Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.6, Dec. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.6.881

악성 URL 탐지를 위한 URL Lexical Feature 기반의 DL-ML Fusion Hybrid 모델

김 대 엽†‡
국방과학연구소 (선임연구원)

DL-ML Fusion Hybrid Model for Malicious Web Site URL Detection Based on URL Lexical Features

Dae-yeob Kim†‡
Agency for Defense Development (Senior Researcher)

요 약

최근에는 인공지능을 활용하여 악성 URL을 탐지하는 다양한 연구가 진행되고 있으며, 대부분의 연구 결과에서 높은 탐지 성능을 보였다. 그러나 고전 머신러닝을 활용하는 경우 feature를 분석하고 선별해야 하는 추가 비용이 발생하며, 데이터 분석가의 역량에 따라 탐지 성능이 결정되는 이슈가 있다. 본 논문에서는 이러한 이슈를 해결하기 위해 URL lexical feature를 자동으로 추출하는 딥러닝 모델의 일부가 고전 머신러닝 모델에 결합된 형태인 DL-ML Fusion Hybrid 모델을 제안한다. 제안한 모델로 직접 수집한 총 6만 개의 악성과 정상 URL을 학습한 결과 탐지 성능이 최대 23.98%p 향상되었을 뿐만 아니라, 자동화된 feature engineering을 통해 효율적인 기계학습이 가능하였다.

ABSTRACT

Recently, various studies on malicious URL detection using artificial intelligence have been conducted, and most of the research have shown great detection performance. However, not only does classical machine learning require a process of analyzing features, but the detection performance of a trained model also depends on the data analyst's ability. In this paper, we propose a DL-ML Fusion Hybrid Model for malicious web site URL detection based on URL lexical features. the propose model combines the automatic feature extraction layer of deep learning and classical machine learning to improve the feature engineering issue. 60,000 malicious and normal URLs were collected for the experiment and the results showed 23.98%p performance improvement in maximum. In addition, it was possible to train a model in an efficient way with the automation of feature engineering.

Keywords: Malicious URL Detection, Phishing Detection, Deep Learning, Machine Learning

I. 서 론

악성 웹 사이트는 가장 보편적으로 이용되는 웹 서비스 환경에서의 대표적인 사회공학적 사이버 위협으로, 부주의한 사용자의 심리를 악용한다는 점에서 이를 활용한 공격은 인지하기가 어렵다. 또한 다른 공격에 비해 수행하는 방식이 비교적 단순하며, 악성코드 유포 및 계정 탈취와 같은 효과적인 공격이 가능하다[11][13][4]. 따라서 악성 웹 사이트는 사이버 공격에 여전히 많이 활용되고 있으며, 이로 인한

Received(07. 19. 2023), Modified(10. 11. 2023), Accepted(10. 11. 2023)
† 주저자, yeob@add.re.kr
‡ 교신저자, yeob@add.re.kr(Corresponding author)

피해는 계속해서 증가하고 있다[2]. AWPG (Anti-Phishing Working Group)의 통계에 따르면, 22년 1분기에 악성 웹 사이트를 통한 피싱 공격 건수는 300,000건 이상으로 사상 최고치를 기록하였으며, 2년 전보다 3배 이상 증가하였다[14].

기존에는 이러한 악성 웹 사이트를 블랙리스트 방식으로 탐지하고 있지만, 대량의 리스트를 관리해야 하는 어려움이 있으며, 리스트에 등록되지 않은 악성 URL은 탐지하지 못하는 한계가 있다[4][2][8]. 따라서 인공지능을 활용한 탐지 방법이 제안되었는데, 인공지능은 통계를 기반으로 feature를 학습하여 악성 웹 사이트를 탐지하는 방식이기 때문에 이전에 관측되지 않은 악성 웹 사이트도 탐지할 수 있다.

최근에는 빠른 탐지 속도와 높은 탐지 성능을 가진 URL lexical feature 기반의 인공지능 모델이 주요 연구 대상이 되고 있다. 탐지 속도가 빠르면 그만큼 피해 규모를 줄일 수 있을 뿐만 아니라, real-time 시스템과 같은 제한적인 환경에서도 활용이 가능한 장점이 있다[11][8][3]. 따라서, 이러한 URL lexical feature를 고전 머신러닝 또는 딥러닝으로 학습하여 악성 URL을 탐지하는 다양한 인공지능 모델이 연구되고 있다[1][2][4].

고전 머신러닝 기반의 탐지 모델은 저사양의 컴퓨팅 파워로도 학습이 가능하며 데이터의 양이 많지 않아도 모델링이 가능하다는 장점이 있다. 그러나 수동으로 직접 feature를 분석하고 선별해야 하며, 데이터 분석 역량에 따라 탐지 성능이 직접적으로 결정되는 단점이 있다. 반면, 딥러닝의 경우 전문가가 분석하여 선별한 feature보다 더 정교한 feature를 자동으로 추출해주는 장점이 있으며 이는 성능 향상에도 효과적이다[13][11]. 이러한 장단점을 토대로 본 논문에서의 접근 방법은 딥러닝 신경망의 일부 층(layer)을 활용하여 성능 향상은 물론 feature engineering 과정을 자동화하면서 동시에 데이터 분석가의 역량에 대한 의존성 문제를 해결하는 것이다.

이러한 접근 방법으로 본 논문에서는 고전 머신러닝의 단점을 해결하기 위해 URL lexical feature를 학습한 딥러닝 모델 일부를 고전 머신러닝에 결합한 DL-ML fusion hybrid 모델을 제안한다. 제안한 모델을 평가하기 위해 웹에서 직접 수집한 6만 개의 URL 데이터(악성, 정상 5:5 비율)로 실험을 수행하였으며, 성능 평가 메트릭(accuracy, precision, recall, f1)과 ROC curve로 성능을 검증하였다.

고전 머신러닝 알고리즘으로 학습된 단일 모델과 본 논문에서 제안한 모델을 비교했을 때, 탐지 성능이 최대 23.98%p 향상되었으며, 성능 평가 메트릭의 평균은 17.83%p 향상되었다. 또한 제안한 모델이 전체 단일 모델 대비 가장 좋은 탐지 성능을 보였는데, 성능이 가장 높게 측정된 모델의 precision은 93.24%로 측정되었으며, 성능 평가 메트릭의 평균은 93.1%로 측정되었다.

실험을 통해 제안한 모델이 성능 향상에 효과적임을 검증하였으며, feature engineering에 의존하는 고전 머신러닝의 단점이 개선됨을 보였다.

본 논문의 구성은 총 6장으로 구성되어 있으며 순서는 다음과 같다. 1장에서는 서론, 2장에서는 관련 연구를 통해 기존 연구를 분석하고, 3장에서는 기술적 배경지식을 다룬다. 4장과 5장에서는 본 논문의 제안 방법에 대한 설명과 실험 결과를 분석하고, 6장에서 향후 연구 계획과 결론으로 끝을 맺는다.

II. 관련 연구

기존에 제안된 고전 머신러닝 기반의 악성 URL 탐지연구는 대부분 feature engineering에 대한 비용이 발생한다는 점에서 본 논문과는 차이가 있다.

Darling, M. 등[1]은 기존의 고전 머신러닝 기반 모델을 경량화하여 실시간 악성 URL 탐지 모델을 제안하였다. 저자는 모델을 경량화하기 위해 추출 속도가 빠른 URL lexical feature만을 활용하였다. 이러한 과정에서 탐지 속도와 같은 비용적인 부분을 개선하였지만, 본 논문과 다르게 feature 선별 및 추출에 대한 비용이 소요된다.

Gupta, B.B. 등[2]은 다수의 feature를 사용하여 높은 처리량을 요구하는 기존 탐지 모델의 한계를 지적하였다. 따라서 저자는 기존의 URL lexical feature를 최적화하기 위해 feature 중요도를 K best 등의 기법으로 분석하여 9개의 주요 특징을 도출하였다. 이러한 과정에서 feature의 차원은 축소되어 처리량에 대한 비용은 줄였지만, feature engineering에 대한 추가 비용이 발생한다는 점에서 본 논문과 차이가 있다.

Hong, J. 등[3]은 Random forest, Adaboost 등의 고전 머신러닝 알고리즘과 1D CNN, LSTM 등의 딥러닝 알고리즘을 모두 활용하였지만, 각각의 알고리즘으로 학습된 단일 모델만 고려하여 연구를

수행하였다. 저자는 기존에 연구된 18가지 URL lexical feature에 블랙리스트를 추가한 총 19차원의 feature를 활용하였다. 저자가 제안한 모델을 학습하기 위해서는 feature 선별 및 추출에 대한 비용이 추가로 요구되는 이슈가 있다.

딥러닝 기반 악성 URL 탐지모델의 경우 고전 머신러닝과 다르게 더 정교한 feature를 자동으로 추출할 수 있으며 성능 또한 더 뛰어나다.

Le, H. 등[4]은 URL에 유니크한 단어가 많은 경우 단어 벡터를 학습할 때 발생하는 메모리 제약과 단어 벡터를 효과적으로 추출하지 못하는 기존의 딥러닝 모델을 지적하였다. 저자는 CNN(Convolutional Neural Network)을 활용하여 문자 및 단어 수준의 임베딩 벡터(embedding vector)뿐만 아니라 각 단어에 대해서도 문자 수준의 임베딩 벡터를 추출하는 방식으로 새로운 단어도 탐지할 수 있는 딥러닝 모델을 연구하였다. 저자가 제안한 모델은 99%의 높은 정확도를 달성하였다.

Tajaddodianfar, F. 등[5]은 문자와 단어 수준의 특징 벡터를 학습한다는 점에서 [4]의 연구와 유사하지만, FastText 모델을 활용하여 단어의 문맥적 의미를 고려했다는 점에서 차이가 있다. 또한 저자는 다양한 필터 사이즈로 설정된 convolution 층 여러 개를 병렬로 구성하는 FEB(Feature Extraction Block)을 추가로 활용하여 하나는 문자 수준의 특징을, 다른 하나는 단어 수준의 특징을 추출하는 방법을 제안하였다.

III. URL Lexical Feature를 활용한 인공지능 기반 악성 URL 탐지

3.1 URL(Uniform Resource Locator) 구조

URL(Uniform Resource Locator)은 인터넷상에서 자원이 어디에 있는지를 알려주기 위한 규약으로 흔히 웹 주소 또는 인터넷 주소라고 불린다. 표준 구조는 RFC 1738에 정의되어 있으며, Fig.1. 과같이 프로토콜(protocol), 서브 도메인(subdomain), 하위 도메인(sld), 최상위 도메인(tld), 디렉터리(directory), 파일 이름(file name), 쿼리 스트링(query string) 등 7가지 부분으로 구성된다[11][13].

프로토콜은 웹 브라우저가 웹 서버와 통신하는 방법을 정의한다. 도메인 이름은 인터넷상에서 웹 사이트를 유일하게 식별할 수 있는 식별자로, 최상위 도

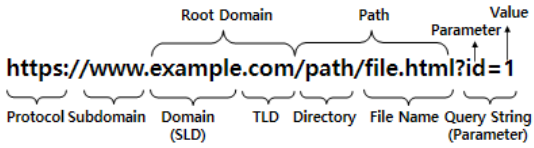

Fig. 1. The structure of URL(Uniform Resource locator)

메인과 하위 도메인으로 구성되어 있다. 서브 도메인은 루트 도메인(root domain)의 보조 도메인으로 웹 사이트 내에서 별도의 웹 사이트 섹션(section)을 만들어 운영할 때 독립된 도메인으로 활용할 수 있는 식별자이다. 경로(path)는 현재 요청하는 파일이 웹 서버 내에서 어디에 위치하는지 디렉터리 경로를 나타낸다. 쿼리 스트링은 특수문자 '?'로 시작하며 사용자가 웹 서버에 요청할 때 같이 전달하는 변수(parameter) 이름과 값(value)으로 구성되어 있다. 여러 개의 변수와 값을 전달할 때는 구분자 '&'를 붙여 전달한다.

3.2 악성 URL의 Lexical Feature

공격자가 악성 URL을 정상 URL처럼 보이도록 위조하는 과정에서 악성 URL의 외관은 정상 URL과 유사해지는데, 문자 구성이나 패턴 등에서는 차이가 발생하게 된다. 악성 URL을 위조하는 방식은 Table 1.과 같이 6가지 정도로 구분되는데, 대부분은 URL 구성 요소(path, subdomain 등)의 일부 문자를 바꾸거나 정상 URL에 자주 사용되는 키워드를 삽입하는 방식이다[7][8][9][10].

Table 1. Types of URL forgery

Type	Description
Type 1	typo-squatting(miss spelling)
Type 2	insertion of sensitive and reliable word
Type 3	insertion of another domain
Type 4	inclusion of special character, ip address, numbers
Type 5	short URL conversion
Type 6	inclusion of malicious executable file name with extension

이처럼 악성 URL을 위조하는 과정에서 발생하는 차이는 기계학습 feature로 활용되는데, 본 논문에서는 이러한 차이점을 고려하여 아래와 같이 크게 4가지의 URL lexical feature Group으로 구분하였다[8].

Feature Group 1: URL의 전체 또는 구성 요소의 길이: 악성 URL은 정상 URL처럼 보이도록 위장하기 위해 잘 알려진 브랜드 이름, 정상 URL의 도메인 등을 subdomain에 삽입하는 경우가 많은데 이처럼 문자나 단어를 추가하는 과정에서 길이가 길어지는 경향을 보이게 된다[3][15]. 따라서 통계적으로 악성 URL의 전체 또는 구성 요소의 길이가 정상 URL보다 긴 경우가 많다.

Feature Group 2: URL에 포함된 특정 문자 개수: 통계에 따르면 악성 URL은 !, #, @, % 등의 특수문자를 포함하는 경향이 있는데, 이는 공격자가 정상 URL 중간에 특수문자를 넣어 위조하거나 일부 특수문자의 기능을 악용하기 때문이다[2][3][15]. 예를 들어, 'www.facebook.com'을 'www.face.book.com'으로 위조하거나, 이전 문자열을 무시하는 특수문자 '@' 또는 '-'를 악용하여 악성 URL 접속을 유도한다. http://normal.com@malware.com을 클릭하면 normal.com은 무시되어 malware.com으로 접속하게 된다.

Feature Group 3: URL에 특정 단어 포함 여부: 악성 URL은 보통 특정 단어를 포함하는 경우가 많다. 예를 들어, 'login', 'admin', 'confirm' 등 정상 URL에 자주 사용되는 키워드나 계정 탈취에 관련된 단어가 포함되며, 악성코드를 유포하는 URL의 경우 '.exe' 등과 같은 실행파일의 확장자가 포함되는 경향을 보인다[3][15]. 특히 악성 웹 사이트는 주로 무료 호스팅 서비스를 활용하기 때문에 도메인에 '000webhost' 등과 같은 무료 호스팅 서비스의 도메인을 포함하는 경우가 많다.

Feature Group 4: URL의 숫자 구성 비율: 악성 URL은 숫자를 포함하는 비율이 정상 URL에 비해 높은 경향이 있다. 이는 악성 웹 사이트가 도메인 주소 대신 IP 주소를 사용하거나 악성 URL이 정상 URL처럼 보이기 위해 정상 도메인의 일부 알파벳을 숫자로 위조하는 과정에서 숫자 구성 비율이 높아지기 때문이다. 예를 들어, 'facebook.com'을 'faceb00k.com'으로, 'netflix.com'을 'netfl1x.com'으로 위조할 수 있다[15].

3.3 URL Lexical Feature를 활용한 악성 URL 탐지모델 기계학습

악성 URL을 탐지하는 것은 주어진 한 개의 URL에 대해서 악성 또는 정상으로 분류하는 이진 분류 문제에 해당한다. 따라서 URL 데이터 셋 T는 T개의 URL 집합 $\mathrm{T}=\{(u_1,y_1),...,(u_T,y_T)\}$로 표현할 수 있다. $t=1,...T$ 일때, u_t는 t번째에 해당하는 한 개의 URL을 의미한다. $y_t\in\{1,-1\}$는 라벨을 의미하며 $y_t=1$는 악성 URL을, $y_t=-1$는 정상 URL을 의미한다.

주어진 한 개의 URL u_t를 탐지하기 위해서는 먼저 $u_t\rightarrow X_t$는 $X_t\in\mathbb{R}^n$일 때 u_t를 n차원의 특징 벡터 X_t로 변환한다. 그다음 특징 벡터 X_t를 악성 또는 정상 클래스로 탐지하는 함수 $f:\mathbb{R}^n\rightarrow\mathbb{R}$를 학습한다. 학습 과정에서 오 분류된 총 URL 개수 $(\sum_{t=1}^{T}I_{\hat{y}_t}\neq y_t)$가 최소화될 때까지 손실 함수를 통해서 학습이 진행된다. 학습한 함수의 탐지 결과는 $\hat{y}_t=sign(f(X_t))$로 표현될 수 있으며 특징 벡터 X_t가 악성 또는 정상 클래스에 속할 예측 점수를 계산한다[4][8].

IV. 제안 방법

4.1 악성 URL 탐지를 위한 URL Lexical Feature 기반의 DL-ML Fusion Hybrid 모델

본 논문에서는 딥러닝과 고전 머신러닝이 결합된 형태로 URL lexical feature를 학습하여 악성 URL을 탐지하는 DL-ML fusion hybrid 모델을 제안한다.

고전 머신러닝의 단점을 개선하기 위해 Fig. 2.와 같이 딥러닝 모델의 일부 층을 feature 추출 과정에 활용하여 feature engineering 과정을 자동화하면서 동시에 feature에 대한 의존성을 해결하려는 시도로 이해할 수 있다.

Fig. 2.는 DL-ML fusion hybrid 모델의 구성 과정을 보여주며, FEB1+XGB 모델을 예시로 활용하였다. DL-ML fusion hybrid 모델을 구성하는 과정은 크게 세 단계로 구성된다.

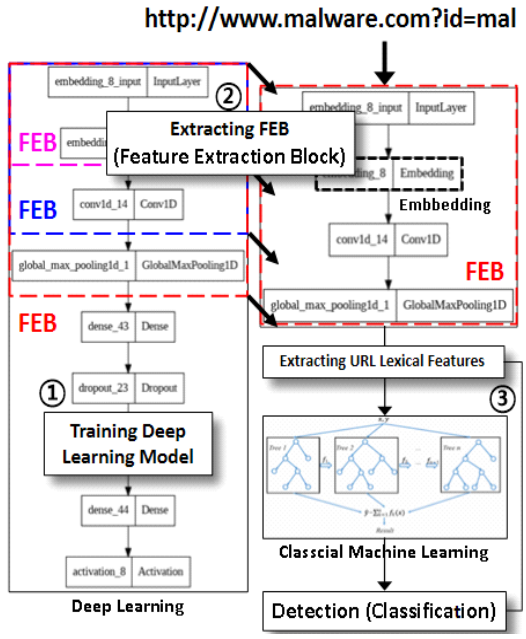

Fig. 2. The fusion process of deep learning and classical machine learning models (FEB1+XGB)

첫 번째 단계에서는 딥러닝으로 URL 데이터를 학습하고 성능을 평가하는 과정을 반복하며 최적의 성능을 내는 탐지모델을 찾는다. 이미 모델이 존재하는 경우, 이를 재활용하여 첫 번째 과정을 생략할 수도 있다.

두 번째 단계에서는 딥러닝 모델의 Input 층을 시작으로 한 층씩 추가하면서 하나의 FEB (Feature Extraction Block)를 구성하는 방식으로 여러 개의 독립된 FEB를 추출한다.

세 번째 단계에서는 추출된 FEB로 URL lexical feature를 추출하여 고전 머신러닝으로 기계학습 및 성능 평가를 수행한다. 세 번째 과정을 반복하면서 최고 성능을 내는 FEB과 고전 머신러닝 모델의 조합을 찾아 최종적으로 DL-ML fusion hybrid 모델을 구성한다.

4.2 DL-ML Fusion Hybrid 모델의 URL Lexical Feature 기계학습 및 악성 URL 탐지 과정

한 개의 URL u_t는 Fig. 3.과 같이 전처리 과정에서 문자 단위로 토큰화된 후 정수로 변환된다. 그다음 길이에 따라 절삭 되거나 패딩 처리되어 L개의 문자로 구성된 문자 집합 $X=\{x_1,...x_L\}$로 변환된다. 변환된 문자 집합은 FEB의 임베딩 층에서 $X_t \in \mathbb{R}^{L\times k}$인 벡터 행렬 X_t로 변환된다. Fig. 2.와 같이 임베딩 층(embedding layer)에 차례대로 연결된 1D Convolution 층과 Global Max Pooling 층은 앞서 변환된 임베딩 벡터 행렬로부터 URL lexical feature를 자동으로 추출한다.

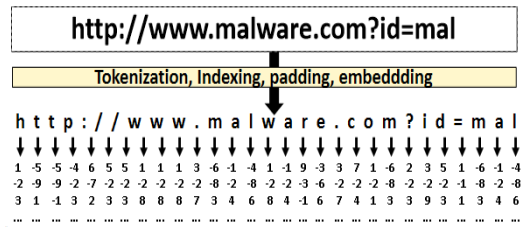

Fig. 3. Character level embedding of a single URL

1D Convolution 연산은 미리 정의된 window로부터 추출된 URL lexical feature 벡터에 dot 연산이 적용된 것을 의미하며 수식 (1)과 같이 표현된다. Global Max Pooling 연산은 추출된 벡터에서 값이 가장 큰 스칼라를 추출하는 연산으로 차원을 감소시키며 계산 복잡도를 줄이는 과정이다.

$$c_i^l = f(b_i + \sum_{h=1}^{H} w_h x_{i+h-1}) \quad (1)$$

1D Convolution 층은 $X_t \in \mathbb{R}^{L\times k}$인 벡터 행렬 X_t에 H길이의 커널w로 1D Convolution 연산을 수행한다. 그다음 비선형 활성화 함수 f를 적용하여 출력값 c_i를 생성한다. 여기서 각각의 세그먼트(segment)는 미리 정해진 stride 값만큼 분리되며, b_i는 bias를 의미한다. 최종적으로 출력값 c_i는 모두 결합(concatenation) 되어 URL lexical feature C를 생성하며, 수식 (2)와 같이 표현된다.

$$C=[c_1,c_2,...,c_{L-H+1}] \quad (2)$$

1D convolution 연산으로 추출된 feature C에 Global Max Pooling 연산을 수행하여 1차원 벡터로 축소하면서 동시에 중요한 feature를 추출한다. 이렇게 자동으로 추출된 feature를 고전 머신러닝으로 학습하여 악성 URL을 탐지하는 모델을 생

성한다. Fig. 2.의 예시에서는 고전 머신러닝 알고리즘으로 XGB(eXtreme Gradient Boosting)를 활용하였는데, XGB는 CART(Classification And Regression Trees) 기반의 모델로 수식 (3)과 같이 표현할 수 있으며, additive 방식으로 트리를 학습시킨다. K개의 트리가 있고 각각의 모델 f_k의 예측값을 더해 최종 예측값 y'_i를 계산한다. y'_i는 데이터 x_i의 예측값이며, K는 사용된 CART의 개수, f는 CART의 모델들이다.

$$y'_i = \sum_{k=1}^{K} f_k(x_i), f_k \in (F) \tag{3}$$

$$obj = \sum_{i=1}^{n} l(y_i, y'_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

각 CART 모델을 학습시키기 위한 목적함수는 수식 (4)와 같다. 여기에서 좌측 항은 실제 값과 예측값과의 차이를 나타내는 training loss이고, 우측 항은 트리 모델의 복잡도를 조절하여 과적합(over fitting)을 방지하는 정규화(regularization) 항이다.

생성된 각각의 트리 f_k는 앞서 추출된 feature를 Input x_i로 입력받아 Output으로 예측값을 도출하는데, 각각의 트리로부터 받은 이러한 예측값들을 합쳐 최종 예측값으로 악성 URL 탐지 유무를 결정한다.

V. 실험 결과

5.1 URL 데이터 수집 및 실험 방법

본 논문에서 제안하는 DL-ML fusion hybrid 모델의 성능을 검증하기 위해 활용된 URL 데이터는 Table 2.와 같다.

22년 6월부터 1년간 정상 URL 30,000개와 악성 URL 30,000개를 실제 웹 환경에서 직접 수집하였다. 정상 URL은 트래픽 순위에 따라 정렬되는 Alexa top site 리스트에서 랜덤으로 크롤링하여 수집하였다. 악성 URL은 실제 사용자들로부터 신고된 악성 사이트를 검사하고 관리하는 PhishTank와 OpenPhish에서 수집하였다.

제안한 모델의 성능을 검증하기 위해서는 먼저 딥러닝과 고전 머신러닝이 결합 되어 있다는 점을 고려해야 한다. 따라서 실험 과정은 먼저 단일 딥러닝 모델과 단일 고전 머신러닝 모델의 성능을 각각 측정한 후, 제안한 모델로 학습했을 때의 성능과 비교하였다. 또한 본 논문에서 제안한 방법이 일부 모델에서만 제한적으로 유효하지 않고 일반적으로도 유효함을 보이기 위해 다양한 딥러닝 및 고전 머신러닝 알고리즘을 활용하여 다수의 실험 결과를 도출하였다.

Table 2. Malicious and normal URL data set

	Normal URLs	Malicious URLs	Total
Training	24,000	24,000	48,000
Testing	6,000	6,000	12,000
Total	30,000	30,000	60,000

세 가지 모델 모두 Table 2.의 데이터를 학습하였으며, 8:2 hold-out 방식으로 검증한 결과를 성능 평가 메트릭(Precision, Recall, Accuracy, f1-score)과 ROC curve로 비교하였다.

제안한 모델의 성능을 평가하는 과정에서 단일 고전 머신러닝 모델을 주 비교 대상으로 하였는데, 본 논문에서의 문제 해결 목표는 딥러닝 모델의 일부를 활용하여 고전 머신러닝의 단점과 성능을 개선하는 것이기 때문이다.

5.2 고전 머신러닝 기반 악성 URL 탐지 성능 평가

고전 머신러닝 모델의 성능을 평가하기 위해 기존 연구에서 많이 활용되고 있는 RF(Random Forest), LGB(Light GBM), XGB(XGBoost), CAT(CatBoost), Ada(AdaBoost), LR(Logistic Regression), DT(Decision Tree) 등 총 8가지 알고리즘을 활용하였다.

feature는 기존 연구에서 주로 많이 활용되는 총 37개의 URL lexical feature를 사용하였으며, 모두 본 논문에서 구분한 4가지 feature group에 해당한다[1][2][3][6][11][12]. 하이퍼 파라미터는 대부분 default 값으로 설정하였다.

각 모델의 성능은 Fig. 4.와 같다. CAT 모델의 성능 평가 메트릭 평균이 88.11%로 가장 높은 탐지 성능을 보였으며, LR 모델의 경우에는 75.11%로 가장 저조한 탐지 성능을 보였다.

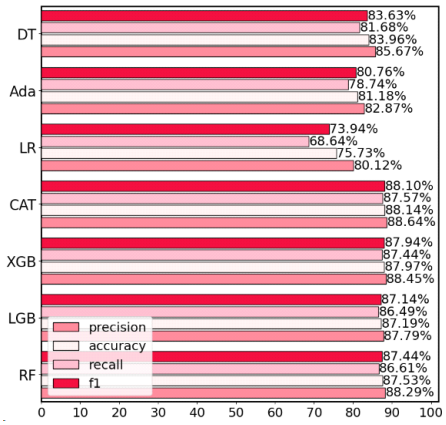

Fig. 4. The performance of classical machine learning models

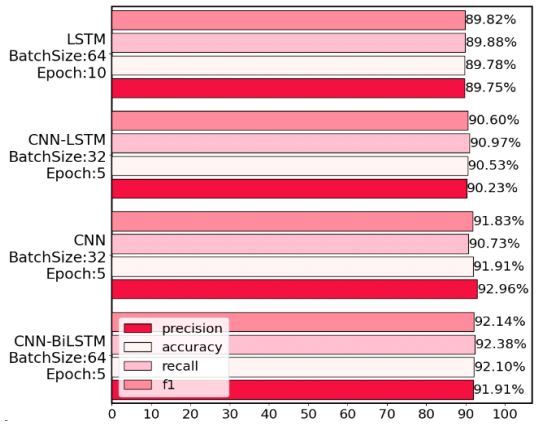

Fig. 5. The performance of deep learning models

5.3 딥러닝 기반 악성 URL 탐지 성능 실험

딥러닝 모델의 성능을 평가하기 위해 텍스트 분류에 좋은 성능을 보이는 LSTM, CNN, BiLSTM 등의 알고리즘을 활용하였으며[3], 문자 수준의 URL lexical feature를 학습하였다.

본 논문에서는 각 URL을 32차원으로 문자 임베딩을 수행하였으며, 패딩(padding) 길이 L은 전체 URL 데이터 셋 길이의 백분위수(percentile)를 계산하여 272로 설정하였다. 따라서 272자 이상의 URL은 272번째 문자에서 절삭 되고, 이하의 URL은 272자가 될 때까지 패딩처리 된다. 이렇게 전처리 된 각 URL u_t는 문자 임베딩을 통해 길이가 272인 32차원($L=272$, $k=32$)의 벡터 행렬 $u_t \rightarrow X_t \in \mathbb{R}^{L \times k}$로 변환된다.

각 딥러닝 모델의 층 구조와 하이퍼 파라미터는 기존 연구를 참고하였으며, 다양한 실험을 위해 32, 64, 128, 256의 batch size와 5, 10, 15, 20의 epoch를 조합하여 모든 경우의 수로 기계학습을 진행하였다. 딥러닝 모델의 성능 결과는 Fig. 5.와 같다.

CNN과 BiLSTM로 구성된 functional 모델이 64의 batch size와 5의 epoch로 학습을 진행했을 때, 성능 평가 메트릭 평균이 92.14%로 가장 높은 성능을 보였다. 반면 64의 batch size와 10의 epoch로 학습을 진행했을 때 가장 높은 성능을 보인 LSTM의 평균은 89.81%로 다른 딥러닝 알고리즘에 비해 가장 저조한 성능을 보였다.

5.4 DL-ML Fusion Hybrid 기반 악성 URL 탐지 성능 실험

제안한 DL-ML fusion hybrid 모델의 성능을 검증하기 위해 앞서 실험한 딥러닝 모델에서 Fig. 2.와 같이 층별로 구성할 수 있는 모든 조합의 FEB를 추출하였다. 그다음 각 고전 머신러닝 알고리즘과 결합할 수 있는 모든 경우의 수로 기계학습을 진행하였다.

앞서 실험한 단일 고전 머신러닝 모델을 기준으로 성능이 가장 높게 측정된 모델들의 조합은 Fig. 6.과 같으며, 각 FEB의 상세 구조는 Table 3.과 같다. 단일 모델과 비교했을 때 대부분의 모델 조합에서 성능 향상이 있었지만, 그중에서도 성능 향상이 가장 많았던 모델 조합은 FEB3+LR이다. 해당 모델 조합의 성능을 단일 LR 모델과 비교했을 때 recall의 증가 폭이 23.98%p(68.64%→92.62%)로 가장 높았으며, 성능 평가 메트릭 평균의 증가 폭도 17.83%p(74.61%→92.44%)로 가장 높았다.

전체 모델 중에서 탐지 성능이 가장 뛰어났던 모델은 FEB1+CAT이다. 특히 precision이 93.24%로 가장 높게 측정되었으며, 성능 평가 메트릭의 평균도 93.1%로 가장 높았다. 해당 모델 조합의 경우에도 성능 평가 메트릭의 평균을 기준으로 단일 고전 머신러닝 모델 CAT과 비교했을 때 4.99%p(88.11%→93.1%)만큼 성능이 향상되었으며, FEB1의 단일 딥러닝 모델인 'CNN-32-5'와 비교했을 때도 1.24%p(91.86%→93.1%)만큼 향상되었다. 따라서 본 논문에서 제안한 모델로 단일 고전 머신러닝 모델뿐만 아니라 단일 딥러닝 모델의 성능도 개선됨을 알 수 있었다.

각 실험 결과에 따른 ROC curve와 AUC를 Fig. 7., Fig. 8., Fig. 9.에 표현하였다. Fig. 7.은 성능 향상이 가장 많았던 FEB3+LR 모델 조합과 단일 LR 모델의 성능 차이를 보여준다. LR 모델의 경우 다른 고전 머신러닝 모델과 같은 feature를 학습했는데도 성능이 매우 저조하였지만, 단일 LR 모델에 FEB3을 결합하여 feature 분석 및 선별과정은 자동화하면서 탐지 성능을 개선할 수 있었다.

DL-ML fusion hybrid 모델과 모든 모델의 성능을 평가하기 위해 Fig. 8.에서는 단일 고전 머신러닝 모델과 성능을 비교하였으며, Fig. 9.에서는 단일 딥러닝 모델과 성능을 비교하였다. 각각의 비교 결과에서 모두 본 논문에서 제안한 FEB1+CAT 모델이 가장 높은 탐지 성능을 보였다.

이러한 실험 결과를 통해 비교적 탐지 성능이 뛰어난 딥러닝 모델일지라도 단일 모델로 사용하는 것보다 고전 머신러닝과 결합한 형태인 DL-ML fusion hybrid 모델로 활용했을 때 성능 향상은 물론 서로 다른 여러 개의 모델을 효율적으로 학습할 수 있었다.

결론적으로, 실험 결과를 통해 feature engineering보다 FEB로 추출된 feature가 악성 URL 탐지 성능을 높이는데 더 효과적임을 알 수 있었으며, feature 분석 및 선별과정을 자동화할 수 있어 효율적인 기계학습이 가능하였다.

Table 3. FEB(Feature Extraction Block) structure

FEB	Model	FEB Layer Structure
FEB1	CNN -32-5	InputLayer Embedding Conv1D GlobalMaxPooling1D
FEB2	CNN-BiLSTM -64-5	InputLayer Embedding Conv1D BatchNormalization Activation MaxPooling1D Bidirectional(LSTM)
FEB3	CNN-BiLSTM-64 -5	InputLayer Embedding Conv1D BatchNormalization Activation MaxPooling1D Bidirectional(LSTM) Dense BatchNormalization
FEB4	CNN-32-5	InputLayer Embedding Conv1D GlobalMaxPooling1D Dense Dropout Activation Dense

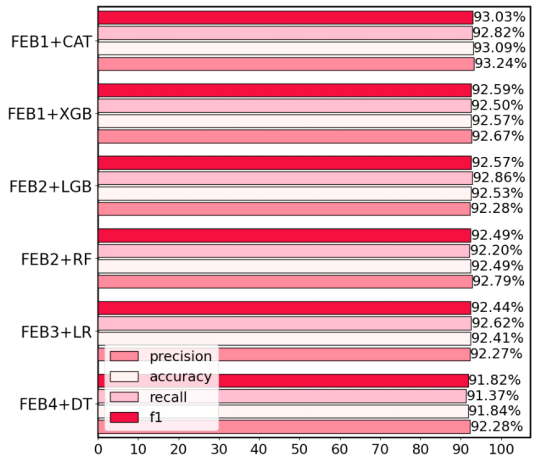

Fig. 6. The performance of DL-ML Fusion Hybrid Model

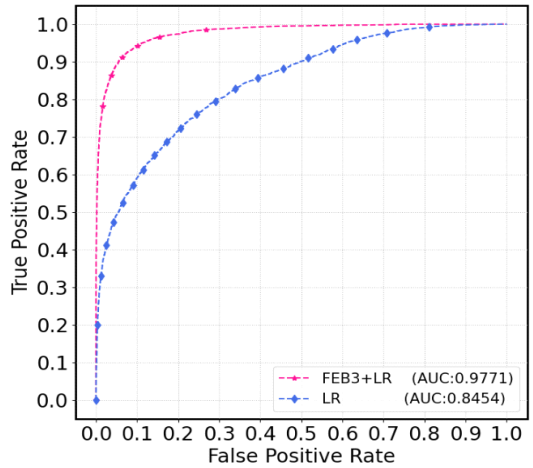

Fig. 7. ROC curve and AUC comparison between FEB3+LR and a single LR model trained on the test data set

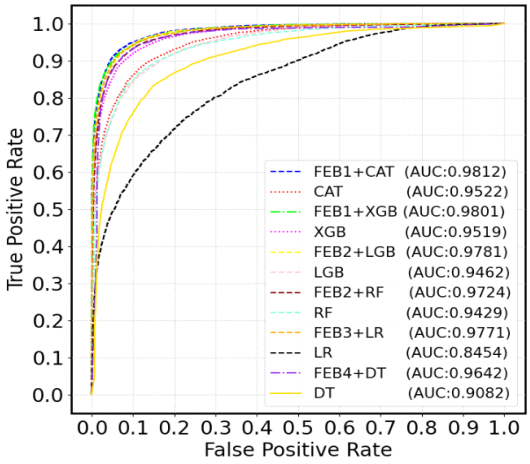

Fig. 8. ROC curve and AUC comparison between DL-ML Fusion Hybrid and every ML model trained on the test data set

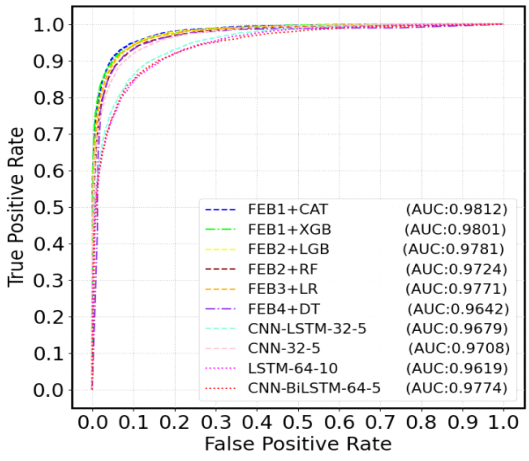

Fig. 9. ROC curve and AUC comparison between DL-ML Fusion Hybrid and every DL model trained on the test data set

VI. 결 론

본 논문에서는 딥러닝과 고전 머신러닝을 혼합하여 악성 URL을 효율적으로 학습하고 탐지할 수 있는 DL-ML fusion hybrid 모델을 제안하였다.

제안한 모델은 딥러닝 모델의 일부를 활용하여 문자 수준의 URL lexical feature를 자동으로 추출하고, 탐지 과정은 고전 머신러닝으로 수행한다. 따라서 탐지 성능을 높이면서 동시에 feature engineering에 대한 의존성을 개선한다.

제안한 모델의 성능을 검증하기 위해 직접 수집한 URL 60,000개를 대상으로 기계학습을 수행하였으며, 8:2 hold-out으로 검증한 결과를 성능 평가 메트릭과 ROC curve로 비교하였다.

고전 머신러닝에 딥러닝 모델의 일부를 혼합했을 때, 성능 평가 메트릭을 기준으로 탐지 성능이 최대 23.98%p 향상되었으며, 평균적으로는 최대 17.83%p 증가하였다. 또한 이처럼 혼합된 모델의 탐지 성능이 전체 모델 중에서도 가장 뛰어났는데, 최대 성능은 93.24%, 평균 성능은 93.1%로 가장 높게 측정되었다.

따라서 제안하는 탐지모델을 활용하면 단일 고전 머신러닝 모델과 단일 딥러닝 모델의 성능을 모두 개선할 수 있다. 또한 한 개의 딥러닝 모델을 부분적으로 재활용하므로 여러 개의 탐지모델을 학습할 때 소요되는 시간을 줄일 수 있을 뿐만 아니라 feature를 분석하고 선별하는 과정 없이 고전 머신러닝을 활용할 수 있다.

본 논문에서 제안한 모델은 FEB에 의존하는 한계가 있다. 그러나 여러 개의 모델을 활용하는 앙상블 학습환경에서는 본 연구에서 제안하는 방법이 기존 연구에서 제안하는 단일 딥러닝 모델을 학습하는 방식보다 학습 시간을 더 단축할 수 있다. 또한 본 연구에서의 실험에 따르면 성능도 개선할 수 있는 장점이 있다. 따라서 서로 다른 여러 개의 모델을 활용하는 배깅 앙상블(bagging ensemble) 학습환경에서 매우 효과적으로 활용될 수 있을 것으로 기대된다.

따라서 향후 연구에서는 딥러닝과 고전 머신러닝을 모두 활용하는 배깅 앙상블 학습환경에서 DL-ML fusion hybrid 모델을 활용하여 학습 시간은 줄이면서 성능을 높일 수 있는 효율적인 앙상블 기계학습 연구를 진행할 예정이다.

References

[1] Darling, M., Heileman, G., Gressel, G., Ashok, A. and Poornachandran, P., "A lexical approach for classifying malicious URLs", In 2015 international conference on high performance computing & simulation (HPCS) IEEE, pp. 195-202, July. 2015.

[2] Gupta, B.B., Yadav, K., Razzak, I., Psannis, K., Castiglione, A. and Chang,

X.., "A novel approach for phishing URLs detection using lexical based machine learning in a real-time environment", Computer Communications, 175, pp. 47-57, July. 2021.

[3] Hong, J., Kim, T., Liu, J., Park, N., & Kim, S. W., "Phishing url detection with lexical features and blacklisted domains", Adaptive autonomous secure cyber systems, pp. 253-267, Feb. 2020.

[4] Le, H., Pham, Q., Sahoo, D., & Hoi, S. C., "URLNet: Learning a URL representation with deep learning for malicious URL detection", arXiv preprint arXiv:1802.03162, Feb. 2018.

[5] Tajaddodianfar, F., Stokes, J. W., & Gururajan, A., "Texception: a character/word-level deep learning model for phishing URL detection", In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) IEEE, pp. 2857-2861, May. 2020,

[6] WSuleman, M.T. and Awan, S.M., "Optimization of URL-based phishing websites detection through genetic algorithms", Automatic Control and Computer Sciences, 53, pp. 333-341, July. 2019.

[7] Garera, S., Provos, N., Chew, M. and Rubin, A.D., "A framework for detection and measurement of phishing attacks", In Proceedings of the 2007 ACM workshop on Recurring malcode, pp. 1-8, Nov. 2007.

[8] J.Sahoo, D., Liu, C. and Hoi, S.C., "Malicious URL detection using machine learning: A survey", arXiv preprint arXiv:1701.07179, Jan. 2017.

[9] Alshboul, Y., Nepali, R. and Wang, Y., "Detecting malicious short URLs on Twitter", Americas Conference on Information Systems, August. 2015.

[10] Chhabra, S., Aggarwal, A., Benevenuto, F. and Kumaraguru, P., "Phi. sh/$ ocial: the phishing landscape through short urls", In Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference, pp. 92-101, Sep. 2011.

[11] Bahnsen, A.C., Bohorquez, E.C., Villegas, S., Vargas, J. and González, F.A., "Classifying phishing URLs using recurrent neural networks", In 2017 APWG symposium on electronic crime research (eCrime) IEEE, pp. 1-8, April. 2017.

[12] Verma, R. and Dyer, K., "On the character of phishing URLs: Accurate and robust statistical learning classifiers", In Proceedings of the 5th ACM Conference on Data and Application Security and Privacy, pp. 111-122, March. 2015.

[13] Yan, H., Zhang, X., Xie, J. and Hu, C., "Detecting malicious URLS using a deep learning approach based on stacked denoising autoencoder", In Trusted Computing and Information Security: 12th Chinese Conference, CTCIS 2018, Wuhan, China, October 18, 2018, Revised Selected Papers 12 Springer Singapor., pp. 372-388, October. 2019.

[14] APWG(Anti-Phishing Working Group), "Phishing Activity Trends Report, 3rd Quarter", Dec. 2022.

[15] Wei, B., Hamad, R.A., Yang, L., He, X., Wang, H., Gao, B. and Woo, W.L., "A deep-learning-driven light-weight phishing detection sensor", Sensors, 19(19), pp. 4258, Sep. 2019.

[16] Butnaru, A., Mylonas, A. and Pitropakis, N., "Towards lightweight url-based phishing detection", Future internet, 13(6), pp. 154. Jun. 2021.

〈저 자 소 개〉

김 대 엽 (Dae-yeob Kim) 정회원
2014년 2월: 인천대학교 컴퓨터공학과 학사 졸업
2016년 2월: 과학기술연합대학원대학교 정보보호공학과 석사 졸업
2016년 3월~2019년 3월: 한컴시큐어 미래인증개발 2팀 대리
2019년 9월~2022년 6월: 한국인터넷진흥원 보안위협대응R&D팀 선임연구원
2022년 6월~현재: 국방과학연구소 정보보호팀 선임연구원
〈관심분야〉 인공지능, 웹 보안, 보안관제, 사용자인증