[기계학습 및 데이터 마이닝]

Term Project Final Report

- 그거 맞아? (피싱 사이트 탐지) -

팀 명: 트릴리언

안윤회 201620899

김두원 201620630

유상정 201620641

김상우 201620631

이찬호 201620648

1. Introduction

1.1. Background

최근 피싱 사이트에 대한 피해사례가 굉장히 늘고 있다. 실제 과학기술정보통신부에서 실시한 조사결과에 따르면 2016년부터 2019년 8월까지의 4년간 피싱 사이트 신고·차단 건수가 31,000건이 넘어선 것으로 확인됐다.[1] 또한 작년 하반기에는 국내 포털사이트 점유율 2위인 네이버의 로그인 창을 사칭하여 조작된 피싱 사이트로 인하여 개인정보가 유출되는 등 많은 피해사례가 나타났다. 2017년 기준 피싱, 파밍 피해액은 이전 3년간 5405억원으로 절대 무시할 수 없는 수준의 피해가 발생하고 있음을 확인할 수 있다.[2] 이에 무고한 피해자들을 사전에 예방하기 위한 웹 사이트 감지기의 역할이 더욱 중요해지고 있다.

1.2. Project Summary

본 프로젝트의 주제는 '기계학습을 이용한 피싱 사이트의 탐지' 로 다음과 같은 세 가지의 목표를 가지고 프로젝트를 진행하였다.

1) 피싱 사이트의 특징파악 및 Data Feature 선정
2) 알고리즘별 평가를 통한 가장 적합한 알고리즘 판단
3) URL 기반의 피싱 사이트와 정상사이트 구별

위 목표를 이루기 위해 우선 수집한 URL 데이터 셋으로부터 피싱 사이트의 특징을 파악하고 사이트 구분을 위한 Feature 선정을 우선적으로 진행하였다. 이후 선정된 Feature를 분류 알고리즘인 KNN, SVM, RF와 회귀 알고리즘인 Logistic Regression에 적용하고 Accuracy와 ROC 커브 분석을 거쳐 가장 성능이 좋은 알고리즘을 선택하였다. 최종적으로 알고리즘이 적용된 피싱 사이트 분류기에 임의 데이터를 넣어 유의미한 성능을 가지는지에 대해 평가하였다.

---

[1] 김평화,「개인정보 노리는 피싱사이트, 올해는 1만 건 넘긴다」, IT조선, (2019.10.15), <http://it.chosun.com/site/data/html_dir/2019/10/15/2019101502838.html>

[2] 유진상,「피싱·파밍 피해액, 최근 3년간 5405억원 달해」, IT조선, (2017.10.17), <http://it.chosun.com/site/data/html_dir/2017/10/17/2017101785025.html>

2. Data Understanding

2.1. Data set

본 프로젝트에서 사용한 데이터는 Kaggle의 phishing site와 정상 site의 URL 데이터를 사용하였다. 또한 테스트를 위한 데이터로 한국 인터넷 진흥원의 국내 URL 데이터를 추가로 이용하였다. 이 데이터 셋은 Feature가 존재하지 않고, label만 존재하며, 정상이면 1, phishing이면 0으로 이루어져 있다.

2.2. Phishing site 특징

Phishing site의 특징은 2015년 UCI 내에 등재된 Rami M. Mohammad 등 3명의 연구 자료를 참고하여 16개의 특징을, 추가적으로 6개의 영향이 생길만한 특징을 고려하여 선정하였다.

특징	설명
Double_slash_ redirecting	URL 내에서 맨 앞의 'http://'을 제외하고 한 번 더 '//'가 존재하면 다른 사이트로 리다이렉션 된다는 의미로 볼 수 있다.
having_At_Symbol	URL에 '@'가 사용되면 기호 앞이 무시되고 종종 '@' 뒤가 실제 주소가 된다.
Prefix_suffix	정상 URL에서는 '-'가 거의 사용되지 않는다. 보통 피싱 사이트에서 정상 사이트로 위장하기 위해 '-' 기호를 사용한다.
having_Sub_Domain	대부분 피싱 사이트는 2개 이상의 서브 도메인을 갖는다.
Alexa_Ranking	Alexa 데이터 베이스에는 방문자 수와 그들이 방문한 페이지 수를 기반으로 웹 사이트의 순위가 존재한다. 피싱 사이트는 비교적 짧은 기간 존재하기 때문에 데이터 베이스에 조회되지 않는다면 피싱 사이트로 의심할 수 있다.
duplicated_Tag	정상 사이트의 소스코드 내에는 <head> 태그와 <body> 태그가 보통 한 번만 사용된다.
Disabling_Right_Click	피셔는 피싱 사이트의 소스코드를 숨기기 위해서 우 클릭을 막을 수 있다.
Count_Redirection	사이트 접속 시 리다이렉션 횟수가 많다면 피싱으로 의심할 수 있다.
getAnchorResult	<a> 태그의 대부분이 웹 페이지에 연결되어 있지 않거나 웹 사이트와 다른 도메인 이름을 가진다면 피싱 사이트일 확률이 높다.
getLinksInTags	HTML 문서의 특성을 담고 있는 <meta> 태그나 자바 스크립트를 실행하는 <script> 태그, 외부 문서를 연결하는 <link> 태그가 많다면 외부와 연결된 것이 많다고 예상되므로 피싱 사이트일 확률이 높다.
Prompt_in_Popup	대부분의 정상 사이트에는 사용자에게 입력을 요구하는 팝업창이 존재하지 않는다. 따라서 정보 입력 창이 존재하는 팝업이 있다면 피싱 사이트일 확률이 높다.
DomainName_ in_Source	정상 사이트의 경우 소스코드 내에 사이트의 도메인 이름이 포함된 경우가 많다. 따라서 타 사이트의 도메인이 소스코드 내에 포함되어 있는 경우 피싱 사이트일 확률이 높다.
Domain_registration_ length	피싱 사이트의 경우 단기간만 활동하는 경우가 많기 때문에, 도메인 가입 기간이 짧은 경우 피싱 사이트로 의심할 수 있다.
Shortening_Service	URL의 길이를 상당히 줄여도 리다이렉션을 통해서 원하는 페이지로 가게 하는 방법이다. 피싱 사이트의 경우 이 방법을 통해 URL을 숨길 가능성이 높다.

favicon	웹 사이트를 대표하는 아이콘인 favicon이 존재하지 않거나 외부 도메인에서 로드되었다면 피싱 사이트일 확률이 높다.
URL_Length	피셔는 의심스러운 부분을 숨기기 위해 URL의 길이를 길게 할 수 있다. 따라서 지나치게 긴 URL인 경우 피싱 사이트로 의심할 수 있다.

표 1 참고 피싱 사이트 특징

특징	설명
Length_of_Source	HTML 소스코드의 길이가 지나치게 짧은 경우 피싱 사이트일 확률이 높다.
form_action_Handler	<form action>을 통해 사용자가 입력한 정보를 서버에서 처리해줘야 한다. 따라서 action 속성에 공백이나 'about:blank'가 존재한다면 정보에 대한 처리가 없다는 의미이므로 피싱 사이트일 확률이 높다.
onMouseOver	피셔는 사용자에게 상태 표시줄에 가짜 URL을 보여주기 위해 특히 'on mouse over' 기능을 사용할 수 있다.
abnormalUrl	WHOIS 데이터 베이스에 도메인이 존재하지 않다면 피싱 사이트일 확률이 높다.
External_Load_Script	피싱 사이트의 경우 정상 사이트의 스크립트를 그대로 가져오는 경우가 있다. 따라서 스크립트가 100% 외부에서 로드 되었다면 피싱 사이트일 확률이 높다.
Remain_Expiration	피싱 사이트의 경우 대부분 도메인 유효기간을 갱신하지 않기 때문에 남은 기간이 짧다.

표 2 추가로 고려된 피싱 사이트 특징

3. Data Preparation (pre-processing)

3.1. Data cleaning

데이터 셋에 존재하는 정상 URL 중 사이트에 접속이 되지 않는 경우 접속이 필요한 모든 Feature가 0으로 분류되기 때문에 결과에 영향을 줄 것이라 판단하였다. 따라서 이와 같은 URL을 이상치로 분류하여 제거하였다. 또한 수집한 데이터 셋 중, 동일한 Domain을 가진 데이터를 제거하여 15만개의 데이터 셋으로 줄이고, 15만개의 데이터 중 정상 70%, 피싱 30% 비율로 15666개의 Data를 선택하였다.

3.2. Feature 생성 과정

```
def External_Load_script(url):
    try:
        http = httplib2.Http(timeout = 5)
        status, response = http.request(url)
        tags = BeautifulSoup(response, parse_only=SoupStrainer(['script']))
        good = 0
        bad = 0
        if tags.get('src'):
            bad += 1
        else:
            good += 1
        if bad / (bad+good) == 1:
            return 0
        else:
            return 1
    except:
        print("Exception")
        return 0
```

```
def Length_of_Source(url):
    try:
        res = requests.get(url)
        if len(res.text)> 50000:
            return 1
        else:
            return 0
    except :
        return 0
```

위와 같은 Feature 생성 코드를 직접 코딩하여, 각 sample의 feature 22개를 뽑았다.

3.3. Feature 선정

학습을 위한 Feature 생성에는 위에서 알아본 피싱 사이트와 정상 사이트의 특징 22가지를 기준으로 알고리즘을 직접 구현하여 데이터 셋에 적용하였다.

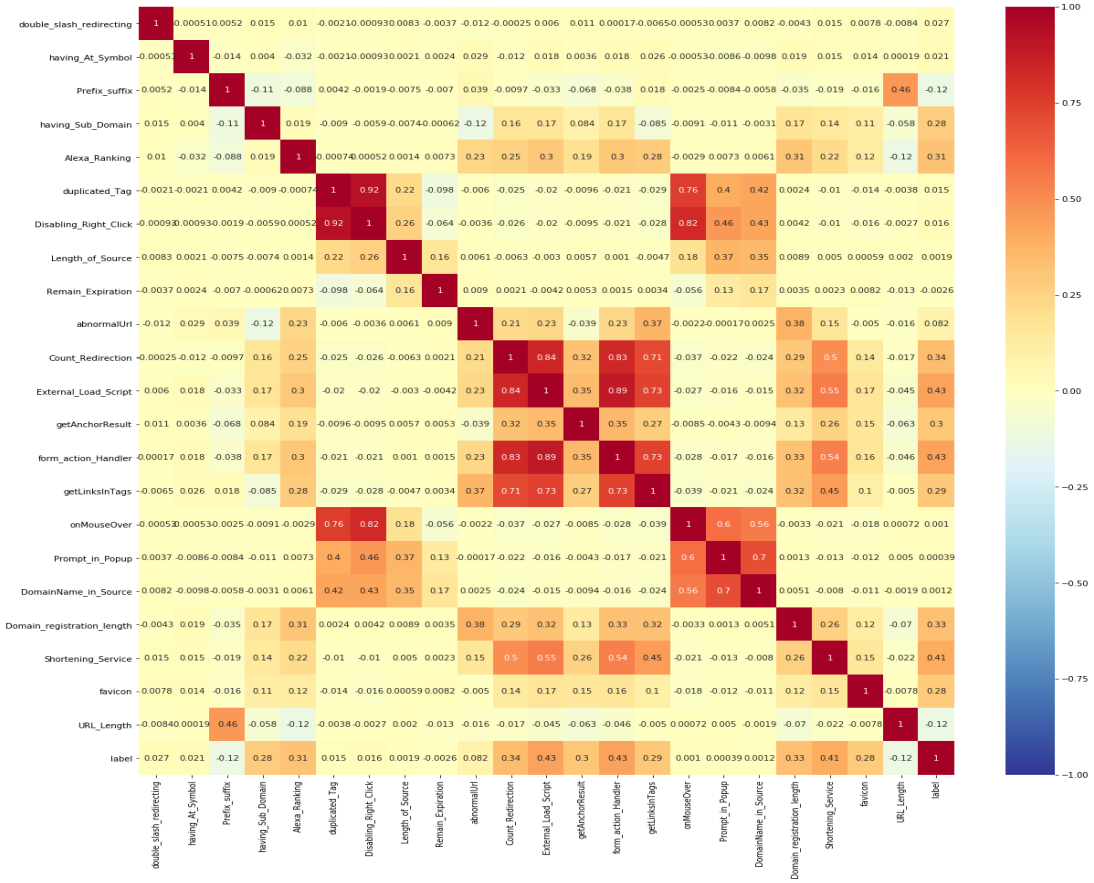
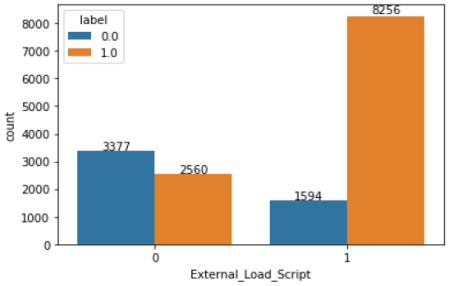
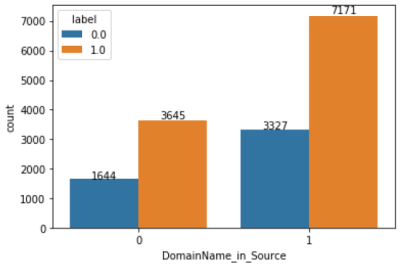


그림 1 전체 Feature의 Heatmap

High correlation	Low correlation
	
External_Load_Script	DomainName_in_Source

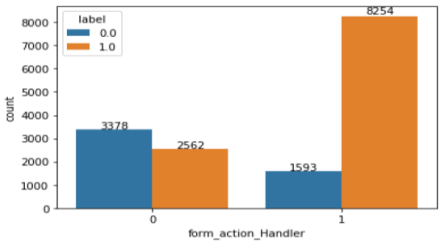
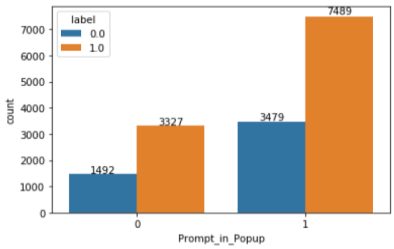
	
form_action_Handler	Prompt_in_Popup

표 3 feature correlation example

위와 같이 Heatmap상 Correlation값이 낮은 데이터들과 높은 데이터들의 일부를 EDA 해보면 확연한 차이가 있다는 것을 알 수 있다. 따라서 Correlation값이 0.1 이상인 값을 가진 Feature들을 뽑아 최종적으로 22개의 Feature중 12개의 Feature를 선정하였다.

3.4. 새롭게 생성한 데이터 셋

	having_Sub_Domain	Alexa_Ranking	Count_Redirection	External_Load_Script	getAnchorResult	form_action_Handler	getLinksInTags	Domain_registration_length	Shortening_Service	favicon
0	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	0	1	1	1	1	1
3	1	1	0	0	0	0	1	1	0	0
4	1	1	1	1	0	1	1	1	1	1
...	...	...	...	...	...	...	...	...	...	...
15783	1	0	1	1	0	1	1	1	1	0
15784	1	0	1	1	0	1	1	1	1	0
15785	1	0	0	0	0	0	0	0	1	1
15786	1	1	1	1	1	1	1	1	1	0
15787	1	0	1	1	0	1	1	0	1	1

15666 rows × 10 columns

그림 2 알고리즘 학습용 데이터 셋

데이터 셋의 전체 22개의 Feature 중 Correlation이 높은 12개의 Feature를 선정하였다. 또한 전체 Sample 15788개 중, Outlier로 판별된 122개의 Sample을 제거한 15666개의 새로운 데이터 셋을 생성하였다.

4. Data Modeling

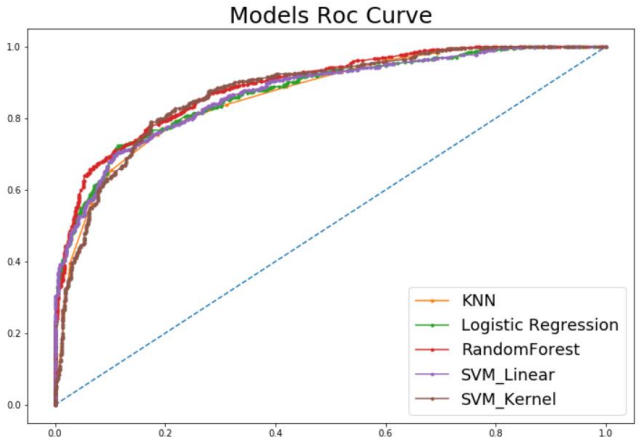
4.1. Algorithm

기계 학습을 하기 위한 알고리즘으로는 대표적인 분류 알고리즘인 KNN, RF, SVM과 회귀 알고리즘인 Logistic Regression을 사용하였다. 위 알고리즘이 선정된 이유는 전처리 과정을 거친 데이터 셋의 feature가 모두 1과 0으로 이루어진 카테고리형 데이터이기 때문이다.

모든 알고리즘에서 Hyperparameter를 선정하는 과정은 GridSearch-CV를 사용하여 선정하였다.

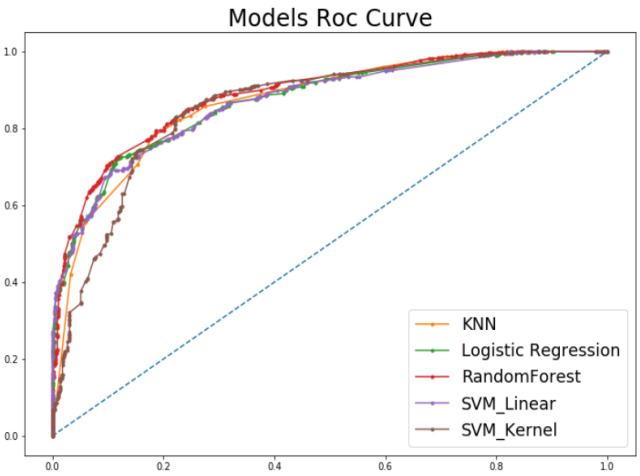
먼저 선정한 feature들이 잘 선정되었는지 둘을 비교해보았다.

22 Features



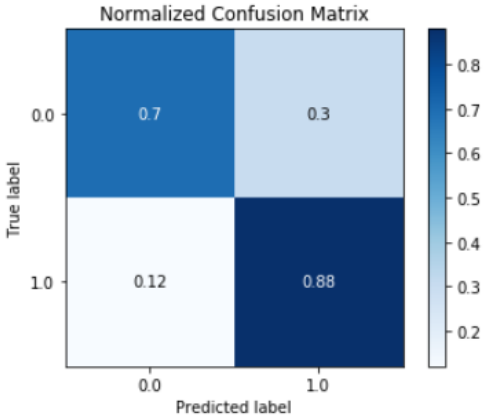
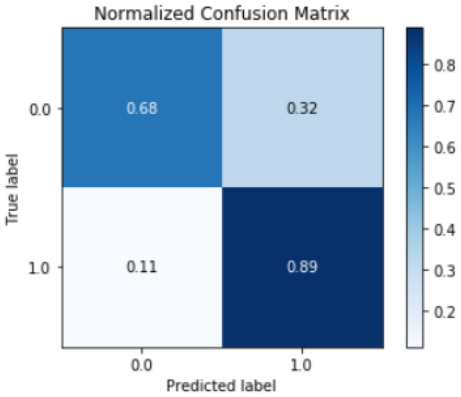
	Model	AUC
0	KNN	0.862031
1	Logistic	0.872105
2	Random_Forest	0.889422
3	SVM_Linear	0.871815
4	SVM_Kernel	0.874478

12 Features



	Model	AUC
0	KNN	0.868123
1	Logistic	0.872171
2	Random_Forest	0.885689
3	SVM_Linear	0.869749
4	SVM_Kernel	0.858659

Classification 알고리즘에선 정확도만으로 알고리즘의 성능을 따지기에는 충분하지 않다. 따라서 Recall과 Fall-out을 사용하는 ROC Curve를 피싱 사이트 판별 알고리즘 평가지표에 사용하는 것이 적절하다고 판단하였다. 그러므로 선정한 Hyperparameter를 사용하여 각 알고리즘의 ROC curve를 구하고 AUC를 계산한 결과, Random Forest 알고리즘에서 AUC 값이 0.89으로 가장 높게 나타났다.

22 Features	12 Features
	

위의 표는 Random Forest에서 최적의 Hyperparameter를 사용하여 학습한 경우 각 Feature의 수에 따른 테스트 set에서의 Confusion Matrix이다.

5. Conclusion

5.1. 결과 분석

1) Feature 선정과정을 통한 Feature 제거 전과 제거 후의 차이.

위의 EDA를 통해 결과 예측에 도움이 되지 않는다고 판단한 Feature 10개를 제거하여 정확도 및 AUC 값의 증가를 기대하였다. 제거 후의 정확도와 AUC 값은 전체 Feature를 선택했을 때와 큰 차이는 없었다. 하지만 마지막 테스트 과정에서 제거 후의 Feature로 학습된 모델에서 무작위로 선택한 피싱, 정상 사이트를 테스트한 결과, 제거 전 테스트 결과와 비교하여 더 잘 판별하였다.

따라서 최종 알고리즘에 10개의 feature를 제거하는 것으로 결론을 내렸다.

그림 3. 22 feature 학습 결과

그림 4. 12 feature 학습 결과

2) 알고리즘 학습결과를 통한 피싱사이트 탐지 가능성

ROC 커브를 통하여 피싱사이트를 가장 잘 탐지하는 RandomForest를 최종 모델로 선정하였고 AUC값은 0.89정도로 높은 값을 가지며 fall-out값을 통하여 Phishing site를 76%정도의 확률로 잡을 수 있다는 것을 확인할 수 있다.

따라서 위와 같은 feature 알고리즘들과 RandomForest 알고리즘을 사용하여 피싱사이트로 의심가는 URL을 접속 전에 먼저 체크해보고 탐지할 수 있다.

5.2. 개선 방안

1) Feature 구현 알고리즘 개선

URL을 통한 Feature 알고리즘들을 구현하였다. 웹 크롤링을 사용하는 알고리즘이 많았기에 이미 서비스가 종료된 사이트나 오류로 인한 크롤링이 불가능할 때를 대비하여 Timeout을 걸어놨다. 이 부분에서 정상 사이트와 피싱 사이트의 구분이 모호해지는 부분이 많았기 때문에 이를 개선시킬 필요가 있다.

2) 현재 서비스 종료된 사이트에 대한 해결방안 강구 및 데이터 셋 전처리 과정 개선

보통 피싱사이트들은 적발되면 사이트를 서비스를 종료하기 때문에 Kaggle 데이터 셋이나 PhishTank에 있는 피싱사이트의 Data는 대부분 서비스가 종료되어 있었다. 따라서 URL로만

사용할 수 있는 Feature 알고리즘은 상관이 없었지만 크롤링을 사용하는 알고리즘들은 정확하게 Feature들을 구별할 수 없었다. 이에 대한 해결방안이 필요하다고 생각된다.

또한 Kaggle의 데이터 셋에선 정상사이트에서도 서비스를 종료한 사이트도 많았고 같은 도메인 주소를 가진 Data를 제거하였음에도 15만개의 Data가 있었기에 이를 모두 확인하는 것은 불가능했다. 이 부분에 대해서도 개선을 할 필요가 있다고 생각된다.

6. Reference

[1] Lee, Jin Lee, Park, Doo Ho, and Lee, Chang-Hoon. "웹사이트 특징을 이용한 휴리스틱 피싱 탐지 방안 연구." 정보처리학회논문지:컴퓨터 및 통신 시스템 4, no. 10 (2015.10.31)

[2] Rami M. Mohammad, Fadi Thabtah, Lee McCluskey. "Phishing Websites Features"

[3] 홍준표 외 4명. "피싱 웹사이트 URL 분류를 위한 컨볼루션 게이팅 신경망 기반 보안전문가 지식 퓨전." 한국정보과학회 학술발표논문집, (2019.12)

[4] Altyeb Altaher. "Phishing Websites Classification using Hybrid SVM and KNN Approach." (IJACSA) International Journal of Advanced Computer Science and Applications. Vol. 8, No. 6 (2017)

[5] 심영호. "TLD Zone 파일을 활용한 피싱사이트 탐지 기법에 관한 연구." 석사학위. 동국대학교 국제정보대학원. (2014)