머신러닝 기반 악성 URL 판별 기법 개발

팀 명 : 푹신푹신

지 도 교 수 : 양환석 교수님

팀 장 : 강수진

팀 원 : 강민성

문동준

오현진

주현우

2023. 11.

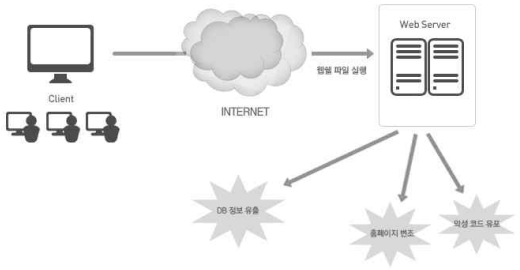
중부대학교 정보보호학과

목 차

1. 서 론

1.1 연구배경

최근 들어 악성 URL을 통한 범죄가 급증하고 있다. 악성 페이지를 기존 페이지와 흡사하게 만들어 정보를 입력하면 빼가는 수법, 접속하면 걸리는 랜섬웨어 등 여러 가지 방면으로 범죄가 급증하고 있다. 이를 방지, 대응하고자 악성 URL 판별 프로그램을 만들도록 한다.



[그림 1] 악성URL

1.2 연구 필요성

기존의 악성 URL과 다른 새로운 유형의 악성 URL들이 발견되고 있으며 이에 대응하기 위한 연구가 빠르게 진행되고 있다. 그 중에서도 머신러닝과 알고리즘을 악성 URL 판단 과정에 접목시켜 특징적인 차이점을 학습하는 연구가 활발히 진행되고 있으며 각 모델을 조합하는 방식에 대한 연구도 활발하다. 그러나 기존의 특징 기반 머신러닝 예측 방식은 구(舊)유형의 악성 URL의 높은 가변성에 효과적인 특징값 추출에 있어 어려움이 발생하고 있다.

1.3 연구 목적 및 주제선정

본 연구는 악성 URL에 의한 피해를 방지하기 위한 방법을 고민해보았고, 과거부터 최신까지 악성 URL들이 공통적으로 갖는 유의미한 특징을 조사하여 특징값을 선정하고 실시간 데이터셋 최신화 과정을 통해 기존보다 보완된 머신러닝 기반 악성 URL 예측 시스템 구현에 집중하여 진행되었다.

2. 관련연구

2.1 Python

Python은 이해하기 쉬운 구문, 다양한 라이브러리, 높은 확장성을 가진 인터프리터 언어이다. 컴파일러 단계가 없어 과정이 단순해 생산 속도가 빠르지만 한 줄 단위로 실행되기 때문에 실행 속도가 느리다. 파이썬은 웹 개발 뿐만 아니라 데이터 분석, 머신러닝 등 여러 분야에서 활용되고 있다. 무료로 다운로드 가능하며 다양한 운영 체제에서 호환이 가능하다.

2.2 Flask

Flask는 Python기반의 웹 프레임워크다. 특별한 도구나 라이브러리가 필요 없다. 폼(form), 데이터베이스(database)를 처리하는 기능이 없지만 Flask는 확장 모듈을 사용해 이 점을 보완한다. Flask 서버를 만들 때, 프로젝트 폴더 안에 static 폴더(css, images, javascript 파일을 넣어두는 폴더), templates 폴더(html 파일을 넣어두는 폴더), app.py(프로그램을 실행시키는 파일)을 항상 만들어두어야 한다.

2.3 DGA(Domain Generation Algorythms)

DGA는 특정 URL이 악의적인 목적하에 동적으로 생성되는 알고리즘 클래스다. 일반적으로 도메인은 악성 행위자의 명령 및 제어 서버에 대한 콜백을 용이하게하기 위해 멀웨이 및 봇넷에서 랑을 생성 할 수 있으며 대부분은 등록되지 않은 도메인이다. 엄청난 수의 등록되지 않은 도메인이 등록 된 도메인을 가장하는데 사용되어 감염된 봇넷이 서명 또는 IP 평판 기반 보안 탐지 시스템에 의한 탐지 및 억제를 피할 수 있다. DGA 활동은 일반적으로 XNUMX 가지 일반적인 단계로 네트워크 패킷을 캡쳐하고 분석하여 감지된다.

2.4 CNN(Convolutional Neural Network)

CNN은 합성곱 신경망으로 데이터의 특징(feature)을 추출하여 그 특징들이 가진 패턴을 파악하는 구조다. 이미지의 공간 정보를 유지한채 학습하기 때문에 주로 이미지나 영상 데이터를 처리할 때 사용된다. CNN은 전처리 작업이 들어가는 뉴런 네트워크 모델이다. CNN 과정은 Convolution 과정과 Pooling 과정을 통해 이루어지며, Layer 역시 Convolution Layer와 Pooling Layer를 복합적으로 구성되어 알고리즘을 만든다.

2.4 RF(Random Forest)

Random forest는 여러 결정 트리를 묶어 하나로 만든 것으로, 각기 다른 방향으로 과대적합된 트리들을 묶음으로 평균을 냄으로써 예측 정확성은 향상 시키고 과적합은 줄일 수 있다. 이러한 방식을 구현하기 위해서는 개별 결정 트리들을 많이 만들어야하며 각각의 결정 트리는 타킷 예측을 잘해야 하고 다른 트리와는 구별되는 특성을 지녀야 한다. 다양성 확보를 위해 Bootstrap aggregating(bagging) 방식을 사용하고, 무작위성 확보를 위해서는 random subspace 방식을 사용하여 다양성과 무작위성을 확보하는 것이 Random Forest의 목표이다.

2.4 GB(Gradient Boosting)

Gradient Boosting은 회귀 및 분류에 사용되는 기계 학습 기술로 각 후속 트리가 이전 트리에서 발생한 오류를 수정하려고 시도하는 일련의 의사 결정 트리를 반복적으로 구축하여 작동한다. 손실 함수를 최소화하기 위해 Gradient descent를 사용한다. 각 반복에서 예측 값에 대한 손실 함수의 기울기를 계산하고 기울기 방향으로 예측을 업데이트한다.

3. 본 론

3.1 특징값 추출

기존의 머신러닝을 활용한 악성 URL탐지 모델에서는 대부분 개수 기반, 길이 기반, 존재 기반, 비율 기반등의 어휘적 특징이 많이 사용된다. 그러나 프로젝트 진행 과정에서 어휘적 특징만을 활용한 모델 구축으로는 신규 악성 URL에 대한 악성 유무 판단에서 한계가 있음을 확인하여 도메인 수명, 트래픽 길이, 도메인 생성 일자와 같은 도메인 정보와 관련된 특징값들을 추가로 추출하여 모델 학습 과정에 활용했다. 다음과 같은 특징값들을 포함하여 학습 과정에서 모델내 특징값별 기여도를 참고하여 시중에 있는 논문을 참고하여 35개의 특징값을 최종 선별했으며 그 종류는 다음과 같다.

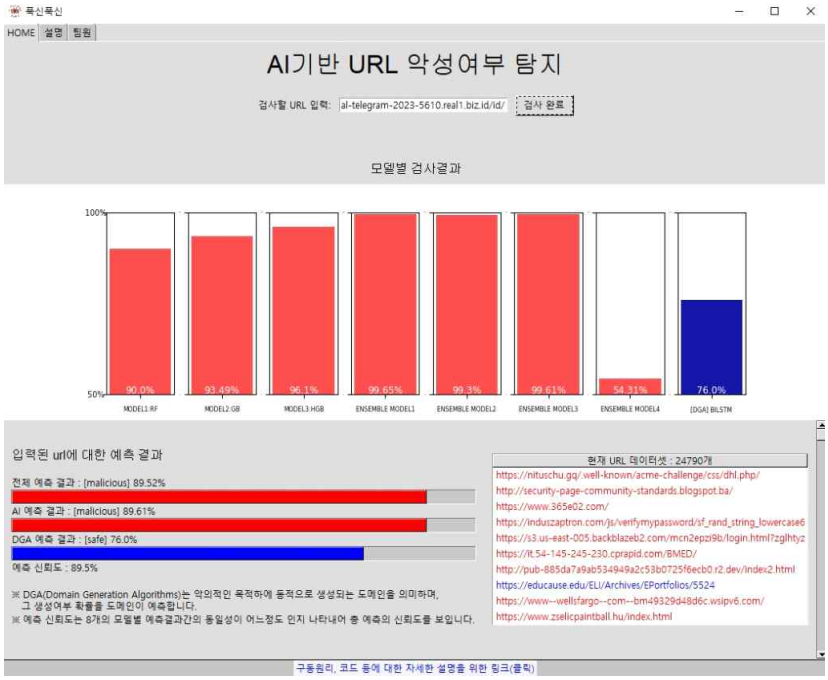
CATEGORY	FEATURES	NUMBER
개수 기반	도메인 개수, http개수, https개수, .개수, //개수, -개수, @개수, www개수, =개수, _개수, ~개수, ?개수, &#%개수, 악성문자열 개수, 숫자 개수, 쿼리 개수, 쿼리 악성 문자열 개수	17개
길이 기반	url 길이, url path 길이, url netloc 길이, url tld 길이, 쿼리 길이	5개
존재 기반	쿼리 인코딩 유무, ip포함유무, 단축서비스 유무	3개
비율 기반	랜덤한 정도, 대문자 알파벳 비율	2개
도메인 기반	포트번호, 도메인 생성일~현재, 현재~도메인 만료일, 도메인 전체수명, 트래픽길이, abnormal유무	6개

[표 1] 특징값들

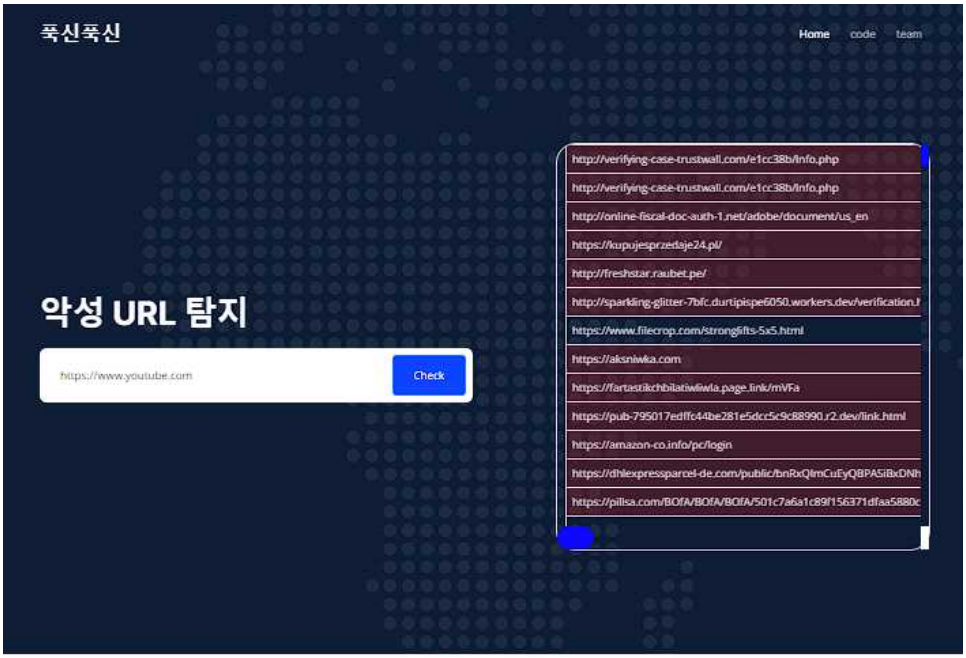
특징값의 추출 과정 역시 파이썬에서 이루어졌으며 도메인 정보와 같은 정보는 urllib 라이브러리를 활용하여 추출했으며 출력된 특징값을 데이터화 하여 저장하기 위한 csv 라이브러리, 학습이 가능한 형태로 저장하고 활용하기 위한 pandas 라이브러리도 활용되었으며 모델 학습을 위한 sklearn 라이브러리도 활용했다. 모델은 총 10개의 단일 모델을 활용했으며 각 모델은 joblib 라이브러리를 활용해 .h5확장자 파일로 저장하여 활용했다. 학습에 활용되지 않은 데이터를 활용한 성능 테스트 결과 RF(Random Forest), GB(Gradient Boosting), HGB(Hist Gradient Boosting)의 3가지 모델이 단일 모델로는 가장 높은 성능을 보였으며 조합 모델은 대체적으로 높은 성능을 보였으나 스태킹 방식으로는 RF-GB-ET-MLP-LR, DT-RF-KN-MLP-LR, RF-GB-MLP-AB-HGB가 보팅 방식으로는 RF-GB-HGB가 가장 높은 성능을 보여 다음 7개의 모델을 최종 선택하게 되었다.

3.2 프로그램 구성

3.2.1 GUI



3.2.2 Web



3.2.2.1 DGA

DGA를 탐지하는 CNN 모델을 구현하였으며 모델과 전처리, 임베딩 과정은 "남궁주홍. DGA 도메인 탐지를 위한 효율적인 딥러닝 모델." 국내석사학위논문 강원대학교 대학원, 2020. 강원도 논문을 기준으로 구성하였다. 논문에서는 패딩 사이즈를 73, 출력 차원을 32로 설정하였고 오버 피팅을 막기 위해 dropout(0.5)과 regularization을 적용하였는데 사용한 데이터셋에서 가장 긴 URL의 데이터 길이가 64이기 때문에 패딩 사이즈를 64로 제로 패딩하였고 차원은 똑같이 32, dropout은 0.5, L2 regularization 0.15로 진행하였다.
데이터 셋은 https://www.kaggle.com/datasets/gtkcyber/dga-dataset 에서 확인 가능하다.

CNN 모델의 구성은 커널의 입력 길이를 2-5까지 각각 다르게 하는 4개의 1D Convolutional layer를 하나의 Concatenate layer로 합친 후 3개의 Hidden layer를 거치고 출력된다. Hidden 레이어의 구성은 Fully-connected 후 activation function인 ELU를 사용

하며 Batch Normalization(배치 정규화)후 0.5의 가중치를 준 dropout이다. 논문에서 그린 모델의 아키텍쳐는 다음과 같다.
해당 모델은 다중 분류 모델로 dga와 non-dga로 나뉘며 dga의 종류에는 cryptolocker, newgoz, gameoverdga, nivdort, necurs, goz, bamital이 있고 non-dga는 alexa, legit로 나뉘어 진다.

DGA	Description
CryptoLocker	DGA를 사용하여 C&C 서버에 연결하고, 암호화된 파일을 복원하기 위해 피해자에게 요구하는 도메인을 동적으로 생성
NewGoz	금융 부문을 타켓으로 하는 악성 코드로, DGA를 사용하여 악성 서버와 통신하고 금전적 이익을 추출하는 데 활용
GameOverDGA	DGA를 사용하여 봇넷과 통신하며, 금융 정보를 탕취하고 악성 활동을 숨기기 위해 동적 도메인을 생성
Nivdort	DGA를 활용하여 트로이 목마를 배포하고, 사용자의 개인 정보를 탕취하거나 다른 악성 코드를 설치
Necurs	대규모 스팸 및 악성 파일 배포를 위해 DGA를 사용하며, 다양한 악성 활용에 이용
Goz	금융 정보를 탈취하고 악성 활동을 숨기기 위해 DGA를 활용하는 악성 코드
Bamital	클릭 사기 및 광고 클릭 부정행위를 실행하며 DGA를 사용하여 도메인을 동적으로 생성하여 악성 활동을 수행

[표 2] DGA 종류

4. 분 석

4.1 활용 결과 및 성능

본 프로젝트에서는 파이썬(Python) 환경에서 여러 머신러닝 모델을 구축하고 각 모델 내 테스트 데이터를 활용한 성능 테스트 외에도 몇몇 모델에 대해서는 학습에 활용되지 않은 신규 url을 기준으로 성능 테스트를 진행하여 실효성에 대한 검증도 함께 진행하였다.

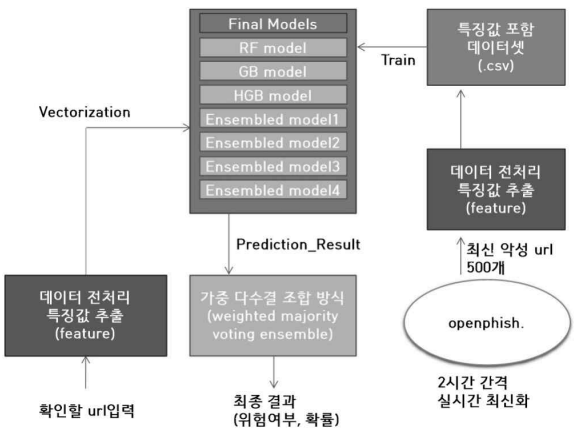
훈련 과정에서 단일 모델로는 다수의 DT(Decision Tree) 모델로부터 분류 또는 평균 예측치를 출력하여 결과를 출력하는 RF(Random Forest) 모델이 93.86%의 탐지 성능을 보였으며 신규 url에 대한 예측률도 악성 기준 90%, 정상 87.5%로 가장 준수한 성능을 보였다. 그 외 GB model, HGB model이 각각 90.74%, 90.39%로 좋은 성능을 보였다. 라벨링된 데이터 분류에 특화된 DT(Decision Tree)모델도 90.03%로 준수한 성능을 보였으나 신규 데이터 기준으로는 상대적으로 낮은 예측률을 보였다. 다중 모델 방식은 STACK방식과 VOTE방식을 활용했으며 VOTE방식은 상대적으로 예측 방식이 간단한 편인 Hard Voting 방

식으로 결과물을 출력했다.

다중 모델은 대체적으로 단일 모델에 비해 높은 구별 성능을 보였다. 특히 DT의 해석력과 RF의 다수 결정 트리를 결합하여 조합된 다중 모델들의 성능이 크게 높게 나왔다.

Algorithm	Test Set	New Data Set	
	Accuracy	Benign Accuracy	Malicious Accuracy
DT	90.93%	75%	80%
KN	84.51%	77.5%	67.5%
GNB	50.62%	17.5%	95%
MLP	71.36%	52.5%	85%
RF	93.86%	90%	87.5%
LR	81.23%	55%	67.5%
AB	88.07%	90%	72.5%
GB	90.74%	87.5%	80%
ET	80.72%	92.5%	70%
HGB	90.39%	87.5%	77.5%
[S]RF-GB-ET-MLP-LR	94.18%	90%	85%
[S]DT-RF-KN-MLP-LR	93.95%	90%	87.5%
[S]RF-GB-MLP-AB-HGB	94.53%	90%	85%
[S]KN-GNB-MLP-RF-GB	93.41%	90%	87.5%
[V]DT-KN-GNB	96.88%		
[V]DT-MLP-RF	99.11%		
[V]KN-GNB-MLP	93.98%		
[V]DT-GNB-RF	99.11%		
[V]KN-MLP-GNB	92.08%		

[표 3]모델별 결과값



[그림 2] 모델 동작과정

이후 성능이 높은 7개의 모델을 선정하여 모델의 성능 별 가중치를 부여하는 가중 다수결 조합방식(Weightes Majority Ensemble)을 활용해 url의 악성유무를 판단하는 최종 결과를 출력했다.

4.2 추후 보완사항

사용자에게 받은 URL이 저장된 DB를 기존의 알고리즘에 새롭게 보내는 작업

5. 결 론

5.1 결 론

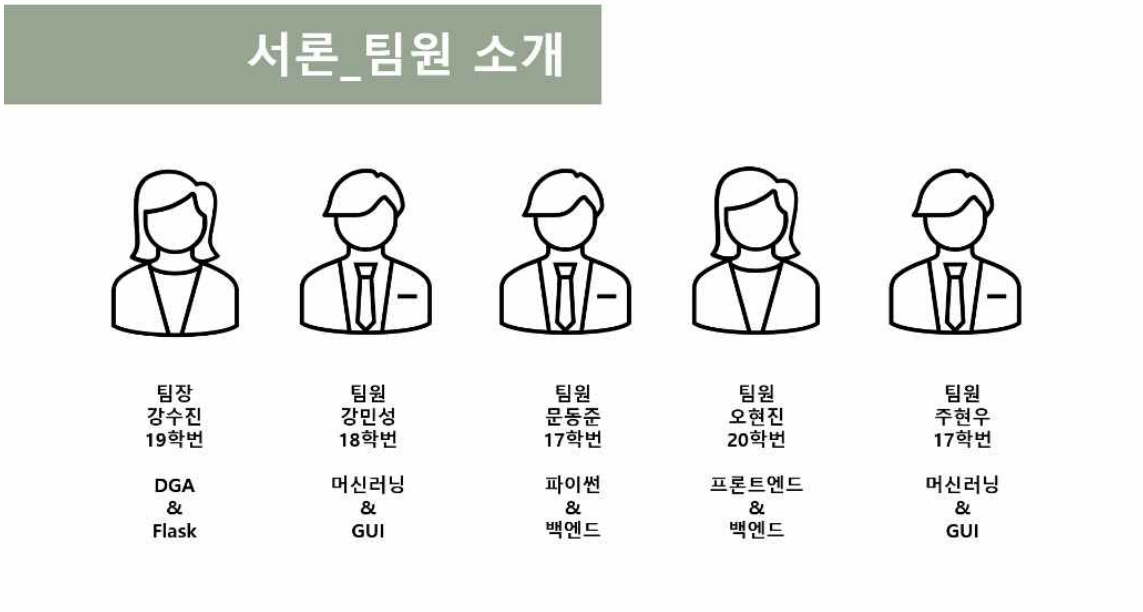
최근 악성 URL 을 활용한 사이버 위협이 지속되고 있으며 새로운 패턴에 대해 예측하여 대응하기 위한 정보 보안 시스템의 중요성이 강조되고 있다. 본 프로젝트는 파이썬(Python) 환경에서 실시간 데이터 최신화, 유의미한 특징값들을 간소화된 과정으로 신속하게 출력하고 이를 다수의 머신러닝 알고리즘 혹은 다중 머신러닝 알고리즘에 학습시킨 후 고성능의 모델을 활용해 실제 웹 기반 악성 URL판별 서비스를 제공함으로써 그 효용성에 대해 연구할 수 있었다.

5.2 기대효과

향후 연구에서는 특징값 별 유효도 검사를 통해 기존의 특징 추출 과정을 보완 및 학습에 용이한 새로운 특징값을 확보하고 다양한 머신러닝 알고리즘의 강점을 연구하면서 딥러닝 알고리즘까지 접목시킴으로써 예측 성능을 점차 고도화시킬 예정이다.

6. 별 첨

6.1 팀원 소개

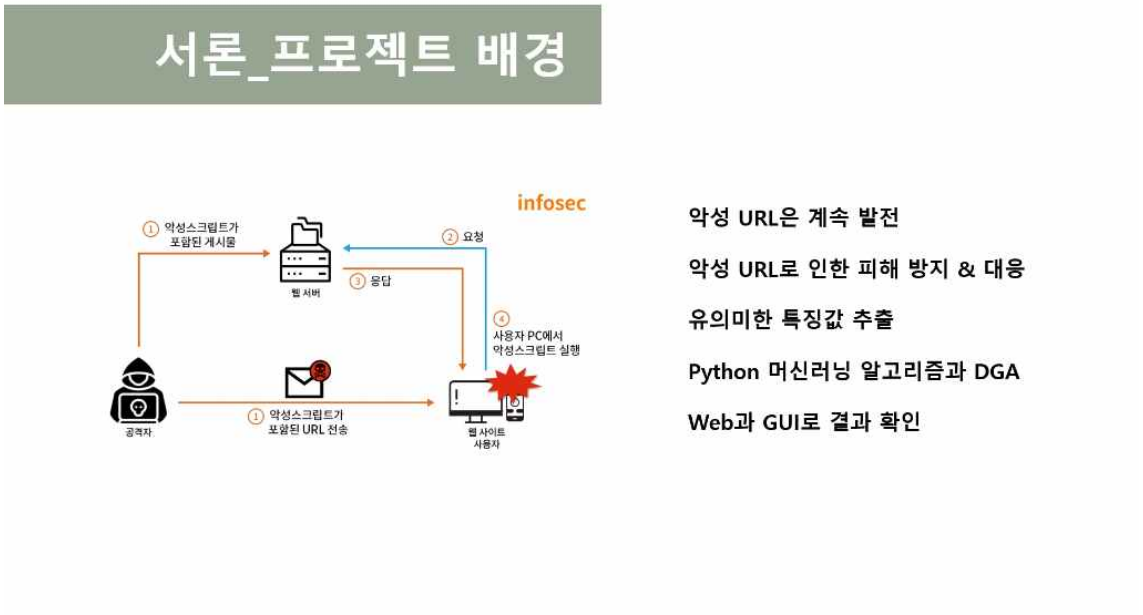


6.2 소스 코드

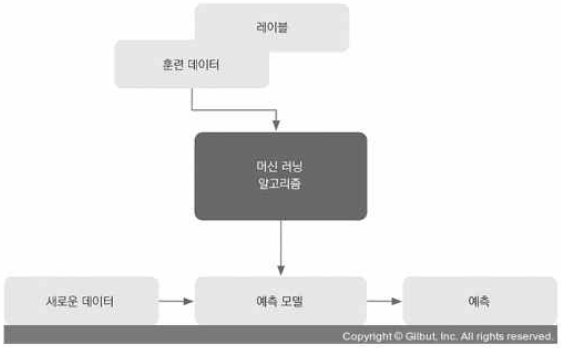
해당 단계에서는 url에서 선정된 특징값들을 추출하고 학습하여 모델을 생성하고 해당 모델과 url을 입력값으로 하여 url의 악성여부를 예측하는 함수를 제작했다.

깃허브 주소 : https://github.com/ansehd1123/aiurl

6.3 발표 자료



서론_계획안



파이썬 기반
Flask로 연결
API 사용

악성 URL feature 추출
->
추출 값을 알고리즘에 학습

본론_특징값 추출

CATEGORY	FEATURES	NUMBER
개수 기반	도메인 개수, http개수, https개수, .개수, //개수, -개수, @개수, www개수, =개수, _개수, ~개수, ?개수, &#%개수, 악성문자열 개수, 숫자 개수, 쿼리 개수, 쿼리 악성 문자열 개수	17개
길이 기반	url 길이, url path 길이, url netloc 길이, url tld 길이, 쿼리 길이	5개
존재 기반	쿼리 인코딩 유무, ip포함유무, 단축서비스 유무	3개
비율 기반	랜덤한 정도, 대문자 알파벳 비율	2개
도메인 기반	포트번호, 도메인 생성일~현재, 현재~도메인 만료일, 도메인 전체수명, 트래픽길이, abnormal유무	6개

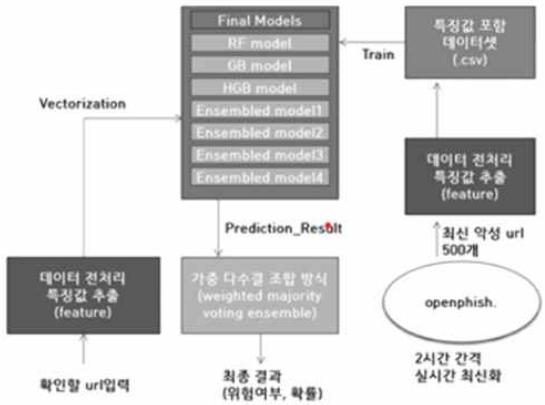
기존의 머신러닝을 활용한 악성 URL탐지 모델:

개수 기반, 길이 기반, 존재 기반, 비율 기반 등의 어휘적 특징

추가한 URL탐지 모델:

도메인 기반
도메인 수명, 트래픽 길이, 도메인 생성 일자 등

본론_머신러닝



urllib 라이브러리: 도메인 정보 출력

csv 라이브러리: 출력된 특징값을 데이터화 하여 저장

pandas 라이브러리: 학습 가능한 형태로 저장

joblib 라이브러리: .h5 확장자 파일로 저장

본론_머신러닝

CNN을 활용해 총 8개의 모델을 기준으로 악성 여부 판단

특정 URL이 악의적인 목적하에 동적으로 생성되는 도메인인지 확인 과정 필요

최신 악성 URL을 전처리를 통해 모델에 학습

검사가 필요한 URL 입력

학습된 모델로 결과 확인

본론_DGA

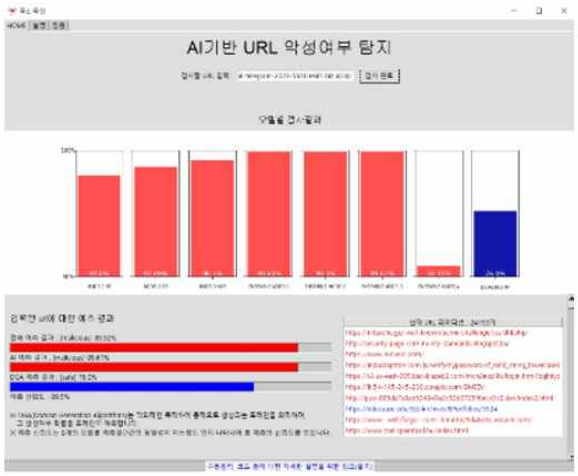
DGA	Description
CryptoLocker	DGA를 사용하여 C&C 서버에 연결하고, 암호화된 파일을 복원하기 위해 피해자에게 요구하는 도메인을 동적으로 생성
NewGoz	금융 부문을 타겟으로 하는 악성 코드로, DGA를 사용하여 악성 서버와 통신하고 금전적 이익을 추출하는 데 활용
GameOverDGA	DGA를 사용하여 봇넷과 통신하며, 금융 정보를 탈취하고 악성 활동을 숨기기 위해 동적 도메인을 생성
Nivdort	DGA를 활용하여 트로이 목마를 배포하고, 사용자의 개인 정보를 탈취하거나 다른 악성 코드를 설치
Necurs	대규모 스팸 및 악성 파일 배포를 위해 DGA를 사용하며, 다양한 악성 활동에 이용
Goz	금융 정보를 탈취하고 악성 활동을 숨기기 위해 DGA를 활용하는 악성 코드
Bamital	클릭 사기 및 광고 클릭 부정행위를 실행하며 DGA를 사용하여 도메인을 동적으로 생성하여 악성 활동을 수행

"남궁주홍, DGA 도메인 탐지를 위한 호율적인 딥러닝 모델"
국내석사학위논문 강원대학교 대학원, 2020, 강원도 논문

DGA :
Cryptolocker, newdoz, gameoverdga, Nivdort, necurs, goz, bamital

Non-DGA :
alexa, legit

결론_GUI

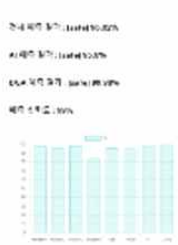


결과를 사용자의 가시성을 위해 GUI와 Web을 활용

URL의 입력
->
해당 URL을 모델에 입력
->
결과물 반환

검사 결과
AI 검사 결과
DGA 검사 결과
모델별 측정 결과
결과에 대한 신뢰도 항목

결론_Web



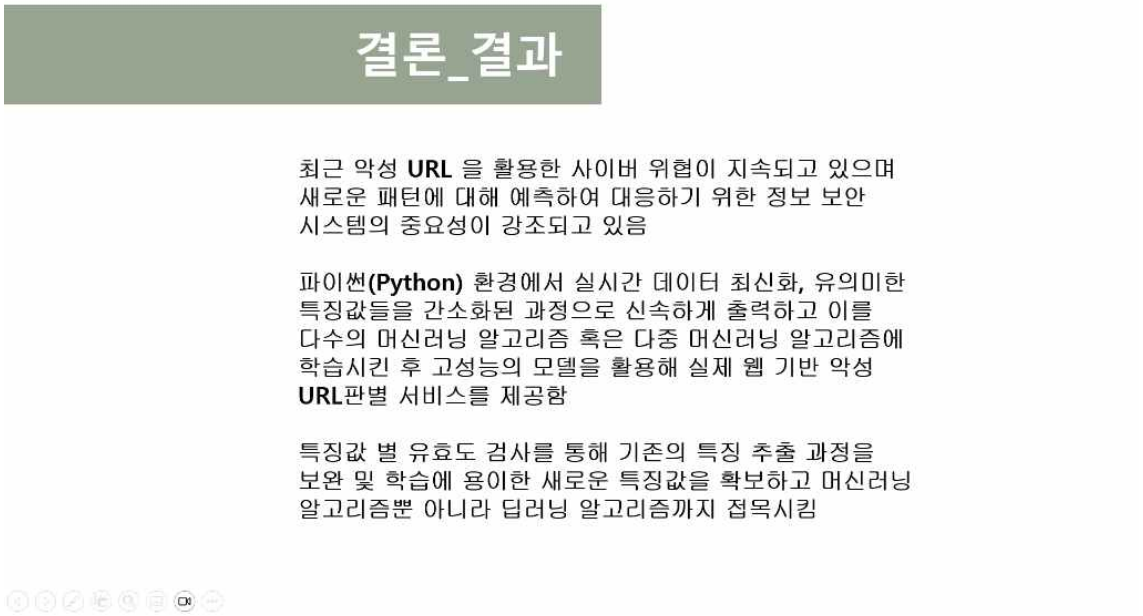
결론_성능

RF : 93.86%
GB : 90.74%
HGB : 90.39%
DT : 90.03%

스태킹 방식
RF-GB-ET-MLP-LR
DT-RF-KN-MLP-LR
RF-GB-MLP-AB-HGB

보팅 방식
RF-GB-HGB

Algorithm	Test Set	New Data Set	
	Accuracy	Benign Accuracy	Malicious Accuracy
DT	90.93%	75%	80%
KN	84.51%	77.5%	67.5%
GNB	50.62%	17.5%	95%
MLP	71.36%	52.5%	85%
RF	93.86%	90%	87.5%
LR	91.23%	55%	67.5%
AB	89.07%	90%	72.5%
GB	90.74%	87.5%	80%
ET	90.72%	92.5%	70%
HGB	90.39%	87.5%	77.5%
[S]RF-GB-ET-MLP-LR	94.18%	90%	85%
[S]DT-RF-KN-MLP-LR	93.95%	90%	87.5%
[S]RF-GB-MLP-AB-HGB	94.53%	90%	85%
[S]KN-GNB-MLP-RF-GB	93.41%	90%	87.5%
[V]DT-KN-GNB	96.88%		
[V]DT-MLP-RF	99.11%		
[V]KN-GNB-MLP	93.98%		
[V]DT-GNB-RF	99.11%		
[V]KN-MLP-GNB	92.08%		



6.4 소개 자료

[1] Rami Mustafa A Mohammad (University of Huddersfield, rami.mohammad '@' hud.ac.uk, rami.mustafa.a '@' gmail.com)

LeeMcCluskey(Universityof Huddersfield,
t.l.mccluskey '@' hud.ac.uk)

FadiThabtah(Canadian University of Dubai, fadi '@' cud.ac.ae)

[2] Malicious Code Hidden Site Detection Trend Report in the First Half of 2020 (2020),
https://www.krcert.or.kr/data/reportView.
do?bulletin_writing_sequence=35537 (accessed July 29, 2020).

[3] Development of a Malicious URL Machine Learning Detection Model Reflecting the Main Feature of URLs
한국 정보 통신 학회 논문지 = Journal of the Korea Institute of Information and Communication Engineering v.26 no.12

[4] Design and Implementation of Malicious URL Prediction Systembased on Multiple Machine Learning Algorithms
Hong Koo Kang, Sam Shin Shin, Dae Yeob Kim, Soon Tai Park