JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 10, pp. 2559-2567, Oct. 2023

Check for updates

모바일 버트 임베딩을 이용한 지도 및 비지도 학습 기반 악성 URL 탐지

심 기 천[1] · 김 강 석[2*]
[1]아주대학교 지식정보공학과 석사
[2*]아주대학교 사이버보안학과 교수

Malicious URL Detection based on Supervised and Unsupervised Learning using MobileBERT Embedding

Ki-Chun Sim[1] · Kangseok Kim[2*]

[1]Master's Degree, Department of Knowledge Information Engineering, Ajou University, Suwon 16499, Korea
[2*]Professor, Department of Cyber Security, Ajou University, Suwon 16499, Korea

[요 약]

기술의 발달로 인터넷 보급이 보편화됨에 따라 악성 URL과 같은 다양하고 새로운 사이버 공격의 위험이 급증하고 있다. 기존 연구에서 주로 사용되는 지도 학습 기반 악성 URL 탐지 방법의 경우 새로운 악성 URL을 탐지하는 데 한계가 있어 본 논문에서는 지도 학습뿐만 아니라 비지도 학습 기반 악성 URL 탐지 방법을 제안하였다. MobileBERT를 사용하여 URL 문자열을 토큰화하고 어휘 특성을 고려한 임베딩 벡터로 변환하여 안정성과 정확성을 높이기 위한 시도를 하였다. 임베딩 벡터를 PCA 및 오토인코더로 차원 축소한 후 XGBoost 및 LOF에 입력하여 성능 평가를 수행하였다. 지도 학습의 경우 오토인코더보다 PCA를 활용하여 임베딩 벡터의 차원을 축소했을 때 더 높은 성능을 보였고, 차원 축소 없이 단지 MobileBERT로 임베딩한 벡터의 사용만으로도 높을 확률로 악성 URL을 탐지할 수 있었다. 비지도 학습의 경우 전반적으로 재현율이 정밀도보다 높았으며, 정상 데이터의 샘플 수를 증가시킬수록 탐지 성능이 향상되는 것을 확인할 수 있었다.

[Abstract]

As the spread of the Internet becomes more widespread owing to the development of technology, the risk of various new cyber-attacks such as malicious URLs is rapidly increasing. The supervised learning-based malicious URL detection method mainly used in existing research has limitations in detecting new malicious URLs. Therefore, the current study proposed detection method based on unsupervised as well as supervised learning. An attempt was made to increase reliability and accuracy using MobileBERT to tokenize URL strings and convert them into embedding vectors that take lexical features into account. The embedding vector was dimensionally reduced using PCA and autoencoder and then inputted into XGBoost and LOF to evaluate performance. In the case of supervised learning, higher performance was achieved when the dimension of the embedding vector was reduced using PCA rather than an autoencoder, and malicious URLs could be detected with a high probability just by using vectors embedded through mobileBert without dimensionality reduction. In the case of unsupervised learning, overall recall was higher than precision, and increasing the number of samples of normal data improved detection performance.

색인어 : 악성 URL 탐지, 모바일 버트, 지도 학습, 비지도 학습, 차원 축소
Keyword : Malicious URL Detection, MobileBERT, Supervised Learning, Unsupervised Learning, Dimensionality Reduction

http://dx.doi.org/10.9728/dcs.2023.24.10.2559

Received 30 August 2023; **Revised** 14 September 2023
Accepted 18 September 2023

***Corresponding Author; Kangseok Kim**

Tel: +82-31-219-2496
E-mail: kangskim@ajou.ac.kr

Ⅰ. 서 론

기술의 발달로 인터넷 보급이 보편화됨에 따라 악성 URL과 같은 다양하고 새로운 사이버 공격의 위험이 급증하고 있다[1]. 이로 인해 악성 URL 탐지 연구가 더욱 중요해지고 있다. 악성 URL(Malicious URL, Malicious Website)은 사용자를 바이러스 공격, 피싱(Phishing) 공격, 맬웨어(Malware), 사기 행위 등의 위험에 빠뜨릴 수 있는 악성 웹 페이지로 연결하도록 설계된 링크이다. 사용자가 악성 URL을 클릭하면 트로이 목마(Trojan Horse), 랜섬웨어(Ransomware), 웜(Worm) 등의 악성 소프트웨어가 다운로드 될 수 있으며, 다운로드 된 악성 소프트웨어가 사용자의 개인 정보에 접근하여 사용자의 장치를 손상시키고, 금전적 손실을 입힐 수 있다. 이러한 점으로 인해 웹사이트 보안이 더욱 중요해지고 있는 실정이다.

현재 악성 URL을 식별하는 주요 솔루션 중 하나는 해당 URL을 이미 악성으로 알려진 모든 URL의 목록인 블랙리스트(Blacklist)에 추가하는 것이다[2]. 그러나 블랙리스트를 사용하는 접근 방식은 새롭게 생성된 악성 URL을 탐지하는 데는 적합하지 않아 이러한 단점을 해결하는 악성 URL 탐지 방법이 필요하다. 따라서 최근에는 기계 학습 및 딥러닝 기반 악성 URL 탐지 기술을 통해 악성 URL 탐지기의 일반성을 향상시키는 방법이 연구되고 있다.

악성 URL 분석을 위한 특성 추출에 URL의 정적 분석과 웹사이트 방문의 동적 분석이 활용된다. 그러나 대부분의 연구가 보안 및 비용 상의 이유로 정적 분석에 중점을 두고 있다[3]. 이에 본 연구에서도 Malicious URLs dataset[4]을 사용한 정적 분석을 수행하였다. 악성 URL에는 불법적인 패턴이 URL 문자열에 포함될 수 있고, 악성 URL을 기계 학습 기반 모델에 주입하기 위해서는 도메인 정보, 경로 토큰 수, URL 길이 등과 같은 다양한 어휘적 특성을 URL 문자열로부터 추출하는 것이 필요하다. 따라서 본 연구에서는 입력 데이터인 URL 문자열을 토큰화하고, 어휘적 특성을 고려한 임베딩 벡터로 변환하는 과정에 모바일 버트(MobileBERT)를 사용하였으며, 512차원으로 임베딩된 데이터의 차원 축소를 위해 주성분 분석(Principal Component Analysis, PCA)과 오토인코더(Autoencoder)[5]를 사용하였다. 이후 차원 축소된 임베딩 벡터를 지도 및 비지도 학습 기반 악성 URL 탐지 모델에 입력하였고, 지도 학습 기반 분류기로 XGBoost (Extreme Gradient Boosting)[6]를 사용하였다. 또한 비지도 학습 알고리즘인 오토인코더와 비지도 학습 기반 분류기인 LOF(Local Outlier Factor)[7]를 사용하여 비지도 학습 기반 악성 URL 탐지 실험도 수행하였다. 이렇듯 본 연구에서는 모바일 버트와 지도 및 비지도 기계 학습/딥러닝 모델을 활용한 악성 URL 탐지 방법을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 악성 URL 탐지 방법 관련 연구를 기술하였고, 3장에서는 본 연구에서 사용한 데이터, 임베딩 기법, 제안한 지도 및 비지도 기계 학습/딥러닝 모델 기반 악성 URL 탐지 방법을 서술하였으며, 4장에서는 제안 방법들에 관한 실험 결과를 분석하였다. 마지막으로 5장에서는 결론 및 향후 연구 방향에 대해 제시하였다.

Ⅱ. 관련 연구

기존 연구에서는 악성 URL 탐지에 주로 블랙리스트를 사용하지만 블랙리스트를 활용하는 방식은 새롭게 생성된 악의적인 URL을 탐지하는 데 어려움이 있다[8]. 따라서 최근에는 악성 URL 탐지기의 일반성을 향상시키기 위해 기계 학습 및 딥러닝 기반 악성 URL 탐지 기술이 연구되고 있다[9],[10]. 또한 기존 연구에서는 대부분 지도 학습 기반 악성 URL 탐지 방법이 사용되었지만 본 연구에서는 지도 학습뿐만 아니라 비지도 학습 기반 악성 URL 탐지 방법도 제안한다.

Aljabri 등[1]은 탐지 기술과 사용된 데이터셋의 한계를 고려하여 기계 학습 모델 기반 악성 URL을 탐지하는 방법을 설명하고, 아랍어 악성 웹사이트 탐지와 관련된 연구 방향을 기술하였다. Angadi 등[11]은 악성 URL 탐지에 사용되는 분류기의 효율성을 높이기 위해 URL의 호스트 기반 및 어휘 측면을 활용할 것을 권장하고, 악성 및 양성 URL을 AdaBoost 및 Random Forest 기계 학습 모델을 사용하여 분류하였다. Wang 등[3]은 문자열의 특징을 추출하기 위해 DCNN(Dynamic Convolutional Neural Network) 기반 악성 URL 탐지 모델을 제안하고, 문자 임베딩 기반의 단어 임베딩(Word Embedding based on Character Embedding)을 활용하여 URL의 벡터 표현을 학습하는 임베딩 방법을 제안하였다. Saleem 등[12]은 URL의 어휘적 특성만을 포함하는 기계 학습 기반의 경량 방법을 제안하였다. 제안된 경량화 방식은 URL에서 추출된 특성을 활용하여 실행 시간과 저장 공간을 줄인다. Mehndiratta 등[13]은 등록된 도메인과 URL 경로 사이에 연결된 상호 연관성을 통해 URL을 특성화하고, 속성을 추출하여 다양한 기계 학습 알고리즘에 적용하는 피싱 URL 탐지 방식을 제안하였다. Cui 등[14]은 그래디언트 학습(Gradient Learning)을 기반으로 한 통계 분석과 시그모이드 임계값을 이용한 특징 추출을 결합하여 다양한 기계 학습 기법을 기반으로 하는 탐지 접근 방식을 제안하였다.

Ⅲ. 연구 방법

본 연구에서는 모바일 버트를 사용한 임베딩 기법을 통해 URL 문자열을 512차원의 벡터로 변환하고, PCA 및 오토인코더를 통해 차원 축소를 하거나 잠재 벡터(Latent Vector)를 추출한 후, 지도 학습 기반 분류기인 XGBoost와 비지도 학습 기반 분류기인 LOF를 통해 악성 URL을 탐지한다. 본

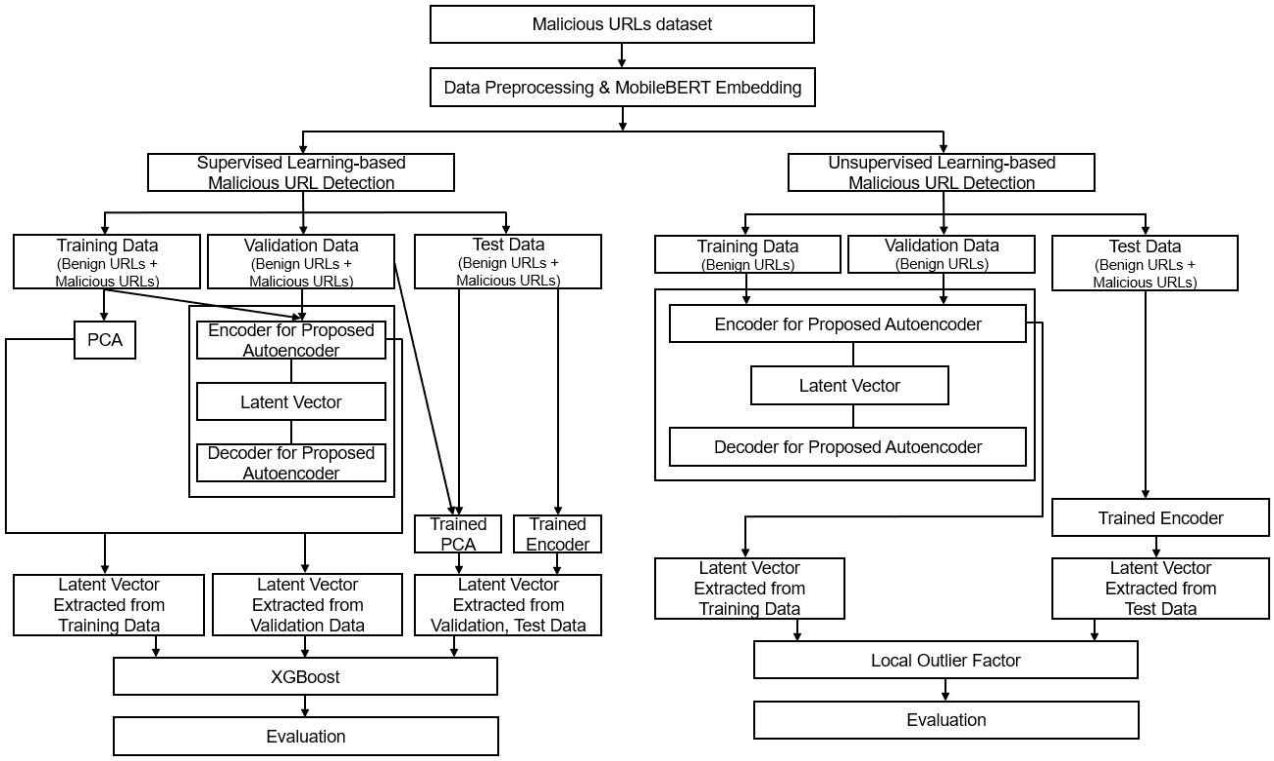

그림 1. 본 연구에서 제안하는 악성 URL 탐지 방법 워크플로우
Fig. 1. Overall workflow for malicious URL detection method proposed in this study

연구에서 제안하는 악성 URL 탐지 방법은 크게 3단계로 나뉜다. 첫 번째로 모바일 버트를 이용하여 URL 문자열을 임베딩 벡터로 변환한다. 두 번째로 변환된 임베딩 벡터를 PCA 및 Conv1D 기반 오토인코더 모델에 입력하여 해당 벡터의 차원을 축소하고, 이를 지도 학습 기반 악성 URL 탐지에 사용한다. 또한 동일한 임베딩 벡터를 Conv1D 기반 오토인코더에 입력하여 잠재 벡터를 추출하고, 이를 비지도 학습 기반 악성 URL 탐지에 사용한다. 세 번째로 차원이 축소된 벡터를 지도 학습 기반 분류기인 XGBoost에 입력하여 악성 URL을 탐지하고, 비지도 학습 기반 분류기인 LOF에 입력하여 이진 분류(Binary Classification)를 통해 악성 URL을 탐지한다. 이후 실험 결과를 분석하고, 악성 URL 탐지 방법의 성능을 평가한다. 그림 1은 본 연구에서 제안하는 악성 URL 탐지 방법의 전체적인 작업 흐름을 나타낸다.

3-1 실험 데이터셋

실험에 사용된 데이터는 Malicious URLs dataset이다[4]. 해당 데이터는 ISCX-URL-2016, malware domain black list dataset, faizan git repo, Phishtank dataset 및 PhishStorm dataset 등에서 수집한 양성 URLs(Benign URLs), 변조 URLs(Defacement URLs), 피싱 URLs(Phishing URLs), 맬웨어 URLs(Malware URLs) 등 총 651,191개의 URL로 구성되어 있으며, 이를 표 1에 정리하였다. 실험 데이터에 있는 악성 URL 공격 (피싱, 맬웨어 및 변조 URL 공격) 기법에 대한 간략한 설명은 다음과 같다. 피싱 URL 공격(Phishing URL Attack)은 신용 카드 번호와 같은 사용자의 개인 정보를 도용하기 위해 사용자의 컴퓨터에 액세스를 시도하는 가짜 웹사이트를 열도록 유도하는 공격이고, 맬웨어 URL 공격(Malware URL Attack)은 사용자의 장치에 랜섬웨어, 키 로거, 트로이 목마, 스파이웨어, 컴퓨터 웜, 바이러스 등의 맬웨어를 설치하는 악성 웹사이트로 연결하는 공격이며, 웹사이트 변조 URL 공격(Website Defacement URL Attack)은 해커에 의해 변경된 악의적인 웹사이트로 사용자를 리디렉션(Redirection) 하는 공격이다[9],[15].

표 1. URL의 유형에 따른 샘플 수
Table 1. Number of samples according to type of URL

Type of URL	Number of URLs
Benign	428,103
Defacement	96,457
Phishing	94,111
Malware	32,520

중복된 URL을 제거하고, 모바일 버트를 사용하여 임베딩 벡터로 변환한 후, 지도 학습 기반 악성 URL 탐지에 사용할 데이터와 비지도 학습 기반 악성 URL 탐지에 사용할 데이터

로 전처리하였다. 먼저 생성된 임베딩 벡터를 6:2:2의 비율로 훈련 데이터(Training Data), 검증 데이터(Validation Data), 테스트 데이터(Test Data)로 나누고, PCA 및 Conv1D 기반 오토인코더 모델에 입력하여 차원을 축소하는 과정을 거쳐 지도 학습 기반 악성 URL 탐지에 사용할 데이터로 가공하였다. 또한 동일한 임베딩 벡터를 이용하여 양성 URL과 악성 URL의 비율이 1:1이 되도록 테스트 데이터를 구성하고, 남은 양성 URL로 훈련 데이터와 검증 데이터를 구성하여 비지도 학습 기반 악성 URL 탐지에 사용할 데이터로 가공하였다. 표 2는 지도 학습 기반 악성 URL 탐지에 사용되는 훈련, 검증, 테스트 샘플 수와 양성, 악성 샘플 수를 나타내고, 표 3은 비지도 학습 기반 악성 URL 탐지에 사용되는 훈련, 검증, 테스트 샘플 수와 양성, 악성 샘플 수를 나타낸다.

표 2. 지도 학습 기반 악성 URL 탐지에 사용된 훈련, 검증, 테스트 샘플 수

Table 2. Number of training, validation, and test samples used for supervised learning-based malicious URL detection

Training Data	Number of Benign Samples	256,848
	Number of Malicious Samples	127,827
Validation Data	Number of Benign Samples	85,616
	Number of Malicious Samples	42,609
Test Data	Number of Benign Samples	85,616
	Number of Malicious Samples	42,609

표 3. 비지도 학습 기반 악성 URL 탐지에 사용된 훈련, 검증, 테스트 샘플 수

Table 3. Number of training, validation, and test samples used for unsupervised learning-based malicious URL detection

Data	Number of samples	Type of Malicious URL		
		Defacement	Phishing	Malware
Training Data	Benign	279,097	280,111	339,196
	Malicious	0	0	0
Validation Data	Benign	53,675	53,877	65,239
	Malicious	0	0	0
Test Data	Benign	47,654	47,046	11,822
	Malicious	47,654	47,046	11,822

3-2 모바일 버트 임베딩(MobileBERT Embedding)

악성 URL에는 위법한 패턴이 URL 문자열에 포함될 수 있기 때문에 기계 학습 기반 모델을 이용하여 악성 URL을 탐지하려면 URL 문자열로부터 다양한 어휘적 특성을 추출하는 것이 필요하다. 이에 본 연구에서는 모바일 버트를 사용하여 입력 데이터인 URL 문자열을 토큰화하고, 어휘적 특성을 고려한 임베딩 벡터로 변환하였다.

임베딩은 원본 데이터를 기계 학습 알고리즘에 직접 입력하는 대신 비지도 사전 훈련을 통해 생성된 원본 데이터의 새로운 표현을 학습하는 방법이다. 이 과정을 통해 원본 데이터의 본질인 내재된 패턴을 학습하고, 내재된 구조 외에 부가적인 노이즈 데이터를 제거한다. 또한 새로운 표현을 지도 학습 알고리즘에 공급하면 노이즈가 줄어들어 일반화 오차도 줄어든다. 본 연구에서는 카네기 멜론 대학(Carnegie Mellon University)과 Google Brain의 공동 연구로 2020년 4월에 공개한 경량화된 BERT(Bidirectional Encoder Representations from Transformers - MobileBERT) 모델을 사용하였다. 모바일 버트는 지식 전이 학습(Knowledge Transfer Learning) 기법을 이용하여 모바일 기기에서도 구동 가능한 경량화된 BERT 모델이다[16]. 2018년 구글에서 발표한 언어 모델인 BERT를 압축하고 가속하기 위한 양방향 트랜스포머인 BERT-Large 만큼 복잡한 모델을 학습시킨 후, 학습한 지식을 가벼운 모델로 전이하는 방법을 사용한다.

3-3 차원 축소

차원 축소는 모델에 사용되는 특성의 개수를 줄이면서 유용한 특성들을 추출하여 모델의 탐지 속도를 높이고, 성능을 향상시키는 기술이다[17]. PCA는 가능한 한 많은 변동성을 보존하는 것에 초점을 두어 선형 변환을 하는 차원 축소 방법이고[18], 오토인코더는 입력 데이터의 차원을 낮추어 잠재 벡터를 추출하는 인코더(Encoder)와 추출된 잠재 벡터를 사용하여 입력 데이터와 유사한 데이터를 재구성하는 디코더(Decoder)가 결합한 딥러닝 기반 비지도 학습 방법이다[19]. 본 연구에서는 512차원의 임베딩 벡터를 PCA에 입력하여 50, 100, 150, 200 차원으로 축소 시켰고, 동일한 임베딩 벡터를 Conv1D 기반 오토인코더 모델에 입력하여 50, 100 차원으로 축소 시켜 지도 학습 기반 악성 URL 탐지에 사용하였다.

3-4 지도 및 비지도 학습 기반 이상 탐지 방법

1) 지도 학습 기반 XGBoost(Extreme Gradient Boosting)

본 연구에서는 지도 학습 기반 앙상블 알고리즘이며, 최적화된 분산 그래디언트 부스팅 라이브러리인 XGBoost[20]를 사용하였다. 결정 트리(Decision Tree)의 앙상블 알고리즘인 XGBoost는 일반적으로 성능 및 속도가 뛰어나 산업계에서 널리 사용되고 있다. 본 연구에서는 PCA를 사용하여 유용한 특성이 추출된 50, 100, 150, 200차원의 벡터들과 Conv1D 기반 오토인코더 모델을 사용하여 유용한 특성이 추출된 50, 100차원의 벡터들을 지도 학습 기반 분류기인 XGBoost에 입력하여 실험을 진행하였다.

2) 비지도 학습 기반 오토인코더와 LOF

본 연구에서 개발한 Conv1D 기반 오토인코더 모델은 입력층(Input Layer) 및 Conv1D 층을 포함하는 인코더와 Conv1DTranspose 층 및 출력층(Output Layer)을 포함하는 디코더로 구성되어 있다. 해당 모델에 데이터가 입력되면 인코더에서 필터(Filter) 수가 8에서 64까지 2배씩 순차적으로 증가하는 4개의 Conv1D 층, 4개의 MaxPooling1D 층 및 다차원 배열을 1차원 배열로 선형 변환하는 Flatten 층을 거쳐 잠재 벡터로 압축된다. 이후 디코더에서 압축된 잠재 벡터가 4개의 Conv1DTranspose 층을 거쳐 입력 데이터와 유사하게 재구성된다. 또한 모델 컴파일(Compile) 단계에서는 'Adam'을 최적화 알고리즘(Optimizer)으로 사용하고, 'Mean Squared Error'를 손실 함수(Loss Function)로 사용한다.

본 연구에서는 모바일 버트로 임베딩한 512차원의 벡터를 Conv1D 기반 오토인코더 모델에 입력하여 내재되어 있는 유용한 패턴을 학습시키고, 50, 100, 128차원의 잠재 벡터를 추출하였다. 이후 잠재 벡터를 주어진 데이터셋에 속한 각 개체의 제한된 주변만 고려하여 개체가 얼마나 외곽에 있는지 정량화하는 밀도 기반 이상 탐지 알고리즘인 LOF[7]에 입력하여 비지도 학습 기반 악성 URL 탐지 실험을 수행하였다. 그러나 실험에 사용된 악성 URL의 유형(클래스) 별(변조, 피싱, 맬웨어 URL) 데이터가 불균형을 이루고 있기 때문에 적은 데이터 유형에 대한 탐지율이 낮아질 수 있어 비지도 학습 기반 실험에서 각각의 악성 URL 유형에 대해 이진 분류를 수행하였다.

3-5 제안한 악성 URL 탐지 방법의 성능 평가 지표

본 연구에서는 지도 학습 기반 악성 URL 탐지를 위해 차원 축소된 임베딩 벡터를 XGBoost 분류기에 입력하여 실험을 진행하였다. 또한 비지도 학습 기반 악성 URL 탐지를 위해 차원 축소된 임베딩 벡터를 Conv1D 기반 오토인코더에 입력하여 잠재 벡터를 추출한 뒤, LOF에 입력하여 이진 분류를 수행하였다. 수행된 실험을 통해 나온 결과를 분석하고, 정확도(Accuracy), 모델이 악성 URL로 판단한 것 중 실제로 악성 URL인 것의 비율인 정밀도(Precision), 실제 악성 URL 중 모델이 악성 URL로 판단한 것의 비율인 재현율(Recall), F1-Score, AUC(Area Under the ROC Curve) 등의 지표를 사용하여 모델의 성능을 평가하였다.

표 4. 실험 환경
Table 4. Experimental environment

OS	Ubuntu 18.04.6 LTS
CPU	Intel(R) Xeon(R) Gold 5120
GPU	NVIDIA RTX A5000
RAM	264GB
Python	3.10.9
Scikit-Learn	1.2.1
Keras	2.11.0

Ⅳ. 실험 결과 및 분석

본 연구에서 제안하는 악성 URL 탐지 방법에 따라 악성 URL 탐지 실험을 진행하고, 성능 평가를 수행하였다. 수행된 실험 환경은 표 4와 같다.

3장에서 설명하였듯이 본 연구에서는 악성 URL이 포함된 실험 데이터를 모바일 버트를 사용해 512차원의 임베딩 벡터로 변환하였다. 이후 임베딩 벡터를 PCA에 입력하여 50, 100, 150, 200차원으로 축소하고, Conv1D 기반 오토인코더 모델에 입력하여 50, 100차원으로 축소한 후, 차원이 축소된 벡터들을 XGBoost 분류기에 입력하여 지도 학습 기반 악성 URL 탐지 실험을 수행하였다. 또한 동일한 임베딩 벡터를 Conv1D 기반 오토인코더 모델에 입력하여 50, 100, 128차원의 잠재 벡터를 추출한 후, LOF에 입력하여 비지도 학습 기반 악성 URL 탐지 실험을 수행하였다. 지도 학습 기반 악성 URL 탐지 실험의 경우, 검증 데이터를 XGBoost에 입력하여 우수한 성능을 도출하는 하이퍼파라미터(Hyperparameter)를 얻은 후, 테스트 데이터를 XGBoost에 입력하여 모델의 성능을 평가할 때 사용하였으며, 그 결과를 표 5에 정리하였다. 비지도 학습 기반 악성 URL 탐지 실험의 경우, 2개의 테스트 데이터를 만들고, 그중 하나를 LOF에 입력하여 우수한 성능을 도출하는 하이퍼파라미터를 얻은 후, 나머지 하나를 XGBoost에 입력하여 모델의 성능을 평가할 때 사용하였다. 또한 비지도 분류 학습의 실험에서 훈련 데이터의 양성(Benign) 샘플 수를 30% 증가시켜, 실험 후 성능 변화에 대한 결과를 표 6에 정리하였다.

3장에서 설명하였듯이 본 연구에서는 악성 URL이 포함된 실험 데이터를 모바일 버트를 사용해 512차원의 임베딩 벡터로 변환하였다. 이후 임베딩 벡터를 PCA에 입력하여 50, 100, 150, 200차원으로 축소하고, Conv1D 기반 오토인코더 모델에 입력하여 50, 100차원으로 축소한 후, 차원이 축소된 벡터들을 XGBoost 분류기에 입력하여 지도 학습 기반 악성 URL 탐지 실험을 수행하였다. 또한 동일한 임베딩 벡터를 Conv1D 기반 오토인코더 모델에 입력하여 50, 100, 128차원의 잠재 벡터를 추출한 후, LOF에 입력하여 비지도 학습 기반 악성 URL 탐지 실험을 수행하였다. 지도 학습 기반 악성 URL 탐지 실험의 경우, 검증 데이터를 XGBoost에 입력하여 우수한 성능을 도출하는 하이퍼파라미터(Hyperparameter)를 얻은 후, 테스트 데이터를 XGBoost에 입력하여 모델의 성능을 평가할 때 사용하였으며, 그 결과를 표 5에 정리하였다. 비지도 학습 기반 악성 URL 탐지 실험의 경우, 2개의 테스트 데이터를 만들고, 그중 하나를 LOF에 입력하여 우수한 성능을 도출하는 하이퍼파라미터를 얻은 후, 나머지 하나를 XGBoost에 입력하여 모델의 성능을 평가할 때 사용하였다. 또한 비지도 분류 학습의 실험에서 훈련 데이터의 양성(Benign) 샘플 수를 30% 증가시켜, 실험 후 성능 변화에 대한 결과를 표 6에 정리하였다.

표 5. 지도 학습 기반 악성 URL 탐지 성능 평가
Table 5. Performance of supervised learning-based malicious URL detection

Dimensionality Reduction	Vector Dimension	Test Data		Accuracy	Precision	Recall	F1-Score
		Number of Benign Samples	Number of Malicious Samples				
PCA	50	85616	42609	0.922	0.916	0.862	0.887
	100	85616	42609	0.937	0.935	0.887	0.909
	150	85616	42609	0.942	0.940	0.893	0.915
	200	85616	42609	**0.945**	**0.944**	**0.900**	**0.920**
Autoencoder	50	85616	42609	0.908	0.899	0.842	0.867
	100	85616	42609	0.918	0.913	0.856	0.882
None	512	85616	42609	0.939	0.930	0.894	0.910

표 6. 비지도 학습 기반 악성 URL 탐지 성능 평가
Table 6. Performance of unsupervised learning-based malicious URL detection

Type of Malicious URL	Dimension of Latent Vector	Training Data	Accuracy	Precision	Recall	F1-Score	AUC
		Number of Benign Samples					
Defacement	50	214690	0.609	0.586	0.742	0.655	0.609
		279097	0.648	0.611	0.814	0.698	0.648
	100	214690	0.617	0.594	0.743	0.660	0.617
		279097	0.663	0.620	0.842	0.714	0.663
	128	214690	0.631	0.601	0.779	0.679	0.631
		279097	0.660	0.618	0.839	0.712	0.660
Phishing	50	215470	0.642	0.608	0.797	0.690	0.642
		280111	0.655	0.615	0.827	0.705	0.655
	100	215470	0.648	0.612	0.809	0.697	0.648
		280111	0.654	0.615	0.822	0.704	0.654
	128	215470	0.658	0.617	0.830	0.708	0.658
		280111	0.657	0.616	0.833	0.708	0.657
Malware	50	260920	0.689	0.635	0.892	0.742	0.689
		339196	0.683	0.631	0.884	0.736	0.683
	100	260920	0.691	0.636	0.894	0.743	0.691
		339196	0.696	0.639	0.898	0.747	0.696
	128	260920	**0.718**	**0.651**	0.937	**0.769**	**0.718**
		339196	0.716	0.649	**0.944**	**0.769**	0.716

4-1 지도 학습 기반 악성 URL 탐지 실험 결과 분석

표 5에서 Conv1D 기반 오토인코더보다 PCA를 사용하여 차원 축소를 했을 때 정확도, 정밀도, 재현율, F1-Score 등 전반적인 탐지 성능이 높은 것을 확인하였다. 향후 학습이 어렵지만 비선형 데이터에 적합한 오토인코더의 성능을 높이기 위하여 파인 튜닝(Fine Tuning)이 필요하다고 판단된다. 또한 임베딩 벡터를 PCA에 입력하여 200차원으로 축소한 후 XGBoost 분류기에 입력했을 때 정밀도가 0.944, 재현율이 0.9로 가장 높았으며, 차원을 축소하지 않은 원본 512차원 입력 데이터의 경우에도 높은 성능을 보였는데, 이는 문자열의 문맥 의미(Semantic)를 고려하여 특성을 추출하는 모바일 버트의 임베딩 방법 때문으로 보인다. 향후 모바일 버트를 활용한 더 다양한 전이 학습을 통해 탐지율(Recall)을 높이기 위한 의미 있는 URL 문자열 특징을 추출하는 연구를 진행할 것이다. 또한 일반적으로 알려진 공격 유형의 학습을 통해 공격을 추론하는 지도 학습의 특성상 탐지율이 정밀도에 비해 떨어지는 경향을 보였고, 지도 학습 기반 악성 URL 탐지 성능을 본 연구에서 수행된 비지도 학습 기반 악성 URL 탐지 성능의 기준(Baseline)으로 활용하였다.

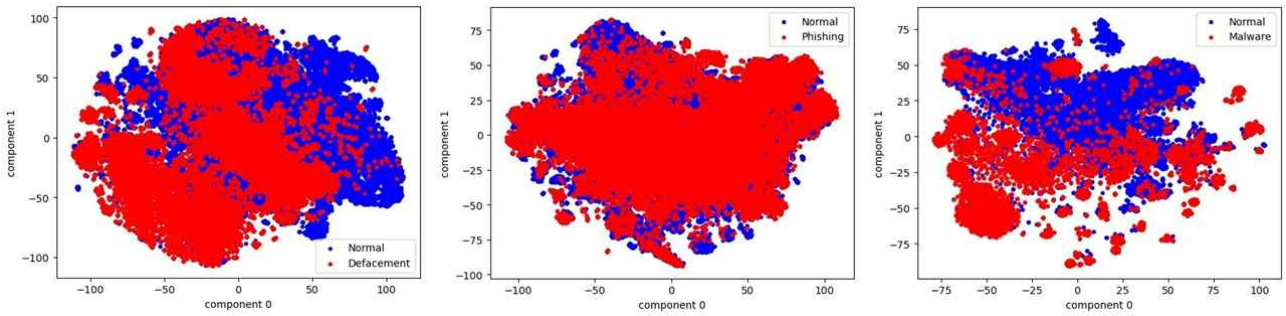

그림 2. t-SNE를 사용하여 테스트 데이터를 2D로 축소한 시각화 : Benign vs (Defacement, Phishing, Malware)
Fig. 2. 2D reduced visualization of test data using t-SNE : Benign vs (Defacement, Phishing, Malware)

4-2 비지도 학습 기반 악성 URL 탐지 실험 결과 분석

비지도 학습 기반 악성 URL 탐지 실험에서는 악성 유형별 데이터의 불균형에 따라 이진 분류를 수행하였다. 사용자가 악성 URL로 인해 개인 정보의 유출이나 금전적 손실 같은 심각한 피해를 입을 수 있기 때문에 아직 알려지지 않은 새로운 악성 URL을 탐지하는 것이 중요하다고 판단되어 재현율을 높이는 데 주력하였으며, 이는 표 6에서 확인할 수 있다. 또한 (그림 2)에서 유사한 샘플(URL)은 가까이, 유사하지 않은 샘플은 멀리 떨어지도록 학습하여 차원을 축소하는 알고리즘인 t-SNE(t-Distributed Stochastic Neighbor Embedding) [21]를 활용하여 테스트 데이터를 시각화하였다. 맬웨어 URL 탐지율이 다른 두 악성 URL 유형에 비해 높았으며, 정밀도가 0.651, 재현율이 0.944를 보였다. 시각화를 통해서도 맬웨어가 다른 두 유형에 비해 잘 구분되는 것을 확인할 수 있다. 웹사이트 변조 URL 공격 유형의 경우 탐지율이 다른 두 유형에 비해 떨어지는데 이는 사용자가 신뢰할 수 있는 웹사이트라도 손상된 사기 URL을 전달할 수 있는 공격 특성상 URL 문자열로만 판별하기 어렵기 때문이다.

표 6을 살펴보면 전반적으로 잠재 벡터의 차원이 작아질수록 탐지 성능이 감소하는 모습을 보였으며, 이는 어느 임계점을 넘어 원본 데이터를 잠재 벡터로 압축하면 정보 손실이 발생하기 때문으로 보인다. 또한 일반적으로 기계 학습 모델을 학습시키기 위하여 정기적으로 새로 획득한 양성 패턴을 학습 데이터에 반영하는 것이 필요하다는 가정 하에 실험을 진행하였다. 훈련 데이터의 양성 샘플 수를 증가시키면 악성 URL의 유형이나 잠재 벡터의 차원과 상관없이 탐지 성능이 향상되는 것을 확인할 수 있으며, 이는 모델이 양성 URL에 내재된 패턴을 더 많이 학습했기 때문으로 분석된다. 향후 더 많은 양성 및 악성 URL 데이터를 수집하여 실험을 수행할 것이다.

전반적으로 아직 알려지지 않은 새로운 공격 유형을 탐지하기 위한 비지도 학습 특성상 악성 URL 탐지 성능이 지도 학습 기반 악성 URL 탐지 성능에 미치지 못하고, 특히 정밀도가 재현율에 비해 뒤떨어지기 때문에 향후 재현율 외에도 정밀도를 향상시키는 연구가 필요하다.

V. 결 론

본 연구에서는 URL 데이터를 모바일 버트를 사용하여 임베딩한 후 PCA 및 오토인코더로 차원 축소하여 지도 학습 기반 XGBoost 알고리즘 및 비지도 학습 기반 LOF 알고리즘을 이용한 악성 URL 탐지 방법을 제안하였다. 실험 결과 지도 학습의 경우 URL 데이터를 모바일 버트로 임베딩하기만 해도 준수한 성능이 도출되었으며, PCA를 활용하여 임베딩 벡터의 차원을 150, 200으로 축소했을 때 기존보다 탐지 성능이 향상되었다. 비지도 학습의 경우 잠재 벡터의 차원이 감소할수록 탐지 성능이 저하되는 경향을 보였으며, 이는 어느 임계점을 넘어 원본 데이터를 잠재 벡터로 압축하면 정보 손실이 발생하기 때문으로 보인다. 또한 훈련 데이터의 양성 샘플 수를 증가시키면 탐지 성능이 향상되는 것을 확인할 수 있었으며, 이는 모델이 양성 URL 데이터에 내재된 패턴을 더 많이 학습하기 때문으로 보인다. 전반적으로 정밀도가 재현율에 미치지 못하기 때문에 향후 연구에서는 보다 더 다양한 어휘 특징 추출 및 효과적인 임베딩을 위한 표현 학습 방법을 고려하여 비지도 학습 기반 악성 URL 탐지 방법의 성능을 향상시킬 것이다.

감사의 글

본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1-F1A1059036).

참고문헌

[1] M. Aljabri, H. S. Altamimi, S. A. Albelali, M. Al-Harbi, H. T. Alhuraib, N. K. Alotaibi, ... and K. Salah, “Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions,” *IEEE Access*, Vol. 10, pp. 121395-121417, November 2022. https://doi.org/10.110

9/ACCESS.2022.3222307

[2] H. V. S. Aalla, N. R. Dumpala, and M. Eliazer, “Malicious URL Prediction Using Machine Learning Techniques,” *Annals of the Romanian Society for Cell Biology*, Vol. 25, No. 5, pp. 2170-2176, 2021.

[3] Z. Wang, X. Ren, S. Li, B. Wang, J. Zhang, and T. Yang, “A Malicious URL Detection Model Based on Convolutional Neural Network,” *Security and Communication Networks*, Vol. 2021, pp. 1-12, May 2021. https://doi.org/10.1155/2021/5518528

[4] Kaggle. Malicious URLs dataset [Internet]. Available: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset.

[5] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, Vol. 313, No. 5786, pp. 504-507, July 2006. https://doi.org/10.1126/science.1127647

[6] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD ’16)*, San Francisco: CA, pp. 785-794, August 2016. https://doi.org/10.1145/2939672

[7] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying Density-Based Local Outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data (SIGMOD ’00)*, Dallas: TX, pp. 93-104, May 2000. https://doi.org/10.1145/342009.335388

[8] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs,” in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD ’09)*, Paris, France, pp. 1245-1254, June 2009. https://doi.org/10.1145/1557019.1557153

[9] D. Sahoo, C. Liu, and S. C. H. Hoi, “Malicious URL Detection Using Machine Learning: A Survey,” arXiv:1701.07179, August 2019. https://doi.org/10.48550/arXiv.1701.07179

[10] V. Vundavalli, F. Barsha, M. Masum, H. Shahriar, and H. Haddad, “Malicious URL Detection Using Supervised Machine Learning Techniques,” in *Proceedings of the 13th International Conference on Security of Information and Networks (SIN 2020)*, Merkez, Turkey, pp. 1-6, November 2020. https://doi.org/10.1145/3433174.3433592

[11] S. Angadi and S. Shukla, “Malicious URL Detection Using Machine Learning Techniques,” in *Proceedings of the 18th International Conference on Information Systems Security (ICISS 2022)*, Tamil Nadu, India, pp. 657-669, December 2022. https://doi.org/10.1007/978-981-19-2894-9_50

[12] R. A. Saleem, R. Vinodini, and A. Kavitha, “Lexical Features Based Malicious URL Detection Using Machine Learning Techniques,” *Materials Today: Proceedings*, Vol. 47, pp. 163-166, 2021. https://doi.org/10.1016/j.matpr.2021.04.041

[13] M. Mehndiratta, N. Jain, A. Malhotra, I. Gupta, and R. Narula, “Malicious URL: Analysis and Detection Using Machine Learning,” in *Proceedings of the 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, pp. 1461-1465, March 2023. https://ieeexplore.ieee.org/document/10112229

[14] B. Cui, S. He, X. Yao, and P. Shi, “Malicious URL Detection with Feature Extraction Based on Machine Learning,” *International Journal of High Performance Computing and Networking*, Vol. 12, No. 2, pp. 166-178, September 2018. https://doi.org/10.1504/IJHPCN.2018.094367

[15] C. Johnson, B. Khadka, R. B. Basnet, and T. Doleck, “Towards Detecting and Classifying Malicious URLS using Deep Learning,” *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, Vol. 11, No. 4, pp. 31-48, December 2020. https://doi.org/10.22667/JOWUA.2020.12.31.031

[16] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, and D. Zhou, “MobileBERT: A Compact Task-Agnostic BERT for Resource-Limited Devices,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 2158-2170, July 2020. https://doi.org/10.18653/v1/2020.acl-main.195

[17] J.-E. Yoon and K. Kim, “Comparison of Dimensional Reduction and Oversampling Methods for Efficient Network Anomaly Detection,” *Journal of Digital Contents Society*, Vol. 24, No. 3, pp. 583-591, March 2023. https://doi.org/10.9728/dcs.2023.24.3.583

[18] I. T. Jolliffe and J. Cadima, “Principal Component Analysis: A Review and Recent Developments,” *Philosophical Transactions of the Royal Society A*, Vol. 374, No. 2065, 20150202, April 2016. https://doi.org/10.1098/rsta.2015.0202

[19] A. Géron, *Hands-On Machine Learning with Scikit-Learn, Keras, & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 2nd ed. Sebastopol, CA: O’Reilly, 2019.

[20] GitHub. XGBoost [Internet]. Available: https://github.com/dmlc/xgboost.

[21] G. Hinton and S. Roweis, “Stochastic Neighbor

Embedding," in *Proceedings of the 15th International Conference on Neural Information Processing Systems (NIPS '02)*, Vancouver, Canada, pp. 857-864, January 2002.

심기천(Ki-Chun Sim)

2021년 : 아주대학교 수학과(학사)
2023년 : 아주대학교 일반대학원 지식정보공학과(석사)

2021년~현 재: 아주대학교 대학원 지식정보공학과 석사
※관심분야 : 기계 학습(Machine Learning), 정보보안 (Information Security), 블록체인(Blockchain)

김강석(Kangseok Kim)

2007년 : Indiana University (at Bloomington) 컴퓨터공학과 (공학박사)

2010년~2016년: 아주대학교 대학원 지식정보공학과 연구교수
2016년~현 재: 아주대학교 사이버보안학과 부교수
※관심분야 : 정보보안(Information Security), 딥러닝 응용 보안(Applied Deep Learning for Security)