Network Security (01)

HW2 : Network intrusion detection system

2171056 강승연

본 과제는 캡쳐된 패킷을 바탕으로 악의적인 패킷을 탐지하는 것을 목적으로 한다.

실험 환경 : Google Colab Free

데이터셋 : Train Data 20416877, Test Data 3710833

산출물 : 보고서, ipynb 파일, Malicious 하다고 분류된 패킷의 csv 파일(protocol, ip src, ip dest, port src, port dest)

실험 순서

1. 전처리
2. 모델 학습
3. 텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

   자동 생성된 설명csv 추출

결과물의 일부(10행)

총 863열의 malicious 패킷이 추출되었다.

[전처리]

전처리는 모델 학습을 위해 필요한 부분만 진행했다.

1. categorize 될 수 없는 특수한 문자열로 구성되어 다루기 어려운 열은 drop

2. categorize 가능한 문자열은 전부 categorize 후 코드화 했다. 이때, categorize 한 열은 결측치를 전부 0으로 변환했다.

텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명1-12, 15, 17, 19열을 제외한 모든 열을 drop한 코드

Categorize 및 코드화를 진행한 코드

이때 train 데이터와 test 데이터의 코드를 동일하게 만들기 위해 둘을 합쳐 작업한 뒤 분리하는 과텍스트, 스크린샷, 소프트웨어이(가) 표시된 사진

자동 생성된 설명정을 거쳤다.

전체 과정에서 사용된 데이터 중 이후에 사용하지 않는 데이터는 del 로 RAM에서 삭제했다.

[모델 학습] 및 [csv 추출]

모델은 XGBoost 모델을 사용했다.

해당 모델을 선정한 이유는 다음과 같다.

1. 제공된 데이터 셋이 커, 빠른 속도로 학습할 수 있는 모델을 원했다. XGBoost는 병렬 처리를 지원해 큰 데이터셋을 빠르게 처리할 수 있는 모델이다.
2. Test 데이터에 Label을 붙여 Malicious 한 데이터를 추출하는 과제였기에 검증 과정이 따로 필요했다. XGBoost 는 Cross validation 기능을 내장하고 있어 자체적으로 검증을 수행한다.
3. 데이터를 쭉 살펴봤을 때 결측값이 많았다. XGBoost는 결측값을 내부에서 자체적으로 처리해준다.

모델을 학습한 코드이다. label이 주어진 test data 가 주어지지 않아 특별히 하이퍼 파라미터 튜닝은 진행하지 않았다.

전처리를 진행한 데이터가 아닌, 원본 데이터를 추출하고자 where로 인덱스를 찾아 malicious 한 텍스트, 스크린샷, 폰트, 소프트웨어이(가) 표시된 사진

자동 생성된 설명데이터를 추출했다.

마찬가지로 label이 주어진 test data가 주어지지 않아 정확도 측정 등 검증은 진행하지 않았다. 바텍스트, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명로 추출한 데이터를 csv 파일로 변환하여 저장했다.