

프로젝트명

미허가 침입 탐지 모델 구축

2 조 팀원

20165079 김영빈

20165545 조현우

20161569 최경식

프로젝트 목적

- 건물 내 사용되지 않는 공간에 대한 정보를 수집하고, 이러한 정보와 센서를 활용하여 이상 상황을 탐지할 수 있는 모델 구축한다.
- 탐지한 이상 상황을 직관적으로 시각화하여 보안 인력에게 전달함으로써 보안 업무의 효율을 증대하고 신속한 대처를 가능하게 한다.

프로젝트 목표 및 수행 효과

- 중앙대학교 310 관을 대상으로 파일럿 테스트를 진행한다.
- 사이버 물리 보안 시스템의 근간을 이루는 센서의 탐지 데이터를 자세하게 공부하고 실제로 활용해볼 수 있다.
- 실습 시간에 학습한 웹 크롤링과 파이썬에서 제공하는 기능들을 학습해 데이터를 정제해볼 수 있다.
- 기존의 물리 시스템에 모델링을 적용하여 보안 강화에 기여할 수 있다.

구조

/Arduino

└── Motion_Door.ino // Arduino Code

/Crawling

└── /Data

| └── ... // Json Data

└── getTimeTable.py // Crawling Code

/AWS

└── /DynamoDB // CRUD function

| └── delete.js

| └── makeDB.js

| └── scan.js

| └── update.js

└── /Lambda

 └── node_modules // Basic modules

 | └── ...

 └── index.js // AWS Lambda function

 └── package-lock.json

/server

└── /cps_sec

| └── /cps_sec

| | └── ...

| └── /cau

| | └── static // css & js

| | └── templates/cau // html

| | └── urls.py // Url routing

| | └── models.py

| | └── views.py // function

| | └── ...

| └── manage.py // app start

| └── ...

└── /venv

 └── ...

Json Data 구성

310 관강의실.json - 310 관 내 강의실로 사용되는 호실 데이터

timeTables.csv - 중앙대학교 모든 강의 시간표 csv 형식의 파일

timeTable_subject_310 관.csv - 310 관에서 열리는 모든 강의 시간표 csv 형식의 파일

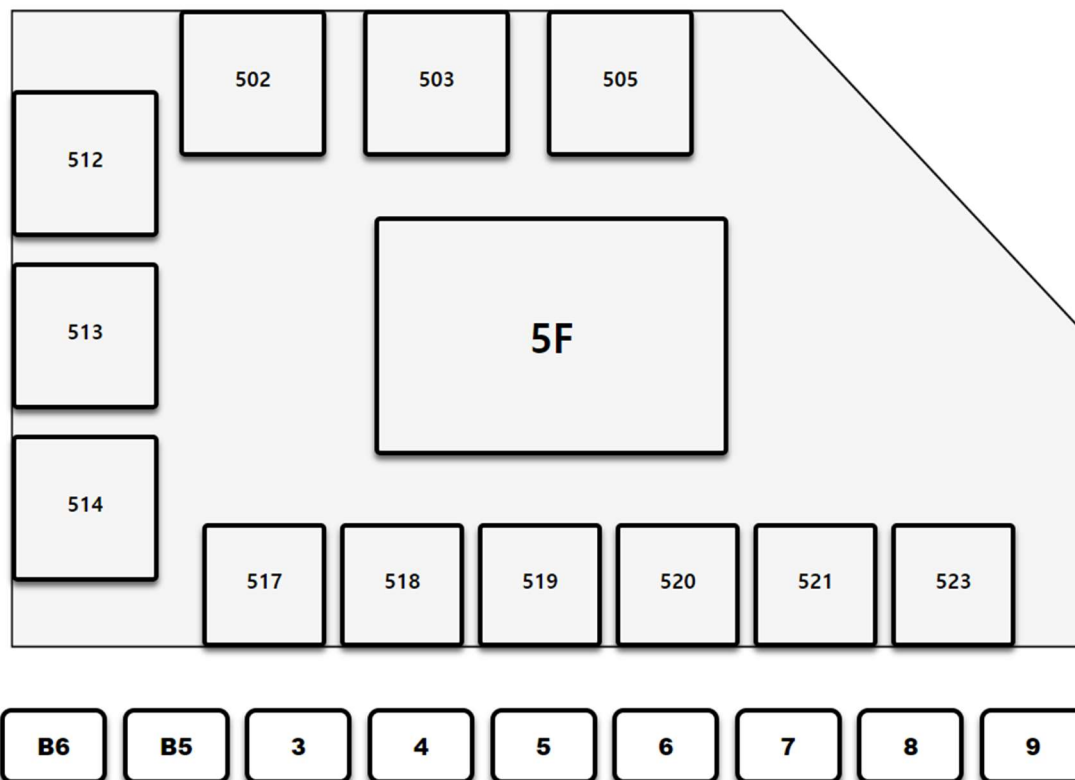
강의시간표.json - timeTables.csv convert to json

강의시간표_정리.json - 강의시간표.json 에서 쓸모없는 열을 제거한 데이터

최종강의시간표.json - 강의시간표_정리.json 데이터에서 시간 형식으로 표현된 데이터를 정제한 데이터

기본적인 프로젝트 동작 화면

중앙대학교 310 관 5 층 구현



(예시 503 호)

- 사용중인 강의실



- 1 단계 경보 (motion or door)



- 2 단계 경보 (motion and door)



- 비어있는 강의실



프로젝트에 사용한 툴

- 파이썬
 - 데이터 크롤링 (BeautifulSoup)
 - 서버 구축 (Django)
- Excel
 - 크롤링 Data 정제
- Github
 - 코드 공유 및 버전 관리
- AWS
 - AWS Lambda 및 DynamoDB 를 통한 서버 구현 및 DB 구성
- NodeJS
 - AWS Lambda 서버 환경 구현
- Notion
 - 프로젝트 결과물 정리
- Zoom
 - 화면 공유, 원격 제어를 통한 Untact 팀플
- HangOut
 - 화상 전화를 이용한 Untact 팀플

프로젝트를 통해 배운 점

- 조건에 맞는 주제를 선정하는 방법
- 파이썬 웹 크롤링 BeautifulSoup Library 의 사용법
- 엑셀을 통하여 데이터를 정제하는 방법

- 아두이노 보드의 사용법
- 센서에서 수집한 데이터를 서버로 전송하는 방법
- 파이썬 Django 서버의 구축 방법
- Django 의 Models, Views, Templates 모델 메커니즘, 구현방법
- Django 의 Database 관리방법
- AWS 클라우드 서비스 사용방법(Lambda 및 DynamoDB)
- AWS API gateway 를 설정하면서 CORS 를 배우면서 GET, PUT, POST, OPTION 메서드에 관한 이해
- 클라이언트와 서버가 동작하는 과정, 웹 브라우저에서 AJAX 통신에 대한 깊은 이해
- Git 의 사용법

시도했는데, 실패했던 경험

• 현 포탈 크롤링 실패

학교 포탈의 경우, 로그인을 통해 접속해야하며, 강의 계획서 검색 부분이 javascript 로 구현되어 있기 때문에 셀레니움을 활용한 크롤링을 먼저 시도해보았다. 셀레니움을 통해 로그인하고, 강의계획서 검색 탭을 여는 것까지는 성공했지만, 여기에서 강의 시간 데이터를 얻어오는 건 불가능했다. 그 이유로는 검색어를 2 자 이상 입력해야하는 조건 때문인데, 모든 강의 시간표 데이터를 가져오기 위해선 모든 데이터에 대해 참인 쿼리를 서버로 요청해야한다. 하지만 sql 인젝션 형태는 학교 포탈에서 유효하지 않았고, 결국 셀레니움을 활용해서는 모든 강의 시간표 데이터를 가져올 수 없었다. Request 모듈을 활용한 방법 또한 Request Header 에 어떠한 정보를 담아야 할 지 알 수 없었고, 결국 팀원이 구 포탈 사이트를 알아내어 데이터 크롤링에 성공하였다.

• SSE(Server-Sent-Events) 구현 실패

웹으로 결과를 보여주겠다는 아이디어의 첫 시나리오는 아두이노의 데이터가 1 로 인식 되었을 때, 즉 이상 상황이 발생했다고 인지되었을 때, 서버로 데이터를 보내고, 서버는 해당 요청을 받았을 때, 클라이언트로 SSE 를 넘겨주려고 했었다. 그리고 클라이언트 측에선 그러한 이벤트를 받을 경우, 화면에서 해당 강의실의 색깔을 바꿔주는 방식으로 시나리오를 계획했었다. 하지만 장고에서 SSE 를 보내는 방법과 클라이언트에서 해당 이벤트를 받았을 때, 동적으로(reload 하지 않고) 색깔을 변경해 줄 수 있는 부분이 없었기 때문에 해당

시나리오는 무산되었으며, 아두이노에서 지속적으로 데이터를 전송하여 DB 를 업데이트 하고, 클라이언트 환경에서도 10 초에 한 번씩 리로드하여 업데이트된 정보를 받아와 강의실에 컬러를 입혀주는 방식으로 구현했다.

• 데이터 정제 자동화 실패 및 실시간 사용 중인 강의실 데이터의 정확성 확보 실패

강의 계획표에 존재하는 강의실과 강의 시간 정보의 경우, 데이터의 형태가 일정하지 않기 때문에 예외가 많았으며, 이러한 예외를 정규표현식을 통해 처리해보려했지만, 정규표현식을 실제 적용하는 데 난이도가 있어서 엑셀을 통해 수작업으로 데이터를 정제해 주었다. 그리고 강의 시간 정보에는 교시 형식(월 1,2,3)으로 표현된 강의가 있으며, 시간 형식(월<10:30 ~ 11:45>)으로 표현된 강의가 존재하는데, 시간 형식으로 표현된 강의의 경우, 30 분 단위로 끊어 이를 월(a,b,c)와 같이 바꾸어서 처리를 해주었다. 하지만 이러한 방식은 실제 현재 시간에 따른 강의실 정보를 가져오는 데 있어 정확하지 않은 정보를 만들어내기 때문에 아쉬웠다.

향후 발전 방향

- 전력 사용량, 진동 감지 센서 등 사람의 존재를 판단하는 지표를 추가 한다면 탐지의 신뢰성을 높일 수 있을 것이다.
- 위험도 판단 알고리즘을 고안하여 지표에 가중치를 부여한다면 표현되는 결과의 신뢰성을 높일 수 있을 것이다.
- 시큐어 코딩이 되지 않았으며, 프로젝트의 구현에만 신경쓰다보니, 코드의 효율성이 떨어지는 부분이 존재한다.
- 클라이언트 단에서 요청이 이루어지도록 하지 않고 서버 측으로 숨겨야 한다.
- 예외 상황을 적용하는 메커니즘을 구현한다 → 강의가 휴강 되었거나 연장되는 경우, 강의가 진행되는 호실이 바뀐 경우 → 학교 측에서 데이터 입력의 형식을 정해주고, 이를 강의 계획 파트에 업데이트 하도록 한다면, 좀 더 쉽게 데이터 활용이 가능할 것이다.

다른 조 대비 내세울 수 있는 점

- CPS 의 근간을 이루는 센서 데이터를 활용하고, 실습시간에 배운 크롤링을 통해 보안에 접목시켰다. 모든 발표 중 해당 강의와 가장 밀접한 관계를 가진 발표 주제라고 생각한다.

- 1 주차 발표팀임에도 불구하고 기한 내에 프로젝트를 모두 완성하였다. (센서데이터부터 시각화까지 모두 구현) 오히려 시각화의 경우 본래 계획에 없었으나 프로젝트 목적을 위해 추가해야 한다 판단하였고, 구현까지 완료하였다.
- 가상의 데이터가 아닌 실제 데이터를 사용함으로써, 실제 사물 시스템의 동작에 대해 더 깊게 이해할 수 있었으며, 비슷한 다른 프로젝트로 발전시킬 때 이식률이 좋을 것이다.
- 팀원 3명 모두 관련 지식이 전무했던 아두이노, 장고, AWS 를 학습하였다. 또 단순히 각각을 학습하는 것에서 그치지 않고 각 기술을 연결하는 방법을 공부하여 하나의 프로그램으로 동작하게 만들었다.
- 모든 팀원들의 의사 소통이 원활하게 잘 이루어 졌고 task 의 분배가 잘 이루어져 협업의 효율성이 높았다.

버전 정보

Server

사용 기술	버전(기기명)
Python	3.8.1
Django	3.0.7
pip	20.1.1

Arduino

사용 기술	버전(기기명)
Arduino IDE	1.8.5
Python	2.7.13
Arduino WiFi board	WeMos D1 R2

AWS

사용 기술	버전(기기명)
Node.js	12.16.1
npm	6.13.4
moment	2.24.0
moment-timezone	0.5.28