**1. Data Curation**

1-1. Data Selection

- raw data 파일 column 이름 할당

- 4개의 raw data 파일 -> 공격 / 정상 DataFrame으로 분리

- 4개의 정상 DataFrame 에서 각각 25% 샘플링

- 4개의 공격 DataFrame 에서 Business Understanding의 목적에 따라서attck\_cat 에서 generic, Reconnaissance 항목만 선택

- 1\_Raw\_DataSet.csv 파일 생성

1-2. Data cleaning

- Raw\_DataSet.csv 행 개수 - 771931

- Raw\_DataSet.csv 결측값 확인 – 결측값이 있는 column 2개 확인

ct\_flw\_http\_mthd - HTTP 프로토콜이 아닌 경우

is\_ftp\_login - FTP 관련 트래픽이 아닌 경우

- 해당 결측치 0으로 대체 (HTTP 요청, FTP 로그인 없음) – unit8 형식으로 변환

- Raw\_DataSet.csv 숨은 결측치 ct\_ftp\_cmd

['0' ' ' '1' '2' '4' '6' '3' '5' '8'] – object 타입

- 해당 공백 0으로 대체하고 int 타입으로 변환 - [0 1 2 4 6 3 5 8]

- Raw\_DataSet.csv 중복 데이터 확인 – 224685개의 중복 데이터 발견 -> 제거

- 0, 1 로 이루어진 범주형 feature unit8 형식으로 변환

- Raw\_DataSet.csv -> 공격 / 정상 DataFrame으로 분리

- 정상 DataFrame Outlier 제거 (IQR)

- 공격 DataFrame에서는 Outlier가 공격 여부를 판단하는 특성이 될 수 있어 Outlier 유지

- 데이터 클리닝 시각화 결과

도표, 원, 스크린샷, 디자인이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

- 일부 Feature 제거

|  |  |  |
| --- | --- | --- |
| 유형 | 예시 | 제거 이유 |
| 고유 식별자 | srcip, dstip | 모델에 의미 없는 유니크한 문자열/주소 |
| 시각 정보 원본 | Stime, Ltime | 시간차(dur)로 대체 가능 |
| 무작위 수치 | stcpb, dtcpb | 시퀀스 번호지만 모델에 불필요한 노이즈 |
| 설명용 변수 | attack\_cat | label로 충분, 종속된 범주값 |

- 2\_Cleaning\_DataSet.csv 파일 생성

1-3. Data Preprocessing (Scaling, Encoding)

- [ 'dur', 'sbytes', 'dbytes', 'Sload', 'Dload', 'Spkts', 'Dpkts', 'smeansz', 'dmeansz', 'Sjit', 'Djit', 'Sintpkt', 'Dintpkt' ] 비대칭 분포 -> 로그변환 + Robust\_Scale

- ['tcprtt', 'synack', 'ackdat'] – Standard Scale

- ['trans\_depth', 'res\_bdy\_len', 'ct\_flw\_http\_mthd', 'ct\_ftp\_cmd'] – MinMax Scale

- 정상 데이터로 fit / 전체 데이터 transform

- Label을 제외한 범주형 데이터 One-Hot Encoding

- 3\_Preprocessing\_DataSet.csv 파일 생성

**2. Feature Engineering**

1. 1차 Feature Selection

- Logistic Regression-based feature selection (L1 regularization) 수행 (범주형 feature)

- 13개 ['proto\_ospf' 'proto\_sctp' 'proto\_udp' 'state\_INT' 'state\_REQ' 'service\_-' 'service\_dhcp' 'service\_dns' 'service\_ftp-data' 'service\_http' 'service\_smtp' 'service\_ssh' 'service\_ssl'] feature 추출

- Correlation Based Feature Selection 수행 (수치형 feature) / (Correlation > 0.1)

- 너무 많은 feature 추출 – 0.3으로 변경

스크린샷, 라인, 도표, 다채로움이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

스크린샷, 라인, 도표, 픽셀이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

- Correlation Based Feature Selection 수행 (수치형 feature) / (Correlation > 0.3)

- 16개 ['dbytes', 'sttl', 'Sload', 'Dload', 'Dpkts', 'dmeansz', 'tcprtt', 'ackdat', 'ct\_state\_ttl', 'ct\_srv\_src', 'ct\_srv\_dst', 'ct\_dst\_ltm', 'ct\_src\_ltm', 'ct\_src\_dport\_ltm', 'ct\_dst\_sport\_ltm', 'ct\_dst\_src\_ltm'] 추출

- Variance Thresholding 수행 (수치형 feature) / (Threshold> 0.01)

- 수치형 변수들의 Variance가 전부 1 이상 -> 적용불가

2. Feature Creation

- 1. 송수신 비율 (트래픽이 한쪽으로 치우쳐 있는지) - load\_ratio = Sload / Dload

- 2. TTL 차이 (패킷의 네트워크 도달 거리 차이 → 비정상 흐름 탐지) ttl\_diff = sttl - ct\_state\_ttl

3. 2차 Feature Selection

- Logistic Regression-based feature selection (L1 regularization) 수행 (범주형 feature)

- 13개 ['proto\_ospf' 'proto\_sctp' 'proto\_udp' 'state\_INT' 'state\_REQ' 'service\_-' 'service\_dhcp' 'service\_dns' 'service\_ftp-data' 'service\_http' 'service\_smtp' 'service\_ssh' 'service\_ssl'] feature 추출

- Correlation Based Feature Selection 수행 (수치형 feature) / (Correlation > 0.3)

- 17개 ['dbytes', 'sttl', 'Sload', 'Dload', 'Dpkts', 'dmeansz', 'tcprtt', 'ackdat', 'ct\_state\_ttl', 'ct\_srv\_src', 'ct\_srv\_dst', 'ct\_dst\_ltm', 'ct\_src\_ltm', 'ct\_src\_dport\_ltm', 'ct\_dst\_sport\_ltm', 'ct\_dst\_src\_ltm', 'ttl\_diff',] feature 추출

패턴, 스크린샷, 픽셀, 사각형이(가) 표시된 사진

AI 생성 콘텐츠는 정확하지 않을 수 있습니다.

- Variance Thresholding 수행 (수치형 feature) / (Threshold> 0.01)

- 적용불가

- 4\_FeatureEngineering\_DataSet.csv 파일 생성

피쳐 중요도

텍스트, 스크린샷, 그래프, 도표이(가) 표시된 사진

AI가 생성한 콘텐츠는 부정확할 수 있습니다.

**3. Modeling**

Logistic Regression을 사용하는 경우

99퍼센트의 정확도 지표를 보입니다.

Random Foreset를 사용하는 경우

100퍼센트의 정확도 지표를 보입니다.

이는stt, ttl-diff 피쳐가 라벨과 98퍼센트의 상관관계를 보이기 때문입니다.

**랜덤 포레스트**는 여러 개의 결정 트리(Decision Tree)를 조합한 **앙상블 기법**입니다.

각 트리는 데이터의 **무작위 샘플**과 **무작위 특성 선택**으로 학습되며, 최종 예측은 모든 트리의 \*\*투표(분류) 또는 평균(회귀)\*\*으로 결정됩니다.

개별 트리보다 **과적합이 적고 안정적인 성능**을 보이며, 특성 중요도도 제공하는 인기 있는 머신러닝 알고리즘입니다.

Sttl이 공격 트래픽과 높은 상관관계를 가지는 이유:

DDoS 공격에서 공격자들은 자신의 실제 IP 주소를 숨기기 위해 가짜 소스 IP 주소를 스푸핑합니다. 이때 스푸핑된 IP 주소들은 실제로는 존재하지 않거나 다양한 지역에 분산되어 있어, 각각 다른 네트워크 경로를 통해 패킷이 전송되는 것처럼 보입니다.

정상적인 통신에서는 동일한 출발지에서 동일한 목적지로 향하는 패킷들이 일관된 경로를 통해 전송되므로, TTL 값이 상대적으로 예측 가능한 범위 내에서 나타납니다. 반면 공격 트래픽에서는 스푸핑된 다양한 IP 주소로 인해 TTL 값이 광범위하고 불규칙하게 분포됩니다.