



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원 저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리와 책임은 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)



석사학위논문

IDF와 문자열 특징을 이용한
머신러닝 기반 악성 URL 탐지

Machine Learning-Based Malicious URL
Detection Using IDF and String Features

김 애 리

한양대학교 대학원

2023년 2월

석사학위논문

IDF와 문자열 특징을 이용한
머신러닝 기반 악성 URL 탐지

Machine Learning-Based Malicious URL
Detection Using IDF and String Features

지도교수 임 을 규

이 논문을 공학 석사학위논문으로 제출합니다.

2023년 2월

한양대학교 대학원

정보보안학과

김애리

이 논문을 김애리의 석사학위 논문으로 인준함

2023년 2월

심사위원장 : 박희진



심사위원 : 임을규



심사위원 : 체동규



한양대학교 대학원

차 례

그림 차례	iii
표 차례	iv
수식 차례	v
국문요지	vi
제1장 서론	1
제1절 연구의 필요성	1
제2절 연구 개요	3
제2장 배경 지식	4
제1절 URL의 구조	4
제2절 TF-IDF	5
제3절 Information Gain	6
제3장 관련 연구	7
제1절 악성 URL 탐지 방법	7
제2절 URL 기반 특징	8
제3절 TF-IDF를 이용한 특징	10
제4장 악성 URL 탐지 시스템	11
제1절 악성 URL 탐지 시스템 구조	11
제2절 데이터셋 구성 방법	12
제3절 악성 URL 탐지를 위해 사용한 특징	12
제5장 실험 및 결과	15
제1절 실험 환경	15

제2절 실험 데이터 구성	15
제3절 평가지표	17
제4절 실험 내용과 결과 분석	19
제4.1절 IDF 특징의 여부에 따른 머신러닝 성능	21
제4.2절 IDF 특징과 TF-IDF 특징 비교	22
제4.3절 Information gain을 이용한 특징 개수 제한에 대한 성능	24
제6장 결론 및 향후 연구	28
제7장 참고 문헌	29



그 림 차 례

[그림 1] 악성 URL 공격 추이	1
[그림 2] URL의 구조	4
[그림 3] 실험 전체 개요	11
[그림 4] 특징 28개에 대한 Information gain 값	25

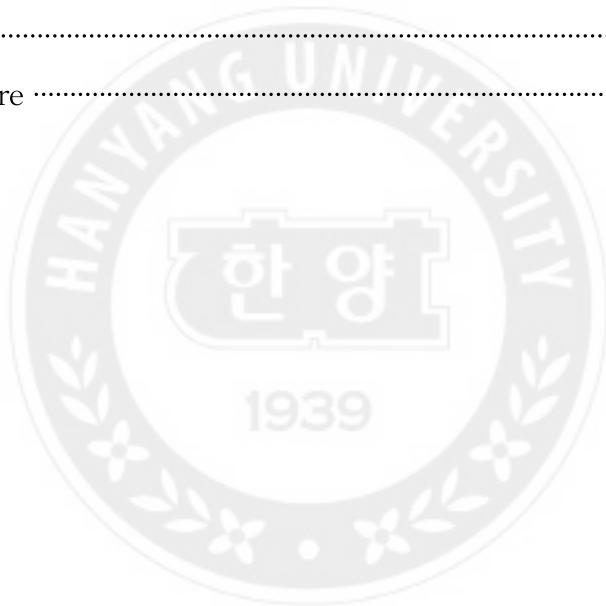


표 차 례

[표 1] 문자열 기반 특징과 IDF 특징	14
[표 2] 전체 데이터셋 구성	15
[표 3] 실험 데이터셋 구성	16
[표 4] Confusion matrix	17
[표 5] 하이퍼파라미터 튜닝	20
[표 6] IDF 특징에 대한 머신러닝 성능	21
[표 7] 문자열 기반 특징 및 IDF 특징을 이용한 머신러닝 결과	22
[표 8] IDF 특징의 포함 여부에 따른 성능 차이	22
[표 9] TF-IDF 특징에 대한 머신러닝 결과	23
[표 10] 문자열 기반 특징 및 TF-IDF 특징을 이용한 머신러닝 결과	23
[표 11] Information gain 상위 10개	26
[표 12] 특징 개수 제한에 따른 성능 비교	27

수식차례

[수식 1] TF-IDF	5
[수식 2] Information Gain	6
[수식 3] Entropy	6
[수식 4] Accuracy	17
[수식 5] Precision	17
[수식 6] Recall	18
[수식 7] F1-score	18



국문요지

피싱 공격은 매해 꾸준히 증가하고 있으며 그에 따라 피해 규모 역시 커지고 있다. 피싱은 개인뿐만 아니라 기업을 상대로 막대한 피해를 입히며 피싱 공격의 형태는 날로 진화되고 있다. 피싱의 증가에 맞서 피해를 최소화하기 위해서는 빠른 악성 URL 탐지가 중요하다.

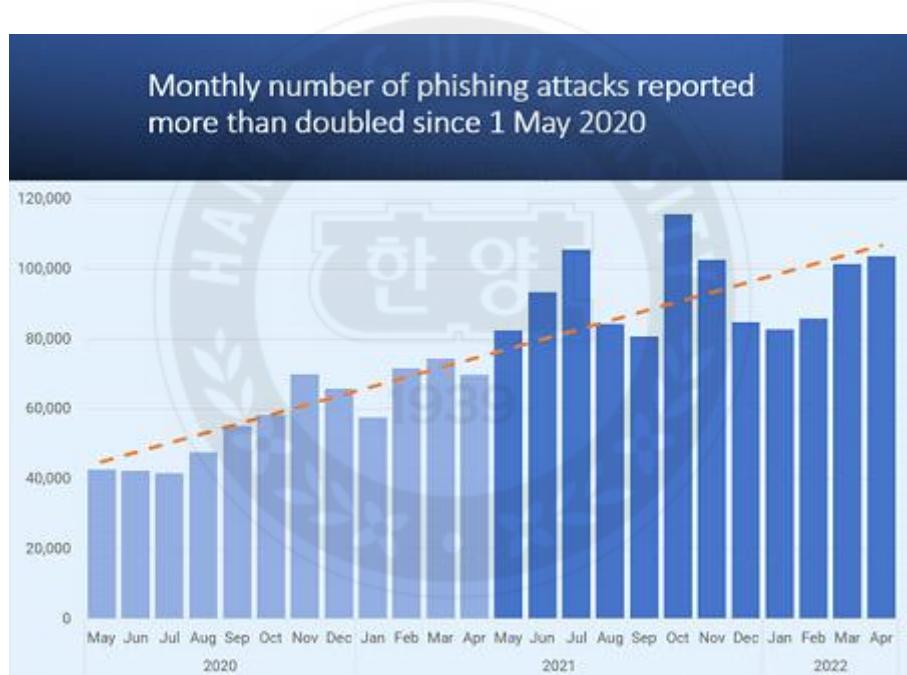
본 논문에서는 상대적으로 응답 시간이 긴 타사 서비스를 제외한 URL 자체에서 추출한 특징만을 사용하여 악성 URL을 빠르게 탐지하는 것을 목표로 한다. 공격자들은 악성 URL인 것을 숨기기 위해 URL의 길이를 늘리는 경향이 있는데 그 중 path 부분을 늘리기도 한다. 본 연구에서는 악성 URL과 정상 URL의 path 패턴이 다르다는 것을 가정하고 IDF를 적용한 특징을 사용했다. Information gain을 통해 특징 중요도를 매긴 결과, IDF 특징은 특징 28개 중 상위 5위로 15.94%의 높은 중요도를 보였다.

URL 자체에서 추출한 문자열 기반 특징과 IDF 특징을 이용하여 악성 URL을 머신러닝으로 탐지한 결과, RF 분류기를 사용했을 때 정확도 92.66%, F1-score 92.65%, AUC 92.66%로 3가지 분류기 중에서 가장 좋은 분류 성능을 보였다.

제1장 서 론

제1절 연구의 필요성

피싱(Phishing)은 개인 정보(Private data)와 낚시(Fishing)의 합성어로 이메일, 문자 메시지, 전화, 웹사이트를 통해 사람들의 민감한 개인정보나 금융정보를 빼가는 사기 수법이다. 피싱 사이트는 합법적인 사이트인 것처럼 위장하여 사용자들의 개인 정보를 유도하거나 악성코드를 유포하여 감염시킨다.



[그림 1] 악성 URL 공격 추이

위 [그림 1]은 Interisle Consulting Group[1]에서 발표한 악성 URL 공격 추이를 나타낸다. 피싱 공격은 2020년부터 꾸준히 증가했으며 2022년에는 2배 이상 증가하였다. 또 APWG(Anti-Phishing Working Group)[2]의 피싱 활동

동향 보고서에 따르면, 2022년 1분기에만 총 1,025,968건의 피싱 공격이 관찰되었고 분기별 총계가 100만 건이 넘은 것은 처음으로 월 68,000 ~ 94,000건의 공격을 관찰한 2020년 초 이후 3배 이상 증가했다고 발표했다. 2022년 기준 피싱은 두 번째로 흔한 침해 원인이었으며 침해 비용이 평균 491만 달러로 가장 많이 발생했다.

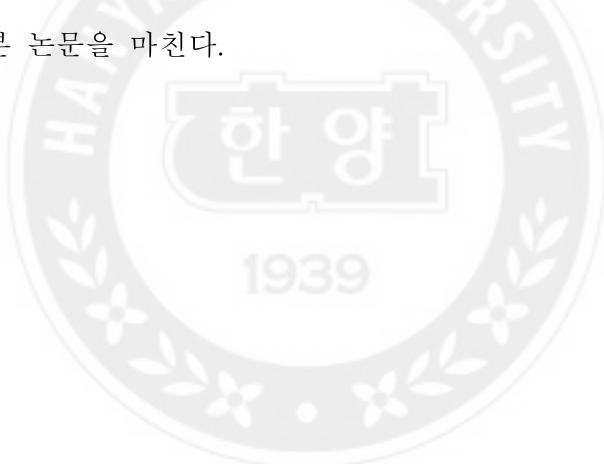
피싱은 사용자들을 속이기 위해 더 교묘해진 사회공학적 기법으로 진화되고 있다. 피싱 공격은 이메일뿐만 아니라 문자 메세지, 소셜 네트워크까지 영역을 확장하여 공격 대상으로 삼는다. 마치 정상적인 사용자나 회사인 것처럼 사칭하여 악성 URL에 접속하게끔 유도한다. 불특정 다수를 대상으로 하는 일반적인 피싱도 있지만 특정 사용자나 회사를 대상으로 하는 스파이피싱도 있다. 이는 특정 대상을 표적으로 삼기 때문에 정교한 공격에 속수무책으로 당할 수 있다.

사용자가 악성 URL을 구별하는 것은 어려운 일이며 피싱의 대응책으로 여러 탐지 방법들이 제시되어왔다. 해당 방법으로 리스트 기반 탐지, URL 기반 탐지, 콘텐츠 기반 탐지 방법이 있으며 대부분의 연구에서 머신러닝 알고리즘을 사용했다. 악성 URL의 증가에 맞서 정확하게 탐지하는 것도 중요하지만 세로 데이 공격에 맞설 수 있는 빠른 탐지도 중요하다. 그렇기 때문에 응답 시간이 오래 걸리지 않는 특징들을 추출하여 악성 URL을 빠르게 탐지하는 것을 목표로 한다.

제2절 연구 개요

본 논문에서는 빠른 탐지를 위해 타사 서비스에 의존하지 않고 URL 자체에서 추출할 수 있는 문자열 특징을 이용한 머신러닝 기반 악성 URL 탐지 방법을 제안한다. 또 URL의 path 부분에 TF-IDF의 IDF를 적용한 특징이 유용한지 Information gain을 통해 확인하고 탐지에 유의미한 특징들을 선별해낸다.

본 논문의 내용은 다음과 같다. 2장에서는 악성 URL 탐지와 관련된 배경지식을 설명하고 3장에서는 악성 URL 탐지 방법과 악성 URL 탐지에 쓰이는 특징들에 대해 살펴본다. 4장에서는 악성 URL 탐지 시스템에 대해 소개하고 5장에서는 실험방법과 결과를 설명하며 마지막 6장에서는 결론 및 향후 연구계획을 끝으로 본 논문을 마친다.



제2장 배경지식

제1절 URL의 구조

URL은 Uniform Resource Locator의 약어로 네트워크 상에서 웹페이지, 이미지, 비디오 등과 같은 리소스의 위치를 알려주기 위해 쓰인다. URL은 크게 Scheme, Authority, Path, Query, Fragment 5가지 요소로 구성되며 [그림 2]는 URL의 구조를 보여준다.

Scheme	Authority	Path	Query	Fragment
http	//www.detection.com:80/test/phishing/result.html	?name=knn&score=90	#content	

[그림2] URL의 구조

- Scheme : 리소스에 접근할 때 사용하는 프로토콜을 나타내며 주로 HTTP 프로토콜을 사용한다.
- Authority : 도메인과 포트로 구성되며 도메인은 IP 주소를 쉽게 기억할 수 있게 부여한 이름이고 포트는 네트워크 서비스 유형을 식별하는 논리단위로 주로 http는 80번, https는 443번을 사용한다.
- Path : 웹 서버의 리소스에 대한 경로이다.
- Query : 웹 서버에 제공하는 매개변수이다. ‘?’ 기호로 시작하고 키와 값은 ‘key = value’ 형식을 가지며 추가적인 파라미터는 ‘&’ 기호로 연결한다.
- Fragment : 보통 한 페이지 내에서 특정 부분으로 이동할 때 사용한다.

제2절 TF-IDF

TF-IDF (Term Frequency–Inverse Document Frequency)[18]는 단어빈도(TF)와 역문서 빈도(IDF)를 곱한 값으로 문서에 나오는 단어들에 대해 중요도에 따라 가중치를 부여하는 방식이다. TF(Term Frequency)는 특정 문서에서 특정 단어가 나오는 빈도이고 DF(Document Frequency)는 특정 단어가 나오는 문서의 빈도이다. IDF(Inverse Document Frequency)는 DF의 역수이다. 즉 특정 문서에서만 자주 나오는 단어는 높은 가중치를 주는 반면, 여러 문서에서 전반적으로 자주 나오는 단어는 폐널티를 준다.

TF-IDF를 수식으로 나타내면 아래 [수식 1]과 같다.

$$tfidf(t, d, D) = tf(t, d) \times idf(t, D)$$
$$tf(t, d) = 0.5 + \frac{0.5 \times f(t, d)}{\max \{f(w, d) : w \in d\}}$$
$$idf(t, D) = \log \frac{|D|}{|\{d \in D : t \in d\}| + 1}$$

(t : 특정 단어, d : 특정 문서, D : 전체 문서)

[수식 1] TF-IDF

제3절 Information Gain

Information Gain(IG)[19]은 어떤 기준으로 데이터를 분류할 때 분류 전의 불순도와 분류 후의 불순도의 차이를 말하며, 각 feature가 데이터를 target에 따라 얼마나 정확하게 잘 분류하는지 지표로 사용된다.

불순도(impurity)는 여러 클래스가 섞여있는 정도를 뜻하며 불순도가 작을수록 클래스가 잘 분리됨을 의미하고 information gain 값은 커진다. 불순도를 측정하는 지표에는 지니계수와 엔트로피가 있다. 보통의 경우, 엔트로피가 지니계수보다 더 좋은 성능을 보이므로 본 실험에서는 엔트로피를 이용하여 information gain을 구한다.

$$\text{InformationGain} = \text{Entropy}(\text{before}) - \text{Entropy}(\text{after})$$

[수식 2] Information Gain

$$\text{Entropy} = - \sum_i^c p_i \log_2 p_i$$

(c : 클래스, p_i : 클래스 i 를 뽑을 확률)

[수식 3] Entropy

제3장 관련 연구

악성 URL을 탐지하기 위해 수십 년간 여러 방법들이 제안되어 왔다. 그동안 진행해왔던 연구들의 악성 URL 탐지 방법과 악성 URL 탐지에 쓰인 특징들을 살펴보려고 한다.

제1절 악성 URL 탐지 방법

악성 URL 탐지 방법에는 크게 리스트 기반, URL 기반, 콘텐츠 기반 탐지 방법이 있다.

리스트 기반 탐지 방법은 화이트리스트와 블랙리스트로 나뉜다. 블랙리스트는 악성 URL 및 IP 주소를 관리하고 화이트리스트는 무해한 URL을 관리한다. 리스트 기반 탐지 방법은 간단한 탐지 방법이지만 새로 생성된 악성 URL 탐지에는 약하다. 또 URL의 일부가 변경하는 경우에 탐지에 실패할 가능성이 크다. 그렇기 때문에 제로 데이 공격에 약하며 최신 상태를 유지하기 위해 자주 업데이트하는 것이 필요하다 [5], [8], [10].

콘텐츠 기반 탐지 방법은 웹페이지에 있는 내용을 분석해서 추출된 특징을 이용하여 탐지하는 방법이다. 하이퍼링크 관련 특징, CSS 관련 특징, 시각적 유사성 특징 등 HTML 소스 코드를 기반으로 하는 특징들을 추출한다 [6], [12], [13]. Zhang, et al. [9]은 TF-IDF를 이용하여 웹콘텐츠의 합법 여부를 따졌다. 웹페이지에 있는 각 용어를 TF-IDF 점수로 환산하고 그 중 상위 5개 용어를 Google과 같은 검색 엔진에 검색하는 방식을 사용하였다. 웹페이지의 도메인 이름이 상위 N개 검색 결과의 도메인 이름과 일치하면 합법, 그렇지 않으면 악성 URL로 간주했다. 콘텐츠 기반 탐지 방법은 URL을 실제로 실행해야하기 때문에 안전에 대한 문제와 악성 URL 분류 시간이 오래 걸린다는 문제점이 있다[3].

URL 기반 탐지 방법은 URL에서 관련 특징을 추출해서 탐지하는 방법이다. URL의 구성 요소에서 문자열 기반 특징을 추출하거나 타사 서비스를 이용하여 URL의 악성 여부를 따진다. 비교적 다른 탐지 기법보다 탐지 속도가 빠르고 URL 자체에서 특징을 추출하는 것이므로 안전하게 탐지할 수 있다.

제2절 URL 기반 특징

Rupa, Ch, et al. [4]는 Random Forest 분류기를 사용하여 악성 URL을 식별하는 시스템을 제안했다. 악성 URL 탐지를 위해 어휘적 특징, URL 특징, 악성 키워드 3종류의 특징을 이용하였다. URL 특징은 도메인의 크기, 연령, 호스트 세부 정보 및 페이지 순위로 추출하고 악성 키워드는 단어 확률을 이용하여 추출하였다.

Gupta, Brij B., et al. [5]은 타사 서비스 특징을 사용하지 않고 짧은 응답 시간으로 실시간 환경에서 악성 URL을 탐지할 수 있는 솔루션을 제안했다. 9 개의 어휘 기반 특징을 이용하고 특징의 중요도를 구하기 위해 spearman, k-best, random forest 3가지 알고리즘을 사용하였다. URL의 어휘적 특성만을 이용하기 때문에 밀리초 내에 피싱 공격을 탐지할 수 있다고 한다.

Yadollahi, Mohammad Mehdi, et al. [6]는 URL 기반 특징, HTML 기반 특징, 통계 기반 특징 및 자연어 처리(NLP) 기반 특징들을 사용하여 실시간 피싱 방지 시스템을 제안했다. URL에서 26개, 웹페이지 콘텐츠에서 8개, 네트워크 활동에서는 4개의 특징을 추출하였으며 규칙 기반 온라인 학습 시스템인 XCS를 이용하여 악성 URL을 식별했다.

Sadique, Farhan, et al. [7]는 악성 URL의 자동 탐지를 위한 프레임워크를 제안했다. 어휘적 특징, 호스트 기반 특징, GeoIP, WHOIS 4종류의 특징을 사용하였고 drop-column 방식을 이용하여 각 특징의 중요도를 계산했다. 142개의 특징 중에서 상위 20개의 절반이 어휘적 특징이었고 URL 문자열의 엔트

로피 특징이 2.75%의 기여도를 갖는 가장 중요한 특징인 것을 보여주었다.

Aljofey, Ali, et al. [8]는 웹페이지의 URL 특징만을 이용하여 피싱 사이트를 빠르고 정확하게 탐지할 수 있는 딥러닝 기반 솔루션을 제안했다. Hand-crafted, 문자 임베딩, 문자 수준 TF-IDF, 카운트 벡터 4종류의 특징을 사용했고 총 95개의 특징을 이용하여 문자 수준 CNN을 훈련시키고 피싱 탐지 모델을 구현하였다.

Korkmaz, M, et al. [11]는 실행 시간을 줄이기 위해 타사 서비스 특징을 사용하지 않는 것을 목표로 하는 머신러닝 기반 피싱 탐지 시스템을 제안했다. URL 분석을 통해 58개의 특징을 추출하고 RF 분류기를 통해 좋은 특징들을 선별하였다. 선별된 48개의 특징을 사용하여 머신러닝을 훈련시켰다.

Rao, et al. [12]는 기준의 안티피싱 기술의 단점을 극복하기 위해 URL, 소스 코드 및 타사 서비스에서 추출한 휴리스틱 특징을 기반으로 하는 악성 URL 분류 모델을 제안했다. 실험을 통해 타사 서비스 특징이 모델 성능에 상당한 영향을 미친다는 것을 보여주었다.

Liu, Chunlin, et al. [14]는 웹사이트 URL의 통계적 분석과 머신러닝을 결합하여 악성 URL을 보다 정확하게 분류할 수 있는 시스템을 제안했다. 악성 URL의 본질적인 특징을 추출하기 위해 URL의 문자 빈도 특징에 더 주의를 기울였다. RF를 이용한 구조적 특징의 중요도 평가 실험에서 path 깊이 특징이 분류 효과가 가장 우수하다는 것을 보여주었다.

Sahingoz, Ozgur Koray, et al. [15]는 7가지 머신러닝과 자연어 처리 기반 특징을 사용하는 실시간 안티피싱 시스템을 제안했다. Word Vector, NLP(Natural Language Processing) 기반 및 Hybrid 특징을 사용하였고 3종류의 특징을 비교한 결과, NLP 특징이 Word Vector 특징보다 평균 10.86%가량 더 높았고 Hybrid 특징을 사용했을 때는 NLP 특징과 Word Vector 특징 보다 각각 약 2.24%, 13.14%로 시스템 성능이 향상되는 것을 보여줬다.

[4], [7], [12]의 저자들은 연구에서 WHOIS, Page Rank 등과 같은 타사 서비스를 이용한 특징을 사용하였다. [12]의 연구에서 타사 서비스 특징과 다른 종류의 특징들을 비교하는 실험을 통해 타사 서비스의 특징이 악성 URL 탐지에 있어 중요하다는 것을 보여주었다. 그러나 타사 서비스의 특징은 URL 그 자체에서 추출하는 특징들과 달리 서비스 쿼리로 인한 네트워크 대기 시간이 발생한다. [7]에서는 특징별로 수집에 걸리는 시간을 측정하였다. 어휘적 특징은 평균적으로 3ms 안에 추출되었지만 WHOIS 특징과 GeoIP 특징은 각각 평균적으로 약 1453ms, 4060ms가 소요됨을 보여주었다. [7]의 저자들은 어휘적 특징만을 사용한다면 타사 서비스로 인해 생기는 대기 시간을 절약할 수 있다는 것을 시사했다. 타사 서비스의 긴 응답 시간 문제를 제기하며 타사 서비스를 배제한 특징들을 이용한 연구도 있었다 [5], [6], [11].

제3절 TF-IDF를 이용한 특징

악성 URL을 탐지하기 위해 TF-IDF를 이용한 연구는 CANTINA[9]가 대표적이다. CANTINA는 웹콘텐츠에 있는 용어들을 TF-IDF 점수로 환산하고 그 중 상위 5개 용어를 뽑아 Google과 같은 검색엔진에 어휘 서명으로 제공한다. 어휘 서명으로 얻은 검색 결과에 따라 악성 URL 여부를 판별했다.

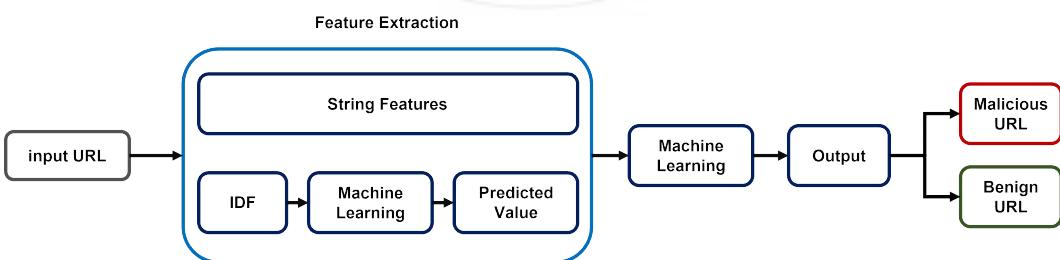
콘텐츠가 아닌 URL 자체에서 특징을 추출하기 위해 TF-IDF를 이용한 연구도 있었다. [8]에서는 URL에 문자 수준의 TF-IDF를 적용하였고 의미 없는 단어를 제외하기 위해 단어를 최대 5000개로 제한하였다. [10]에서도 URL 자체에 TF-IDF를 적용하여 328,992개의 단어 사전을 얻었다. URL의 수에 따라 증가하는 TF-IDF의 사전 크기의 메모리 문제를 해결하기 위해 희소 벡터로 변환하여 사용하였다.

제4장 악성 URL 탐지 시스템

본 논문에서는 문자열 기반 특징과 IDF 특징을 이용해서 머신러닝 기반으로 악성 URL을 탐지하는 시스템을 제안한다.

제1절 악성 URL 탐지 시스템 구조

본 논문에서 제안하는 시스템 구조는 [그림 3]과 같다. 특징 추출은 URL 자체에서 특징을 추출하는 것으로 문자열 기반 특징과 IDF 특징으로 나뉜다. 문자열 기반 특징은 입력된 URL에 대해 길이, 개수, 비율, 여부, 위치를 기반으로 추출된다. IDF 특징은 URL의 path에 있는 단어들을 이용하여 path의 단어들을 IDF 값으로 변환하고 머신러닝을 통해 얻어낸 예측값으로 추출된다. 이 때 사용되는 머신러닝은 LR(Logistic Regression), LGBM(Light Gradient-Boosting Machine), RF(Random Forest), DT(Decision Tree)이고 이 4종류 머신러닝 중 가장 좋은 성능을 보이는 1종류의 머신러닝을 이용하여 예측값을 얻는다. 이와 같이 추출된 문자열 기반 특징과 IDF 특징을 이용하여 WKNN(Weighted k-Nearest Neighbors), RF, DT 3종류의 머신러닝을 훈련시켜서 입력된 URL이 악성인지, 정상인지 판별한다.



[그림 3] 실험 전체 개요

제2절 데이터셋 구성 방법

데이터셋은 과적합을 막기 위해 IDF 특징을 포함한 문자열 기반 특징을 학습시키기 위한 데이터셋 A와 IDF 특징을 추출하기 위한 데이터셋 B로 나눈다. IDF 값을 구하기 위해서 URL의 path 부분을 구성하고 있는 단어들을 이용하여 단어 사전을 만든다. 데이터셋 B를 통해 만들어진 단어 사전은 데이터셋 A에도 그대로 적용하여 URL의 path를 IDF 값으로 변환시키고 머신러닝을 통해 예측값을 얻는다.

제3절 악성 URL 탐지를 위해 사용한 특징

악성 URL을 탐지하기 위해 사용한 특징은 URL 자체에서 추출한 문자열 기반 특징과 IDF 특징이다. 문자열 기반 특징은 길이, 개수, 비율, 여부, 위치 기반 특징으로 구성했다.

1. 길이 기반 특징 (F1 ~ F6) : 공격자들은 악성 URL인 것을 감추기 위해 URL의 특정 구성 요소 길이를 길게 늘리는 경향이 있다. 관련 특징으로 전체 URL의 길이(F1), 도메인 길이(F2), 호스트 길이(F3), path 길이(F4), path 가장 긴 토큰(F5), 긴 쿼리값(F6) 특징들이 있다.
2. 개수 기반 특징 (F7 ~ F16) : 악성 URL은 정상 URL처럼 보이기 위해 특수기호나 숫자를 넣는 경향이 있다. 관련 특징으로 도메인의 점 개수(F7), 도메인 숫자 개수(F8), 도메인 - 의 개수(F9), -의 개수(F10), _의 개수(F11), %의 개수(F12), =의 개수(F13), &의 개수(F14), ?의 개수(F15), ~의 개수(F16) 특징들이 있다.
3. 비율 기반 특징 (F17 ~ F20) : 악성 URL은 정상 URL과 다른 문자 비율을 갖고 있다. 관련 특징으로 문자/숫자 비율(F17), 전체 숫자 비율(F18), 호스트 모음/자음 비율(F19), 대문자 비율(F20) 특징들이 있다.
4. 여부 기반 특징 (F21 ~ F24) : 관련 특징으로 @의 여부(F21), https의 여

부(F22), 파일 확장자 여부(F23), &의 여부(F24) 특징들이 있다.

- https의 여부(F22) : https는 SSL(Secure Sockets Layer)을 적용하여 보안을 강화한 것으로 https가 없는 경우, 악성 URL로 간주한다.
- 파일 확장자 여부(F23) : 공격자는 path에 파일 확장자를 넣어 공격을 일으킬 수 있다. ‘txt’, ‘exe’, ‘js’ 파일 확장자가 있는 경우, 악성 URL로 간주한다.

5. 위치 기반 특징 (F25 ~ F27) : 공격자들은 URL의 구성 요소를 다른 위치에 놓는 경향이 있다. 관련 특징으로 www 위치(F25), TLD 위치(F26), https 위치(F27) 특징들이 있다.

- TLD(Top-level domain) 위치(F26) : 공격자는 도메인 영역 외에 TLD를 넣는 경향이 있다. 실현에 사용된 TLD는 com, org, ru, net, uk, au, in, de, ir, ca [16]이며 도메인 외의 위치에 있는 경우 악성 URL로 간주한다.
- https의 여부(F27) : 공격자는 정상 URL로 보이기 위해 Scheme 영역 외에 ‘https’ 문자열을 넣는 경향이 있다.

6. IDF 특징(F28) : [6], [8]에서는 URL 전체에 TF-IDF 기법을 적용하고 단어 사전의 개수를 제한하여 특징을 추출하였다. 공격자들은 악성 URL임을 숨기기 위해 길이를 늘리는 경향이 있는데 path 부분도 포함된다. Path 부분을 늘릴 때 여러 단어들이 사용되는데 이때 악성 URL과 정상 URL에 쓰이는 단어가 다르다고 가정했다. 악성 URL에서만 자주 등장하는 단어, 정상 URL에서만 자주 등장하는 단어들을 뽑기 위해 TF-IDF 기법을 그대로 적용하는 것이 아닌 IDF 값만 구하여 URL의 path 부분을 변환했다. 변환된 path는 4가지 머신러닝 분류기(LR, LGBM, RF, DT)를 통해 훈련시키고 그 중 좋은 성능을 보였던 분류기로부터 예측값을 받아 특징으로 삼았다.

[표 1] 문자열 기반 특징과 IDF 특징

No.	특징	No.	특징	No.	특징
F1	전체 URL 길이	F11	_의 개수	F21	@의 여부
F2	도메인 길이	F12	%의 개수	F22	https의 여부
F3	호스트 길이	F13	=의 개수	F23	파일 확장자 여부
F4	path 깊이	F14	&의 개수	F24	&의 여부
F5	path 가장 긴 토큰	F15	?의 개수	F25	www 위치
F6	긴 쿼리값	F16	~의 개수	F26	TLD 위치
F7	도메인의 점 개수	F17	문자/숫자 비율	F27	https 위치
F8	도메인 숫자 개수	F18	전체 숫자 비율	F28	IDF
F9	도메인 -의 개수	F19	호스트 모음/자음 비율		
F10	-의 개수	F20	대문자 비율		

[표 3] 실험 데이터셋 구성

데이터셋	구분	개수
데이터셋 A	정상	164,461개
	악성	164,974개
데이터셋 B	정상	82,039개
	악성	81,526개

문자열 기반 특징은 특징마다 값 범위가 다르기 때문에 평균을 0, 분산을 1로 스케일링하는 StandardScaler를 사용하여 표준화하였다.

IDF 특징을 추출하기 위한 실험에서는 훈련과 테스트 데이터를 8:2로, 문자열 기반 특징과 IDF 특징으로 머신러닝을 훈련시키는 실험에서는 7:3으로 분리하여 사용하였다.

제3절 평가지표

본 실험에서는 Accuracy, Precision, Recall, F1-score, AUC 총 5가지 평가지표를 이용하여 평가했다.

[표 4] Confusion matrix

		Predicted class	
Actual Class	True Positive (TP)	False Negative (FN)	
	False Positive (FP)	True Negative (TN)	

- TP (True Positive) : 실제가 True인데 모델에서 True라고 예측한 경우
 - TN (True Negative) : 실제가 False인데 모델에서 False라고 예측한 경우
 - FP (False Positive) : 실제가 False인데 모델에서 True라고 예측한 경우
 - FN (False Negative) : 실제가 True인데 모델에서 False라고 예측한 경우
- Accuracy : 전체 대비 정확하게 예측한 개수의 비율

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

[수식 4] Accuracy

- Precision : 모델이 positive라고 예측한 것 중에서 실제로 positive인 비율

$$Precision = \frac{TP}{TP + FP}$$

[수식 5] Precision

- Recall : 실제 positive인 것 중에서 모델이 positive라고 예측한 비율

$$Recall = \frac{TP}{TP + FN}$$

[수식 6] Recall

- F1-score : Precision과 Recall의 조화평균

$$F1-score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

[수식 7] F1-score

- AUC (Area Under the ROC curve) : ROC curve의 밑면적을 계산한 값으로 분류 모델의 성능을 나타내는 지표로 사용된다. 1에 가까울수록 성능이 우수하다고 판단한다. ROC curve (Receiver-Operating Characteristic curve)는 FPR(False Positive Rate)과 TPR(True Positive Rate)을 각각 x, y축으로 놓은 그래프이다.

제4절 실험 내용과 결과 분석

본 연구에서는 3가지 실험을 진행했다. 첫 번째 실험은 IDF 특징 여부에 따른 머신러닝 성능을 확인하고 두 번째 실험은 IDF 특징과 TF-IDF 특징을 비교한 실험을 진행했다. 세 번째 실험은 Information gain으로 각 특징의 중요도를 확인하고 특징 개수 제한에 따른 머신러닝 성능을 확인하는 실험을 진행했다.

모든 머신러닝 분류기는 10겹 교차 검증을 걸쳤으며, python의 Randomized SearchCV를 사용하여 최적의 하이퍼파라미터를 구하였다. 각 머신러닝 분류기는 아래 [표 5]와 같이 하이퍼파라미터를 고정하여 훈련시켰다. [표 5]에서 데이터셋 A는 IDF 특징을 포함한 문자열 기반 특징을 훈련시키기 위한 데이터셋으로 3종류 머신러닝 분류기(WKNN, DT, RF)을 사용했고 데이터셋 B는 IDF 특징을 만드는 목적인 데이터셋으로 4종류의 머신러닝 분류기(LR, LGBM, RF, DT)를 사용했다.

[표 5] 하이퍼파라미터 튜닝

데이터셋 A		데이터셋 B	
분류기	하이퍼파라미터	분류기	하이퍼파라미터
WKNN	n_neighbors : 10 weights : distance	LR	penalty : l2 max_iter : 100 C : 10
DT	min_samples_leaf : 10 max_depth : 30 criterion : entropy	LGBM	reg_alpha : 0.01 num_leaves : 80 min_child_samples: 15 max_depth : -1 learning_rate : 0.2
RF	n_estimators : 300 min_samples_split : 5 min_samples_leaf : 2 max_depth : 50	RF	n_estimators : 200 min_samples_split : 10 min_samples_leaf : 4 max_depth : 50
		DT	criterion : gini max_depth : 30 min_samples_leaf : 5

제4.1절 IDF 특징의 여부에 따른 머신러닝 성능

첫 번째 실험은 IDF 특징의 사용 여부에 따른 머신러닝 성능을 비교하는 실험이었다. 먼저 URL의 path 부분이 있는 82,039개의 정상 URL과 81,526개의 악성 URL로 구성된 데이터셋 B에서 133,538개 단어를 뽑아냈다. 그 중 상위 15,000개의 단어로 제한하고 path를 IDF 값으로 변환했다. IDF 특징을 구하기 위해 4종류의 머신러닝으로 훈련시켰고 그 중 가장 성능이 좋은 분류기를 골라 예측값을 구했다.

다음 [표 6]은 IDF 특징을 구하기 위해 4종류의 머신러닝을 훈련시킨 결과이다.

[표 6] IDF 특징에 대한 머신러닝 성능

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)
LR	76.22	90.11	58.71	71.10	76.15
LGBM	73.96	92.50	51.96	66.54	73.89
RF	68.68	92.82	39.98	55.89	68.45
DT	61.51	95.04	24.00	38.32	61.38

실험 결과, LR이 정확도 76.22%, F1-score 71.10%, AUC 76.15%로 가장 좋은 분류 성능을 보였고 LGBM은 정확도 73.96%, F1-score 66.54%, AUC 73.89%로 2번째로 나은 성능을 보였다.

RF와 DT는 LR, LGBM에 비해 성능이 좋지 못했으며 Precision이 대체로 높은 반면 Recall이 좋지 않다는 것을 확인하였다. 4종류의 머신러닝 중 LR이 가장 좋은 분류기로 IDF 특징을 추출하기 위한 모델로 사용하게 되었다.

다음 [표 7]은 27개의 문자열 기반 특징과 IDF 특징을 이용해서 3가지 머신러닝을 훈련시켰을 때의 결과로 3가지 분류기 중 RF가 정확도 92.66%, F1-sc

ore 92.65%, AUC 92.66%로 가장 좋은 성능을 보였다.

[표 7] 문자열 기반 특징 및 IDF 특징을 이용한 머신러닝 결과

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)
KNN	91.36	91.79	90.88	91.33	91.36
DT	91.25	92.48	89.83	91.13	91.25
RF	92.66	93.02	92.28	92.65	92.66

아래 [표 8]은 IDF 특징의 포함 여부에 따른 머신러닝 성능 차이를 비교한 것으로 RF 분류기로 훈련 시킨 결과, IDF 특징을 포함했을 때 정확도, AUC는 1.04%, F1-score는 1.05% 더 높게 나온다는 것을 알 수 있었다.

[표 8] IDF 특징의 포함 여부에 따른 성능 차이

RF	IDF 특징 포함 (%)	IDF 특징 제외 (%)	차이 (%)
Accuracy	92.66	91.62	1.04
F1-score	92.65	91.60	1.05
AUC	92.66	91.62	1.04

제4.2절 IDF 특징과 TF-IDF 특징 비교

두 번째 실험은 IDF 특징과 TF-IDF 특징을 비교하는 실험을 진행했다. 본 연구는 URL의 path에 있는 단어들이 악성 URL과 정상 URL에 따라 다르게 구성된다고 가정하여 IDF 특징을 사용했다. Path 부분을 각각 IDF와 TF-IDF로 적용하여 어떤 것이 더 유용한지 확인했다.

[표 9]는 TF-IDF 특징을 구하기 위해 머신러닝을 훈련시킨 결과이다. LR이 정확도 84.44%, F1-score 83.75%, AUC 84.43%로 가장 좋은 분류 성능을 보

였다. 첫 번째 실험의 [표 6]과 비교하면 모든 머신러닝 분류기에서 IDF 특징을 사용했을 때보다 TF-IDF 특징을 사용하는 것이 전체적으로 좋은 결과를 냈다. 가장 성능을 냈던 LR을 기준으로 보면 TF-IDF 특징을 사용할 때가 IDF 특징보다 정확도 8.22%, F1-score 12.65%, AUC 8.28% 정도 더 높았다. 또 제일 낮은 성능을 보여줬던 DT 분류기에서도 정확도 12.86%, F1-score 31.5%, AUC 12.94% 정도 TF-IDF 특징으로 훈련한 결과가 IDF 특징보다 더 좋은 것을 알 수 있다.

[표 9] TF-IDF 특징에 대한 머신러닝 결과

	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	AUC (%)
LR	84.44	87.31	80.47	83.75	84.43
LGBM	83.22	88.66	76.04	81.87	83.19
RF	77.14	83.41	67.55	74.65	77.11
DT	74.37	84.50	59.48	69.82	74.32

[표 10]은 문자열 기반 특징과 TF-IDF 특징을 이용해서 머신러닝을 훈련시킨 결과이다. RF가 정확도 93.45%, F1-score 93.43%, AUC 93.45%로 가장 좋은 분류 성능을 보였고 첫 번째 실험의 [표 7]과 비교해도 정확도, F1-score, AUC에서 대략 0.79% 더 높은 수치를 보였다.

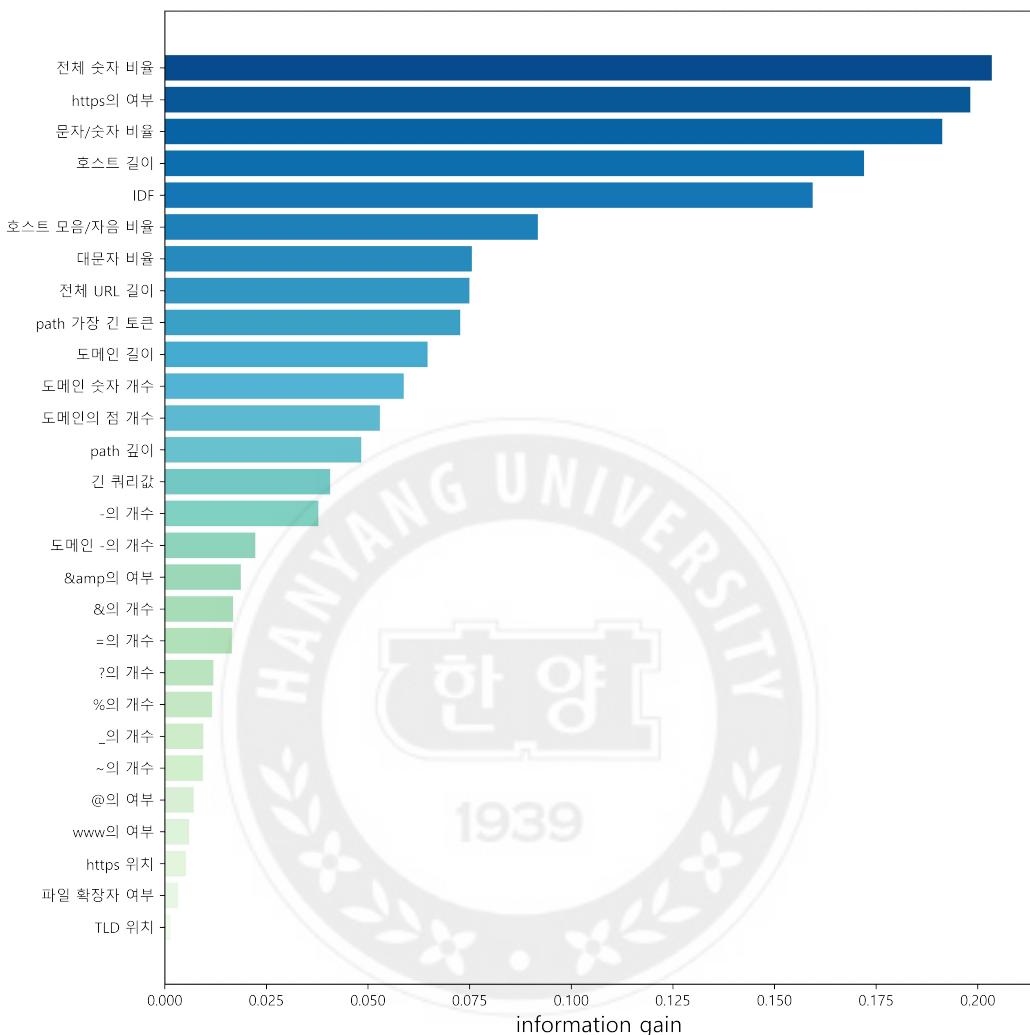
[표 10] 문자열 기반 특징 및 TF-IDF 특징을 이용한 머신러닝 결과

	Accuracy	Precision	Recall	F1-score	AUC
KNN	92.39%	93.22%	91.45%	92.33%	92.39%
DT	92.38%	93.53%	91.08%	92.29%	92.38%
RF	93.45%	93.91%	92.95%	93.43%	93.45%

제 4.3절 Information gain을 이용한 특징 개수 제한에 대한 성능

세 번째 실험은 information gain을 이용해서 각 특징의 중요도를 구하고 특징의 개수를 제한함에 따라 머신러닝 성능이 어떻게 변화되는지 확인하는 실험이다. 많은 수의 특징을 사용하면 노이즈가 발생하고 모델의 복잡도가 증가하여 과적합될 수 있다. 또 특징의 개수에 따라 응답시간이 증가하여 높은 처리 능력이 필요로 하기 때문에 성능에 크게 기여할 수 있는 특징을 선별하는 것이 중요하다[5]. [그림 4]는 특징 28개에 대한 information gain 값을 보여준다.





[그림 4] 특징 28개에 대한 Information gain 값

아래 [표 11]은 상위 10개 특징의 information gain을 보여준다. 대체로 비율 기반 특징들이 높은 information gain을 얻었고 다음으로는 길이 기반 특징이 좋다는 것을 확인할 수 있었다. 가장 높은 information gain을 가진 특징은 ‘전체 숫자 비율’ 특징이었으며 20.34%의 특징 중요도를 갖는다. 다음으로 https의 여부, 문자/숫자 비율, 호스트 길이, IDF 특징이 차례대로 19.81%, 19.12%, 17.20%, 15.94%의 중요도를 가졌다. IDF 특징은 상위 5번째로 높은 information gain을 가졌고 악성 URL과 정상 URL을 구별 짓는데에 15.94%의 중요도를 갖고 있다는 것을 알 수 있었다.

[표 11] Information gain 상위 10개

Information Gain	
전체 숫자 비율	20.34%
https의 여부	19.81%
문자/숫자 비율	19.12%
호스트 길이	17.20%
IDF	15.94%
호스트 모음/자음 비율	9.17%
대문자 비율	7.55%
전체 URL 길이	7.49%
path 가장 긴 토큰	7.26%
도메인 길이	6.46%

다음 [표 12]은 특징 28개를 information gain 하위 5개, 10개, 15개의 특징을 제외했을 때의 머신러닝 성능을 비교한 것이다. 첫 번째 실험에서 가장 좋은 성능을 보였던 RF 분류기를 이용하여 특징 개수에 따른 성능을 확인했다. 먼저 특징을 23개로 제한하면 모든 특징을 사용했을 때보다 정확도 0.06%, F1-score 0.08%, AUC 0.06%가 떨어졌고 특징을 18개로 제한했을 때는 정확도 0.15%, F1-score 0.18%, AUC 0.15%가 떨어졌다. 또 특징을 13개로 제한했

을 때는 정확도 0.88%, F1-score 0.92%, AUC 0.88%가 떨어졌다. 특징을 23개, 18개로 제한했을 때는 0.2% 내외로 떨어졌지만 13개로 제한했을 때는 약 1%가 떨어지는 것을 확인할 수 있었다.

[표 12] 특징 개수 제한에 따른 성능 비교

RF (특징 23개)	Accuracy	92.60%
	F1-score	92.57%
	AUC	92.60%
RF (특징 18개)	Accuracy	92.51%
	F1-score	92.47%
	AUC	92.51%
RF (특징 13개)	Accuracy	91.78%
	F1-score	91.73%
	AUC	91.78%

3가지의 실험을 통해 IDF가 28개 특징 중에 상위 5위로 15.94%의 중요도를 가진다는 것을 확인했고 문자열 기반 특징과 IDF 특징을 같이 사용했을 때 RF 분류기에서 정확도 92.66%, F1-score 92.65%, AUC 92.66%의 성능을 보여줬다. 또 두 번째 실험에서는 IDF 특징보다 TF-IDF 특징이 분류 성능을 더 높인다는 것을 알 수 있었다. 세 번째 실험에서는 information gain을 사용하여 각 특징의 중요도를 확인했다. 그 결과, 문자열 기반 특징에서 비율 기반 특징들이 대체로 높은 중요도를 보였고 다음으로 길이 관련 특징이 높은 중요도를 보였다. 특징 개수를 제한하여 성능을 비교했을 때 특징을 18개, 23개로 제한하면 모든 특징을 사용했을 때보다 성능이 대략 0.2%가 떨어지는 것을 확인했다.

제6장 결론 및 향후 연구

본 연구에서는 악성 URL을 빠르게 탐지하기 위해 타사 서비스를 이용하지 않고 URL 자체에서 특징을 추출하는 문자열 기반 특징과 IDF 특징을 사용하여 머신러닝 기반으로 악성 URL을 탐지하는 방법을 제안했다. 문자열 기반 특징과 IDF 특징을 사용한 결과, RF 분류기에서 정확도 92.66%, F1-score 92.65%, AUC 92.66%로 가장 좋은 분류 성능을 보였다. 또 모델 성능에 크게 기여할 수 있는 특징들을 선별하기 위해 information gain을 사용하여 실험에 사용된 28개 특징의 중요도를 확인한 결과, 대체로 비율 기반 특징과 길이 기반 특징이 높은 기여도를 가진다는 것을 알 수 있었다. IDF 특징은 상위 5위로 악성 URL 분류에 15.94%의 중요도를 가진다는 것을 확인했다.

실험을 통해 IDF 특징의 Recall 값이 좋지 않다는 것을 확인했다. 향후 연구에서 Recall을 개선할 수 있는 방법을 찾고 추가적으로 모델의 성능을 올릴 수 있는 문자열 기반 특징을 찾는 것이 계획이다.

제7장 참고 문헌

- [1] Interisle Consulting Group,
<https://www.interisle.net/PhishingLandscape2021.html>
- [2] APWG, <https://apwg.org/trendsreports/>
- [3] Hong, J., Kim, T., Liu, J., Park, N., & Kim, S. W. Phishing url detection with lexical features and blacklisted domains. In Adaptive autonomous secure cyber systems (pp. 253–267). Springer, Cham. 2020
- [4] Rupa, C., Srivastava, G., Bhattacharya, S., Reddy, P., & Gadekallu, T. R. A machine learning driven threat intelligence system for malicious url detection. In: The 16th International Conference on Availability, Reliability and Security. p. 1–7. 2021.
- [5] Gupta, B. B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A., & Chang, X. A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment. Computer Communications, 175, 47–57. 2021
- [6] Yadollahi, M. M., Shoeleh, F., Serkani, E., Madani, A., & Gharaee, H. An adaptive machine learning based approach for phishing detection using hybrid features. In 2019 5th International Conference on Web Research (ICWR) (pp. 281–286). IEEE. 2019
- [7] Sadique, F., Kaul, R., Badsha, S., & Sengupta, S. An automated framework for real-time phishing url detection. In 2020 10th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 0335–0341). IEEE. 2020
- [8] Aljofey, A., Jiang, Q., Qu, Q., Huang, M., & Niyigena, J. P. An

- effective phishing detection model based on character level convolutional neural network from URL. *Electronics*, 9(9), 1514. 2020
- [9] Zhang, Y., Hong, J. I., & Cranor, L. F. Cantina: a content-based approach to detecting phishing web sites. In Proceedings of the 16th international conference on World Wide Web (pp. 639–648). 2007
- [10] Rao, R. S., Vaishnavi, T., & Pais, A. R. CatchPhish: detection of phishing websites by inspecting URLs. *Journal of Ambient Intelligence and Humanized Computing*, 11(2), 813–825. 2020
- [11] Korkmaz, M., Sahingoz, O. K., & Diri, B. Detection of phishing websites by using machine learning-based URL analysis. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1–7). IEEE. 2020
- [12] Rao, R. S., & Pais, A. R. Detection of phishing websites using an efficient feature-based machine learning framework. *Neural Computing and Applications*, 31(8), 3851–3873. 2019
- [13] Dangwal, S., & Moldovan, A. N. Feature Selection for Machine Learning-Based Phishing Websites Detection. In 2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA) (pp. 1–6). IEEE. 2021
- [14] Liu, C., Wang, L., Lang, B., & Zhou, Y. Finding effective classifier for malicious URL detection. In Proceedings of the 2018 2nd International Conference on Management Engineering, Software Engineering and Service Sciences (pp. 240–244). 2018
- [15] Sahingoz, O. K., Buber, E., Demir, O., & Diri, B. Machine learning based phishing detection from URLs. *Expert Systems with Applications*,

117, 345–357. 2019

[16] statista,

<https://www.statista.com/statistics/265677/number-of-internet-top-level-domains-worldwide/>

[17] Kaggle,

<https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset>

[18] Yang, Y. Research and realization of internet public opinion analysis based on improved TF-IDF algorithm. In 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES) (pp. 80–83). IEEE. 2017

[19] Azhagusundari, B., and Antony Selvadoss Thanamani. "Feature selection based on information gain." International Journal of Innovative Technology and Exploring Engineering (IJITEE) 2.2: 18–21. 2013

Abstract

Machine Learning-Based Malicious URL Detection Using IDF and String Features

Kim, Ae Ri

Dept. of Information Security

Graduate School of
Hanyang University

Phishing attacks are steadily increasing every year, and the scale of damage is also increasing accordingly. Phishing inflicts enormous damage not only to individuals but also to businesses, and the form of phishing attacks is evolving day by day. Rapid detection of malicious URLs is important to minimize damages against the increase in phishing.

In this paper, we aim to quickly detect malicious URLs using only features extracted from URLs themselves, excluding third-party services with relatively long response times. Attackers tend to lengthen the length of URLs to hide malicious URLs, and sometimes lengthen the path part. In this study, it was assumed that the path patterns of malicious URLs and normal URLs were different, and IDF-applied feature was used. As a result of ranking feature importance through information gain, the IDF feature was ranked in the top 5 among 28 features, showing a high importance of 15.94%.

As a result of machine learning detection of malicious URLs using string-based features and IDF features extracted from URLs themselves, when using the RF classifier, accuracy was 92.66%, F1-score 92.65%, and AUC 92.66%, which was the best classification among the three classifiers. showed performance.

감사의 글

석사 생활에서 2번째 겨울을 맞고 있는 지금, 졸업을 앞두고 있다는 것이 실감나지 않습니다. 연구실 생활을 잘할 수 있을지 걱정하던 게 엊그제 같은데 학위논문을 이렇게 마무리하니 감회가 새롭습니다. 석사생활을 잘 보낼 수 있었던 것은 좋은 분들과 함께 할 수 있었기 때문입니다. 제 석사생활을 SSL에서 보내게 되어 또 그로 인해 좋은 분들을 만나 뵙게 되어 감사하다는 말씀 전합니다.

우선 좋은 방향으로 지도해주신 임을규 교수님께 진심으로 감사드립니다. 교수님께서 여러 경험들을 겪을 수 있게 환경을 만들어주시고 고민에 대해 늘 명쾌한 해답을 주신 덕에 어려움을 잘 헤쳐 나갈 수 있었습니다. 또 연구의 주안점을 잘 잡을 수 있게 지도해주셔서 학위논문을 잘 마무리할 수 있게 되었습니다. 교수님의 큰 가르침에 따라 끊임없이 성장하는 사람이 될 수 있도록 노력하겠습니다. 그동안 지도해주셔서 감사합니다.

바쁘신 와중에 제 학위 논문의 심사위원을 맡아주시고 좋은 조언을 해주신 박희진 교수님, 채동규 교수님께도 감사 말씀 올립니다.

연구실을 이끌어주신 최두섭 선배님께 감사합니다. 연구실 선배로서 석사 생활에 필요한 정보들을 공유해주시고 고민에 대해 아낌없이 조언해주셔서 많은 도움을 받았습니다. 또 폐적한 연구실 환경을 조성해주신 덕에 좋은 환경에서 연구를 진행할 수 있었습니다. 감사합니다.

저에게 응원과 함께 밝은 에너지를 주시는 안드리 언니에게 감사합니다. 1년 동안 제주도 워크샵부터 수업까지 여러 활동을 같이 해왔는데 짧은 영어 실력 탓에 다양한 대화를 많이 나누지 못한 것이 참 아쉽습니다. 한국에 있는 동안 좋은 연구뿐만 아니라 행복한 추억들도 많이 쌓고 좋은 사람들도 많이 사귀면 좋겠습니다. 제게 도움을 주기 위해 늘 신경써주셔서 감사합니다.

석사 2년을 함께 걸어온 든든한 동기 박영진에게 감사합니다. 석사 생활동안 여러 일들을 같이 겪으면서 희로애락을 공유했는데 나와 같은 길을 걷는 동반자가 있어 덜 힘들고 더 기뻤습니다. 나보다 어린 동생이지만 내 멘탈을 케어해주고 잘 버틸 수 있

게 도와주어 감사합니다. 그동안 너무 고생했고 앞으로도 잘 해쳐 나가보자.

착실한 후배 김용준에게 감사합니다. 연구 분야가 달라서 학문적으로 도움을 주지 못해 아쉬움이 남습니다. 그래도 1년 동안 재미있는 연구실 생활을 함께 할 수 있어 좋았고 가끔 생기는 고민들을 해결해주어 감사합니다. 남은 1년 동안 원하는 결과를 많이 이루어서 졸업하길 응원하겠습니다.

유쾌한 후배 과리둔에게 감사합니다. 마지막 학기를 앞두고 만난 것이 참 아쉽습니다. 비교적 같이 한 활동은 적었지만 재미있었습니다. 앞으로 남은 3학기동안 많은 것들을 배우면서 다른 선후배들과 좋은 추억을 쌓는 알찬 석사 생활을 보내길 응원하겠습니다.

마지막으로 항상 뒤에서 응원과 격려를 아끼지 않았던 아버지, 어머니, 동생, 친구들에게 감사하다고 전하고 싶습니다.

2년 동안의 석사생활은 새로운 배움의 연속이었습니다. 여전히 배울게 많지만 석사 생활을 통해 많은 것들을 배울 수 있어 전보다 성장할 수 있었습니다. 모두가 건강하고 행복한 일들만 가득하길 바라며 본 논문을 마치겠습니다. 모두 감사합니다.

2023년 2월

김애리

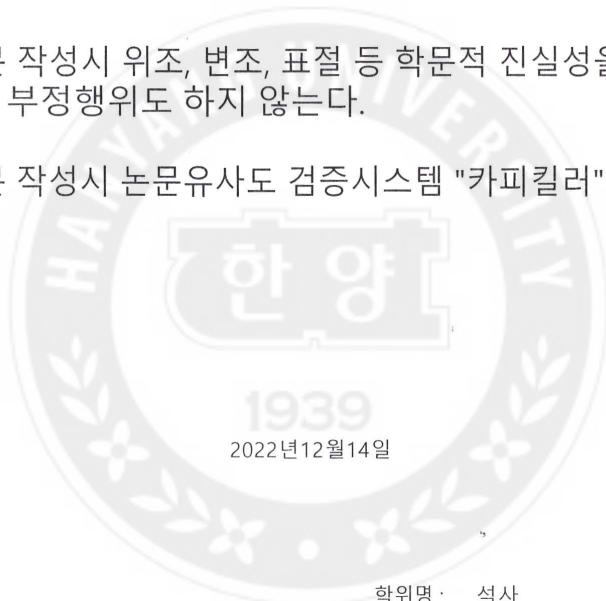
연구 윤리 서약서

본인은 한양대학교 대학원생으로서 이 학위논문 작성 과정에서 다음과 같이 연구 윤리의 기본 원칙을 준수하였음을 서약합니다.

첫째, 지도교수의 지도를 받아 정직하고 엄정한 연구를 수행하여 학위논문을 작성한다.

둘째, 논문 작성시 위조, 변조, 표절 등 학문적 진실성을 훼손하는 어떤 연구 부정행위도 하지 않는다.

셋째, 논문 작성시 논문유사도 검증시스템 "카피킬러" 등을 거쳐야 한다.



학위명 : 석사

학과 : 정보보안학과

지도교수 : 임을규

성명 : 김애리

A handwritten signature in black ink, which appears to be '김애리' (Kim Ae-ri).

한 양 대 학 교 대 학 원 장 귀 하

Declaration of Ethical Conduct in Research

I, as a graduate student of Hanyang University, hereby declare that I have abided by the following Code of Research Ethics while writing this dissertation thesis, during my degree program.

"First, I have strived to be honest in my conduct, to produce valid and reliable research conforming with the guidance of my thesis supervisor, and I affirm that my thesis contains honest, fair and reasonable conclusions based on my own careful research under the guidance of my thesis supervisor.

Second, I have not committed any acts that may discredit or damage the credibility of my research. These include, but are not limited to : falsification, distortion of research findings or plagiarism.

Third, I need to go through with Copykiller Program(Internet-based Plagiarism-prevention service) before submitting a thesis."

DECEMBER 14, 2022

Degree : Master

Department : DEPARTMENT OF INFORMATION SECURITY

Thesis Supervisor : Eul Gyu Im

Name : KIM AERI

[Signature]