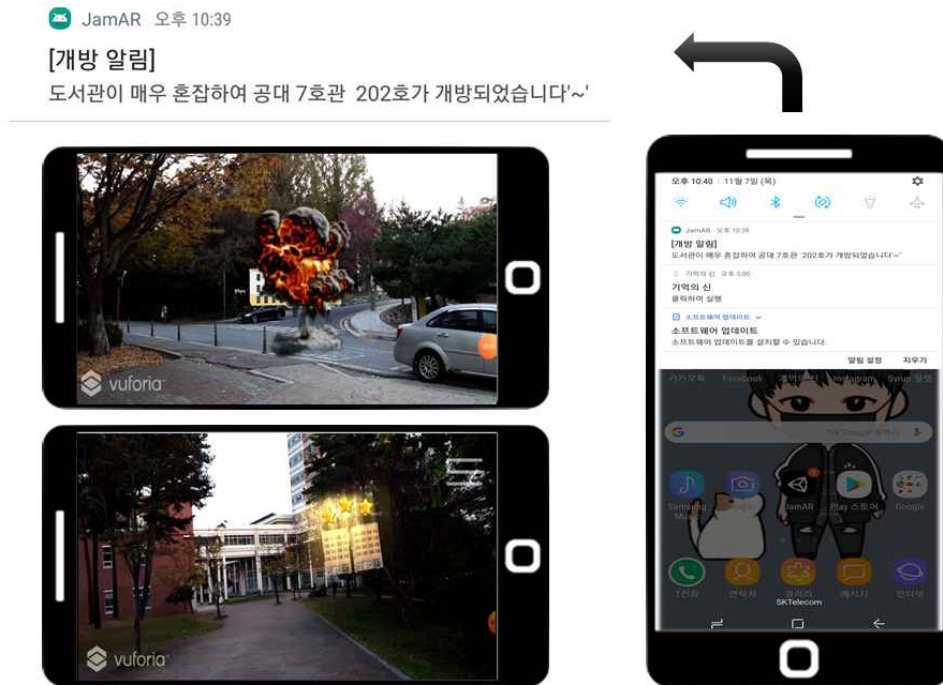
무선랜 접속자 수와 재실 인원수 간에 상관관계를 통계적으로 확인하기 위해 상관분석에서 가장 많이 사용하는 Pearson의 상관계수를 활용하여 변수 간 관련 유무와 방향, 정도를 확인하였다. Pearson의 상관계수에서 정도는 절댓값으로 표현되는 수치를 의미하며 일반적으로 0.7 이상일 경우 아주 높은 상관관계라고 하고, 0.4 이상이면 다소 높다고 할 수 있다. 방향에서는 수치의 부호가 양의 부호일 경우에는 한 변수가 증가함에 따라서 다른 변수도 증가함을 말하고 부호가 음의 부호일 경우에는 한 변수가 증가함에 따라서 다른 변수는 감소하는 것을 의미한다(강주희 2016).

< 표 6 > 실제 인원수에 관한 자료 간 상관관계

상관관계	all_num	temp (in)	humid	co2	voc	light	noise	rain	rain?	temp (out)	windpower	hour
real_number	0.88	0.60	-0.35	0.33	0.29	0.61	0.84	-0.12	-0.10	-0.56	-0.22	0.72

< 표 6 >에 의하면 위에서 추정한 대로 무선랜 접속자 수와 재실 인원수가 0.88로 높은 상관관계가 있음을 알 수 있다. 다음으로 소음, 시간, 빛 세기, 내부 온도, 외부 온도 등의 순으로 상관관계가 높았다.

2) AR 결과물 실험 결과



< 그림 27. 이펙트 구현 및 알림 서비스 구동 모습 >

Apk 파일을 통해 어플리케이션을 다운받아 작동되는지 확인한 결과로 GPS 값이 불안정하여 다른 위치에 뜨는 것을 확인하였고 수차례 세부 수정하여 적절한 위치 값을 도출했다. 건물 오브젝트 클릭시 정상적으로 메뉴가 작동하는지 확인하였고 JSON 파일의 데이터가 제대로 나타나는지 확인했다.

도서관 혼잡도가 95%가 넘어가는 값의 JSON 파일을 입력하여 폭발 이펙트가 뜨는지 확인했고 강의실 개방 시 알림 전송이 전송 되는지, 알림 전송시 개방된 강의실 오브젝트와 이펙트가 나타나는지 확인했다.

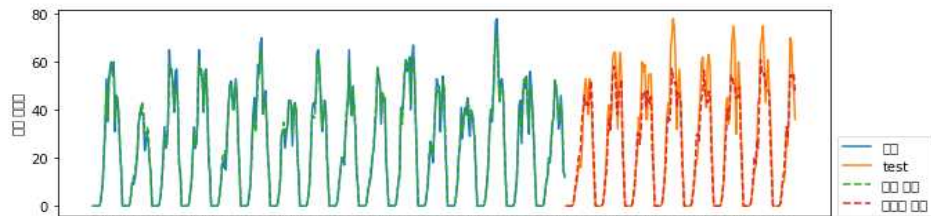
4.3 실험 결과 분석

본 연구의 결과로 무선랜을 활용한 혼잡도 제공 시스템을 갖춘 어플리케이션을 만들었다. 결과는 크게 혼잡도 분석 및 예측 알고리즘과 도서관 실시간 혼잡도 정보 시스템 AR 어플리케이션으로 나누어진다.

1) 혼잡도 분석 및 예측 알고리즘

학습할 때 연속적인 숫자(인원수)를 예측하기 위해 회귀 알고리즘인 랜덤포레스트 회귀를 사용하여 1시간 단위의 예측 재실 인원수를 얻었다. 입력 값으로 상관관계가 높은 것을 조합하여 실험하였고 실험 결과 중에서 가장 성능이 좋은 모델을 선택하였다. 따라서 최종 모델의 입력값은 요일, 시, 내부 온도, 무선랜 접속자 수이며, 출력값은 재실 인원수로 학습시켜 모델을 생성하였다.

훈련 세트 R^2 : 0.99
테스트 세트 R^2 : 0.90
Train RMSLE Value: 0.07230423103226145
Test RMSLE Value: 0.1943087384402291



< 그림 28. 인원수 예측 결과 >

그 결과, 회귀모델의 성능을 나타내는 값인 결정계수(R^2) 값이 테스트 자료에서 0.90, RMSLE 값이 0.19가 나왔다. 결정계수 값은 1로 갈수록 좋은 모델이며 RMSLE는 0으로 갈수록 좋은 모델이라 할 수 있다. 그리고 생성한 예측 모델을 통해 1시간 단위로 재실 인원수를 예측하였다.

2) 도서관 실시간 혼잡도 정보 시스템



< 그림 29. 실제 어플리케이션 구동 모습 >

Unity의 Build기능을 이용하여 AR 콘텐츠, 도서관 실시간 혼잡도 정보, Firebase Cloud Messaging 기능 등을 포함한 JamAR.apk를 생성했다. 이 AR 어플리케이션을 통해 사용자들이 간단히 서비스를 제공받을 수 있다.

5. 결론 및 향후과제

본 연구의 목적은 무선랜 접속자료를 활용하여 전남대학교 도서관 혼잡정보를 제공 시스템 개발하는 것이며, 이 연구의 결론으로 다음과 같은 세 가지를 제시한다. 첫째, 무선랜 데이터를 가공하여 혼잡도를 계산하여 정보제공 시스템을 구축하였다. 둘째, 애플리케이션을 통해 교내 구성원에게 실시간 혼잡정보를 제공하는 체계가 구축 가능함을 확인하였다. 셋째, 실시간 정보뿐만 아니라, 향후 혼잡도 예측이 가능함을 확인하였는데 상관관계수 0.90, RMSLE 0.19로 매우 유의미한 예측성을 지니고 있음을 확인했다.

이 연구는 전남대학교 도서관 무선랜 데이터를 활용하여, 혼잡도 파악이 가능함을 확인하였다. 이 연구의 결과가 지닌 의미를 선행연구 검토 결과와 관련하여 논의한다. 첫째, 이 연구는 무의미하게 소진되어오던 교내 공공 자료를 활용하였다는 점에서 중요한 가치를 가진다. 선행연구에서도 마찬가지로 전남대학교 내 무선랜 데이터를 활용하여 유의미한 정보를 창출 해냈으나(Yanying Gu, 2009; 문준환 외, 2019; 김포타임즈, 2019년 12월 6일), 이 연구에서는 기존의 활용과는 다르게 도서관에서 혼잡정보를 추출하는 데 활용하였다. 둘째, 선행연구에서는 제공하는 장소에 따라 서로 다른 방식으로 혼잡도를 산정하였고 산출한 혼잡정보를 사람들에게 제공하여 유용한 정보가 됨을 알 수 있었다(홍보현, 2019; 김성민, 2018; 한여희, 2018). 따라서 이 연구에서도 마찬가지로 도서관에 맞는 방식으로 혼잡도를 정의하였고 도서관에서도 혼잡정보가 유의미한 정보가 될 것으로 판단하였다.

한 편, 본 연구가 지니는 한계점은 다음과 같다. 첫째, 무선랜 신호 세기 자료를 확보하지 못해 사용하지 못했으나 이를 활용한다면 더 정확한 혼잡정보를 제공할 수 있을 것이다. 둘째, 자료 수집상의 오류와 한계로 자료 수집을 한 달 정도밖에 하지 못했다. 셋째, 본 연구는 도서관에 맞춰진 모델을 생성하였다. 따라서 이를 보완하기 위해 다음의 후속 연구가 필요하다. 첫째, 무선랜 신호 세기 자료를 수집하여 더 정확한 혼잡도를 파악한다. 둘째, 장기간의 자료를 수집하여 1년 이상의 자료로 학습 모델을 구축한다. 셋째, 도서관뿐만 아니라 다른 건물에도 적용될 수 있는 이식성 있는 모델을 구축한다.

6. 참고문헌

1. 김감영·이건학 (2016). 이동통신 빅데이터를 이용한 현재인구 추정과 개선방안 연구. 한국도시지리학회지, 제19권 2호 181-196.=
2. 김성민 (2018). 도시철도 객실혼잡정보 제공에 따른 승객 분산효과에 관한 연구. 한국교통대학교 교통대학원 석사학위 논문.
3. 문준환 외(2019). 빅데이터를 활용한 스마트 관광 도시 사례 분석 연구: 제주특별자치도 관광객 데이터를 중심으로. Information systems review, VOL.21 NO.2: 1-27.
4. 오세웅 (2017). 로지스틱 회귀모형과 랜덤포레스트를 혼합한 변수선택법에 기반하여 의사결정나무를 이용한 외래 관광객 만족도 분석. 한양대학교 대학원 석사학위 논문.
5. 전태준 (2011). 사회적 수용력 연구에 있어 혼잡지각 관련연구 동향분석. 한국여가 레크레이션 학회지, 제35권 제3호.
6. 정용국 (2015). 무선랜 기반의 실내 위치 추적 시스템 구현에 관한 연구. 공주대학교 대학원 석사학위 논문.
7. 조나혜·강영옥 (2016). 로그데이터의 시공간 데이터마이닝 및 시각화 연구동향. 한국지도학회지, 제16권 제3호, 15-27.
8. 한여희 (2017). 교통정보 빅데이터 기반의 시공간 혼잡지표 개발. 서울시립대학교 대학원 박사학위 논문.
9. 홍보현 (2019). 공공도서관 공간 이미지에 대한 공간적 특성과 비공간적 특성의 영향 연구. 연세대학교 대학원 석사학위 논문.
10. Breiman. (2001). random forests. machine learning, 45(1): 5-32.
11. Eroglu, S., & Harrell, G. D. (1986). Retail crowding: Theoretical and strategic implications. Journal of Retailing, 62(4), 346-363.
12. Stokols, D. (1993). On the distinction between density and crowding : Some implication for future research, Psychological Review, Vol.79, 275-278.
13. Yanying Gu, Anthony Lo (2009). A Survey of Indoor Positioning System for Wireless Personal Networks, IEEE Communications Surveys & Tutorials, VOL. 11, NO. 1, First Quarter.
14. 김포타임즈 2019년 12월 6일. 김포시, 빅데이터 적극 활용 “똑똑한 도시로 간다”.
15. 전남대학교 (2018). 대학종합만족도조사에 따른 교육 서비스 개선 계획.
16. 환경부 (2018). 실내오염물질 유지기준 및 권고기준.
17. 환경부 (2018). 실내 소음·진동 관리법.