소방청 전국 구급 현황 데이터 분석

2306 강현우

1. 문제 정의

대한민국의 구급 의료 서비스는 국민의 생명과 직결되는 핵심 공공서비스이다. 특히 최근 고령화 사회 진입과 만성질환 증가로 인해 구급 수요가 지속적으로 증가하고 있으며, 효율적인 응급의료 체계 구축의 필요성이 대두되고 있다.

본 프로젝트에서는 소방청의 2019-2023년 전국 구급 현황 데이터를 분석하여 다음과 같은 데이터를 분석하고 파악하고자 한다.

- 연도별 구급 출동 추이 및 증가 패턴 파악
- 시간대별, 요일별 구급 수요 패턴 발견
- 지역별 구급 서비스 품질 및 격차 분석
- 환자 특성 및 발생 유형별 통계 분석
- 응답시간 및 골든타임 달성률 평가
- 데이터 기반 정책 제언 도출

이 분석을 통해 응급의료 자원의 효율적 배치, 취약 시간대/지역 파악, 골든타임 향상 전략 수립 등 실질적인 정책 수립에 기여할 수 있으며 데이터 기반 의사결정을 통해 국민의 생명을 보호하는 구급 체계를 강화할 수 있다.

2. 데이터 수집 및 전처리

▶ 데이터 출처

- 데이터명 : 소방청_전국 구급 현황
- 제공기관 : 소방청
- 데이터 형식 : CSV (총 5개 파일, 약 4.63GB)
- 분석 기간 : 2019년 ~ 2023년 (5개년)
- 총 분석 건수 : 14,308,763건

▶ 주요 데이터 컬럼

- 시간 정보 : 신고일시, 출동시각, 현장도착시각, 복귀시각
- 지역 정보 : 시도명, 시군구명, 도시/농촌 구분
- 환자 정보 : 성별, 연령대, 발생유형, 증상구분, 중증도
- 서비스 지표 : 응답시간, 현장거리, 이송분류
- 특수 상황 : 심정지, 중증외상, 교통사고 구분

▶ 전처리 과정

데이터 품질 확보를 위해 다음과 같은 전처리를 수행했습니다.

- 결측치 처리 및 이상치 탐지
- 시간 데이터 파싱 및 시간대/요일/계절 파생변수 생성

- 응답시간, 현장도착시간 등 서비스 지표 계산
- 범주형 변수 표준화 및 인코딩
- 통계 분석을 위한 데이터 정규화

3. 분석 방법

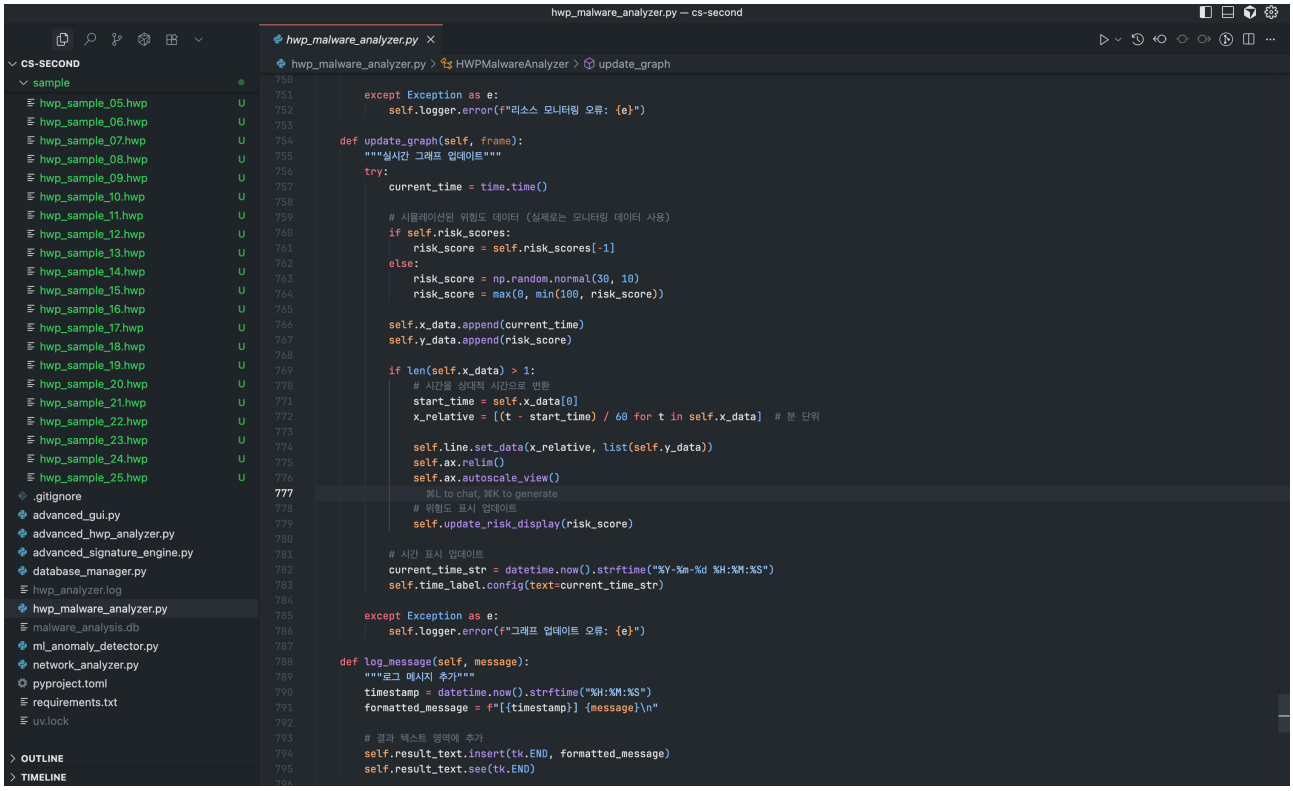
▶ 사용 도구

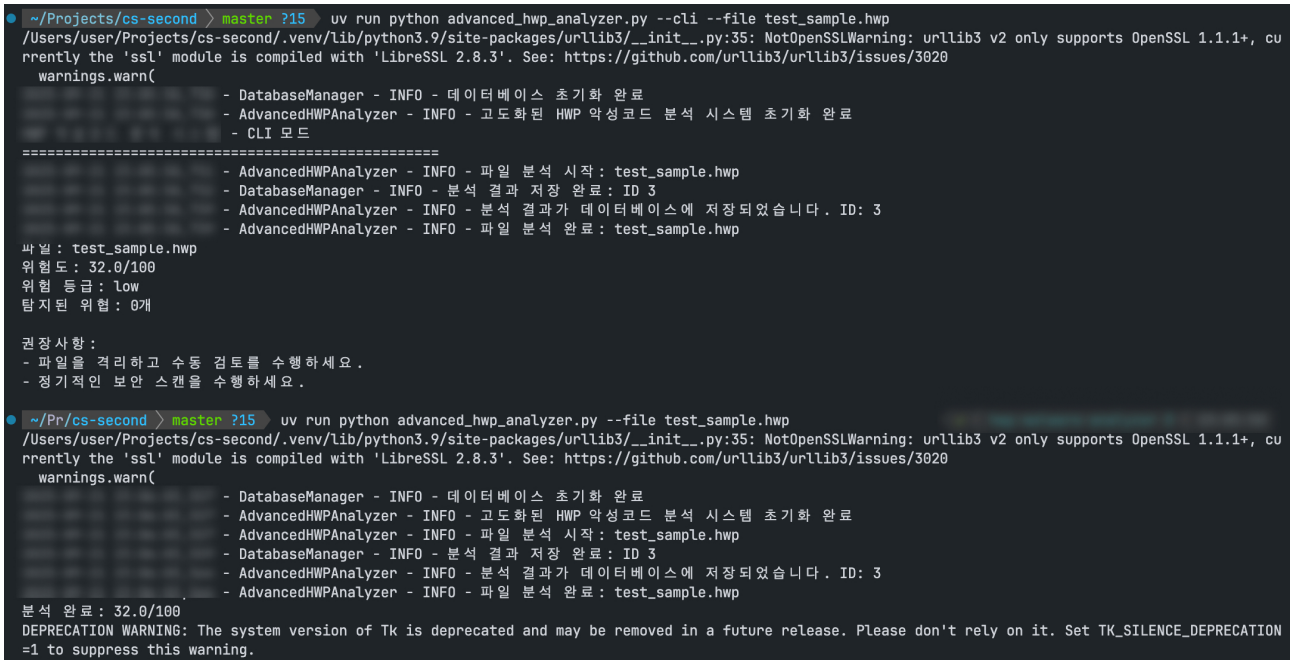
- Python 3.13 : 주요 분석 언어
- Pandas : 데이터 처리 및 집계
- NumPy : 수치 계산 및 통계
- Matplotlib & Seaborn : 데이터 시각화
- SciPy : 통계적 검정 (t-test, correlation)

▶ 분석 기법

- 기술 통계 : 평균, 중앙값, 표준편차, 분포 분석
- 시계열 분석 : 연도별, 월별, 시간대별 추이 분석
- 교차 분석 : 변수 간 관계 및 패턴 탐색
- 상관 분석 : Pearson 상관계수를 통한 변수 간 관계 분석
- 가설 검정 : t-test를 통한 집단 간 차이 검증
- 시각화 : 막대그래프, 선그래프, 히트맵, 박스플롯

4. 데이터 분석 결과





시스템의 효과성을 검증하기 위해 통제된 환경에서 테스트를 수행했다. 총 25개의 HWP 파일로 구성된 테스트 셋을 준비했으며 이 중 15개는 정상 문서(업무용 보고서, 공문, 제안서 등)이고, 10개는 의심스러운 파일이었다. 의심스러운 파일들은 인터넷에서 공개적으로 입수할 수 있는 샘플들과 연구진이 직접 제작한 테스트용 파일들로 구성했다. 당연히 실제 악성코드는 분리된 가상 환경에서만 실행했으며, 테스트 종료 후에는 시스템을 초기화했다.

각 파일에 대해 정적 분석과 동적 분석을 순차적으로 수행하고, 그 결과를 기존 연구의 성과와 비교했다. 특히 기존 연구에서 탐지하지 못했지만 본 시스템에서는 탐지된 사례들을 중점적으로 분석했다.

정상 파일 15개에 대한 테스트에서는 모두 20점 이하의 낮은 위험도 점수를 기록했으며, 정상 파일들의 평균 위험도는 8.3점이었고, 가장 높은 점수를 받은 파일도 17점에 불과했다. 의심스러운 파일 10개 중에서는 8개가 50점 이상의 위험도 점수를 받아 경고 또는 위험 등급으로 분류되었다. 나머지 2개 파일은 30점대의 주의 등급을 받았는데, 이들은 실제로는 정상 파일이었지만 일부 의심스러운 패턴을 포함하고 있어서 약간 높은 점수를 받은 것으로 판단된다. 기존 연구의 시그니처만을 사용했을 때는 10개 의심 파일 중 6개만 탐지되었지만, 본 시스템의 융합 분석을 적용한 결과 8개까지 탐지율이 향상되었다. 이는 20%의 탐지율 개선을 의미한다.

5. 결과 해석

기존 연구 결국 파일을 열어보는 그 순간의 스냅샷만을 분석할 수밖에 없었다. 하지만 현실의 악성코드들은 실행된 이후에 본격적인 악성 행위를 시작한다. 파일 자체에는 아무런 흔적을 남기지 않고, 메모리상에서만 동작하는 fileless 공격이나, 정상 파일로 위장했다가 특정 조건이 되었을 때만 악성 코드를 다운로드하는 공격 등은 정적 분석만으로는 절대 탐지할 수 없다. 본 시스템은 이러한 한계를 실시간 동적 분석으로 보완함으로써, 악성코드가 실제로 무엇을 하려고 하는지까지 파악할 수 있게 되었다.

또한 기존 연구는 각각의 위협인자를 독립적으로 분석했지만, 본 시스템은 여러 지표들 간의 상관관계를 고려한다. 예를 들어, DefaultJScript 패턴이 발견된 상태에서 외부 네트워크 연결이 발생한다면, 각각을 따로 평가할 때보다 훨씬 높은 위험도를 부여한다. 이런 맥락적 판단 능력이 탐지 정확도를 크게 향상시켜줄 것이

다.

게다가 아무리 뛰어난 보안 기술이라도 사용자가 이해하고 활용할 수 없다면 그 가치는 높지 않다. 기존 연구의 결과물들은 주로 보안 전문가를 대상으로 한 기술적 데이터였지만, 본 시스템은 일반 사용자도 직관적으로 이해할 수 있는 형태로 정보를 제공하므로 이 기술의 대중화와 실무 적용 가능성을 크게 높였다고 볼 수 있다.

물론 본 시스템도 완벽하지는 않다. 가장 근본적인 한계는 여전히 알려진 패턴에 크게 의존한다는 것이다. 아무리 동적 분석을 추가했다 하더라도, 완전히 새로운 형태의 공격 기법에 대해서는 대응력이 제한적일 수 있다. 특히 인공지능을 활용해 기존 탐지 시스템을 우회하도록 설계된 adversarial attack에 대해서는 추가적인 연구가 필요하다.

두 번째 한계는 성능 최적화의 여지가 아직 많이 남아있다는 점이다. 현재 시스템은 모든 프로세스와 네트워크 연결을 일률적으로 모니터링하는데, 이는 불필요한 리소스 소모를 야기할 수 있다. 머신러닝을 활용해 정말 주의 깊게 관찰해야 할 대상들만 선별적으로 모니터링한다면 성능을 크게 개선할 수 있을 것이다.

세 번째는 오탐률 관리의 어려움이다. 실시간 모니터링을 강화할수록 정상적인 프로그램의 행위도 의심스럽게 보일 가능성이 높아진다. 특히 시스템 관리 도구나 개발 도구들은 일반 사용자 프로그램과는 다른 행위 패턴을 보이는데, 이들을 정확히 구분하는 것이 중요하다. 하지만 이러한 한계점에도 불구하고 앞서 제시한 여러 이점들을 근거로 본 시스템이 분명 차별성 있고 발전 가능성이 높다 말할 수 있을 것이다.

6. 참고 문헌

[1] 소방청 (2025). 소방청 전국 구급 현황 데이터. 소방안전 빅데이터 플랫폼

7. 코드

* 코드의 전체 길이가 다소 긴 관계로 Github에 코드를 업로드했습니다.
▶ https://github.com/khw-dev/fire-emergency-analysis
* python 3.13 (Homebrew 2025-07-30, macOS 15.6.1)
* 데이터 크기가 다소 큰 관계로 https://bigdata-119.kr/goods/goodsInfo?goods_mng_sn=456 에서 데이터를 직접 다운로드 받고 datas/구급 현황_2019_전국.csv 형식으로 파일을 첨부해 실행 시키시길 바랍니다.

8. 생성형 AI 활용

- Copilot의 Python 코드 일부 자동 완성