Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.34, NO.4, Aug. 2024 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2024.34.4.587

웹 방화벽 로그 분석을 통한 공격 분류: AutoML, CNN, RNN, ALBERT*

조 영 복,[1†] 박 재 우,[2] 한 미 란[3‡]
[1,3]고려대학교 (학생, 교수), [2]KAIST (대학원생)

Web Attack Classification via WAF Log Analysis: AutoML, CNN, RNN, ALBERT*

Youngbok Jo,[1†] Jaewoo Park,[2] Mee Lan Han[3‡]
[1,3]Korea University (Undergraduate student, Professor), [2]KAIST (Graduate student)

요 약

사이버 공격, 위협이 복잡해지고 빠르게 진화하면서, 4차 산업 혁명의 핵심 기술인 인공지능(AI)을 이용하여 사이버 위협 탐지 시스템 구축이 계속해서 주목받고 있다. 특히, 기업 및 정부 조직의 보안 운영 센터(Security Operations Center)에서는 보안 오케스트레이션, 자동화, 대응을 뜻하는 SOAR(Security Orchestration, Automation and Response) 솔루션 구현을 위해 AI를 활용하는 사례가 증가하고 있으며, 이는 향후 예견되는 근거를 바탕으로 한 지식인 사이버 위협 인텔리전스(Cyber Threat Intelligence, CTI) 구축 및 공유를 목적으로 한다. 본 논문에서는 네트워크 트래픽, 웹 방화벽(WAF) 로그 데이터를 대상으로 한 사이버 위협 탐지 기술 동향을 소개하고, TF-IDF(Term Frequency-Inverse Document Frequency) 기술과 자동화된 머신러닝(AutoML)을 이용하여 웹 트래픽 로그 공격 유형을 분류하는 방법을 제시한다.

ABSTRACT

Cyber Attack and Cyber Threat are getting confused and evolved. Therefore, using AI(Artificial Intelligence), which is the most important technology in Fourth Industry Revolution, to build a Cyber Threat Detection System is getting important. Especially, Government's SOC(Security Operation Center) is highly interested in using AI to build SOAR(Security Orchestration, Automation and Response) Solution to predict and build CTI(Cyber Threat Intelligence). In this thesis, We introduce the Cyber Threat Detection System by analyzing Network Traffic and Web Application Firewall(WAF) Log data. Additionally, we apply the well-known TF-IDF(Term Frequency-Inverse Document Frequency) method and AutoML technology to classify Web traffic attack type.

Keywords: Web Attack Detection, WAF Log, TF-IDF, AutoML, Machine Learning

Received(03. 19. 2024), Modified(06. 24. 2024), Accepted(06. 28. 2024)

* 이 논문은 2023년도 한국정보보호학회 동계학술대회에 발표한 우수논문을 개선 및 확장한 것임.

* 본 논문은 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임. (No.2021-0-00903, 고신뢰 온-디바이스 딥러닝 가속기 설계를 위한 물리채널 기반 취약점 검증 및 대응기술 개발) 또한, 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구(No. NRF-00252157)이며, 고려대학교에서 지원된 연구비로 수행되었음.

† 주저자, elluardxii12@korea.ac.kr

‡ 교신저자, blosst@korea.ac.kr(Corresponding author)

I. 서 론

최근 사이버 공격, 위협이 고도화되고 복잡해짐에 따라 공격 유형의 다양성도 커지고 있다. 2014년 Sam Musa[1]는 기관, 기업 등 특정 국가, 단체를 목표로 지속적으로 공격하는 지능형 지속 공격(Advanced Persistent Threat, APT)을 정의하였고, 안전한 사이버 보안 환경을 구축하기 위해 APT 공격에 대비해야 함을 강조하였다. 이에 사이버 보안 산업 종사자들은 단계적이고 점진적인 보안 인력풀의 확대와 역량 강화에 많은 노력을 기울이고 있으며, 기업 및 정부 조직의 보안 운영 센터(Security Operations Center, SOC)는 AI를 도입한 보안 관제의 자동화, SOAR 솔루션 기술 구현 등 현업에서 사용가능한 기술 개발에 힘쓰고 있다. 2022년 Igloo Security[2]는 AI와 SOAR 솔루션을 적용한 후, 중•고급 기준 경보 이벤트 처리를 위해 소요되는 시간이 20~30분에서 10초 이내로 줄어드는 것을 확인하였다. 특히, SIEM(Security Information and Event Management)의 경우 일평균 60개의 경보 이벤트를 처리하여 기존보다 20여건 추가 처리가 가능하게 되었음을 확인하였다. 또한, 해당 연구그룹은 인공지능 기반 시스템을 이용하여 비지도학습 기반 탐지 모델을 개발해, 특정 서버로 과다 접속하는 행위를 탐지하여 목적지 서버의 관리자 페이지가 노출된 것을 탐지한 사례도 있다고 밝혔다. 이처럼, 사이버 공격 및 위협 탐지에 AI 기술을 적용하는 사례는 계속해서 증가할 것으로 보인다. 한편, J.Holland et al.[3]은 네트워크 트래픽을 분석하는데 자동화된 머신러닝(AutoML)을 사용하였다. AutoML은 데이터 전처리와 학습 모델 선정을 자동화할 수 있다는 장점을 가지고 있으며, 이를 활용한 연구 결과는 트래픽 악성•정상 분류나 OS Fingerprinting 등 트래픽 분류에 있어 높은 성능을 보이고 있음을 증명하고 있다. 더불어, 네트워크 트래픽, 로그 분석에 있어 AutoML을 사용하는 것이 유용함을 보여줌과 동시에, 보안 관제 기술 자동화에도 긍정적인 영향을 미치고 있다.

본 논문은 웹 방화벽 로그 데이터를 활용하여, 머신러닝 알고리즘으로 공격 유형을 식별하는 방법을 제시한다. 본 연구는 두 가지의 실험 결과에 초점을 맞추었는데, 첫 번째는 TF-IDF, SVD를 이용하여 전처리한 데이터를 AutoML, CNN, RNN에 적용한 결과를 분석한다. 두 번째는 NLTK를 사용하여 데이터를 전처리하는 ALBERT 모델의 결과를 분석한다. TF-IDF는 전통적인 텍스트 마이닝 기법으로 단어의 중요도를 평가하는 기술인 반면, BERT는 문맥을 고려한 단어의 의미 파악에 초점을 맞춘 기술이다. 우리는 두 기술의 차이점에 주목하였고 이것이 모델의 성능에 어떤 영향을 미치는지 비교 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 AutoML 기술과 AutoML을 이용해 네트워크 트래픽의 악성•정상 분류에서 유효성을 증명한 연구 결과를 소개하고, 웹 방화벽 로그 데이터를 분석한 연구를 소개한다. 3장에서는 본 연구에서 데이터를 전처리할 때 사용한 TF-IDF, SVD와 ALBERT 모델에서 사용하는 NLTK를 상세하게 설명한다. 4장에서는 데이터셋 소개 및 실험 방법과 결과를 서술하며, 마지막으로 5장에서 결론과 향후 연구 방향을 제시하며 마무리한다.

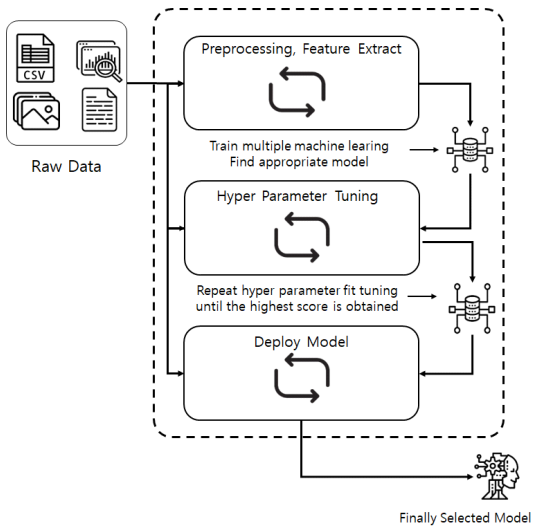

Fig. 1. Automated Machine Learning Process [4]

II. 관련 연구

2.1 자동화된 머신러닝 기술(AutoML)

D.Xin et al.[5]은 Open Source-Software (OSS), Cloud Provider Solutions 등 현재 사용되고 있는 AutoML 시스템에 대해 소개하고, 각 시스템의 장, 단점을 비교 분석 하였다. 특히, OSS 기반 시스템에 대해, Python과 같은 프로그래밍 코드에 라이브러리(Library)로 제공하여 전처리

(Pre-Processing) 단계에서 사용하기 쉽다는 장점과, 모델 성능 평가 등 후처리(Post-Processing) 단계에 대한 기능 지원이 미흡하다는 단점을 서술하고 있다. OSS 기반 시스템의 종류로는 Scikit-Learn을 기반으로 한 Auto-Sklearn[6], Keras를 기반으로 한 AutoKeras[7], Tensorflow를 기반으로 한 AdaNet[8] 등이 사용되고 있다.

2020년 N.Erickson el al.[9]은 OSS 기반 AutoML Framework인 Autogluon-Tabular 모델을 제시하였다. N.Erickson 연구그룹에 따르면, Autogluon-Tabular 모델은 best modeling practices를 통해 데이터 분석 및 분류에서 각 모델별 우수한 성능을 보여주고, CSV 파일 데이터셋 분석에서 다양한 옵션을 제공해 편리함을 가진다. 이처럼, 다양한 AutoML System 개발과 관련된 연구가 다수 등장하였고, AI를 활용하는 다양한 분야에서 AutoML을 사용하고자 하는 시도가 많아질 것으로 예상된다.

2.2 AutoML을 활용한 네트워크 트래픽 분석 연구

네트워크 악성 공격 분류에서도 AutoML을 이용해 높은 성능을 보여준 연구가 등장하였다. J.Holland et al.[3]은 네트워크 트래픽의 악성과 정상을 분류하는 데 있어 패킷을 표현하는 새로운 방식인 nPrint를 제시하였다. 자동화된 기계 학습 파라미터 튜닝 기법인 AutoML의 Autogluon-Tabular 모델을 이용하여 다양한 트래픽 분석 태스크 학습하는 시스템을 개발하였다. 이를 nPrintML이라고 하며, OS Fingerprinting, IoT 네트워크 트래픽 분류 등 여러 분야의 트래픽 데이터셋 분류에 있어 높은 정확도를 보인다고 서술하고 있다.

Jaewoo Park, Minsu Kim, Heejun Roh [10]은 nPrint, nPrintML이 TLS(Transport Layer Security) 기반 암호화된 악성 트래픽의 분류에 있어 유효성을 보이는 것을 검증하였다. 해당 연구그룹은 암호화된 악성 트래픽을 탐지, 분류할 때 암호화 되지 않은 정보를 추출하거나, 암호화 트래픽을 복호화 하지 않고도 암호화된 악성 트래픽을 탐지, 분류하는 것이 가능함을 제시하였다. 또한, 네트워크 트래픽을 탐지, 분석하는데 있어서 AutoML을 사용하는 것이 유의미함을 보여주었다.

2.3 웹 방화벽 로그(WAF) 분석과 공격 유형 탐지

A.Razzaq et al.[11]은 웹 해킹 공격의 빈도가 증가함에 따라 웹 방화벽(WAF)을 사용하는 것의 중요성을 서술하였고, WAF 로그 데이터를 분석하여 악성 공격을 탐지하는 연구를 수행하는 것의 중요성을 강조하였다. Y.Gao et al.[12]은 웹 로그 데이터에서 이상징후 탐지를 수행하기 위해 우선적으로 4가지의 Feature(Web Resource, Status Code, Content Length, Referrer)를 추출하였고, K-means Clustering을 이용해 웹 트래픽의 정상/악성 분류를 높은 정확도로 수행하였다. 그러나, 웹 해킹 공격 페이로드의 길이는 증가하고 암호화된 트래픽의 페이로드가 복잡해짐에 따라 기존 연구 방법으로는 악성 웹 트래픽 로그를 탐지하고 분류하는 것이 어려워졌다.

J.Zhan et al.[13]은 문자 간 분리, 바이트 쌍 인코딩(BPE), TF-IDF를 융합한 특징 추출 방법을 제안하였다. TF-IDF는 각 단어의 출현 빈도와 문서 내에서의 상대적 중요성을 고려한 값으로써, HTTP Request에서 토큰의 중요도를 보다 정밀하게 평가하는데 사용된다. 문자 분리, BPE를 통해 토큰화된 각 HTTP Request에서 TF-IDF 적용은 Feature 추출을 가능케 하였고, 복잡한 웹 로그 데이터를 분석하는데 있어 중요한 방법을 제시했다고 할 수 있다.

III. 전처리 및 피쳐구성

3.1 데이터 전처리

웹 로그 데이터는 다양한 형식과 복잡한 구조를 가지고 있으며, 이를 머신러닝 또는 딥러닝 알고리즘에 입력 데이터로 사용하기 위해서는 가장 먼저 데이터 전처리가 필요하다. 본 연구에서는 다음 세 가지 방법을 이용해 데이터 클리닝을 진행하였다.

- ***대소문자 통일***: 모든 페이로드의 대소문자를 통일하여 일관성을 보장하고, 분류의 유연성을 증가시켰다.
  예) "GET /index.html HTTP/1.1" → "get /index.html http/1.1"

- ***URL인코딩 및 HTML이스케이프 문자 복원***: 페이로드 내 URL 인코딩 및 HTML 이스케이프 문자를 복원하여 모델이 데이터를 올바르게 해석할 수 있도록 하였다. 예를 들어, URL 인코딩인 "%20"은 공백 문자로 치환하여 페이로드에서 공백 처리를 진행하였다.
  예) %20″ → ″ ″ (공백)
  예) ″<script>″ → ″〈script〉“

- ***NUL 문자, 중복 데이터 제거***: 문자열의 종료를 나타내는 NUL(NULL) 문자를 빈 문자열로 변환하였고, CSV 파일 내 중복 데이터를 제거하였다.
  예) ″alert(′test′)\0″ → ″alert(′test′)″

위 과정을 통해 데이터 전처리를 수행하여, 데이터 복잡성을 감소시키고 중요한 패턴이나 정보를 명확하게 추출할 수 있었다.

3.2 TF-IDF(Term Frequency-Inverse Document Frequency)

먼저, TF-IDF 기법을 통해 데이터 변환을 수행하였다. TF-IDF는 문서 내에서 단어의 빈도(Term Frequency)와 문서 집합 내에서 단어의 상대적 빈도(Inverse Document Frequency)를 이용하여 단어의 중요도를 파악하는 방법이다. 이는 아래 세 가지 수식으로 표현된다.

- **Term Frequency(TF)**: 특정 단어가 문서 내에서 얼마나 자주 등장하는지를 나타내는 지표로, 해당 단어의 문서 내 빈도수를 전체 단어의 총 빈도수를 나눈 값이다. t는 단어, d는 문서, f(f,d)는 단어 t가 문서 d에서 등장하는 빈도수, |d|는 문서 d의 총 단어 수이다. 수식은 다음과 같다.

$$TF(t,d) = \frac{f_{t,d}}{\Sigma_{t' \in d} f_{t',d}} \quad (1)$$

- **Inverse Document Frequency(IDF)**: 특정 단어가 문서 집합 내에서 얼마나 드물게 등장하는지를 나타내는 지표로, 해당 단어가 등장하는 문서의 수를 전체 문서의 수로 나눈 값의 로그 값이다. D는 문서 집합, |D|는 문서 집합의 총 문서 수, |{d in D | t in d}|는 단어 t가 등장하는 문서의 수이다. 수식은 다음과 같다.

$$idf(t,D) = \log \frac{N}{|d \in D : t \in d|} \quad (2)$$

- **TF-IDF**: TF와 IDF를 결합한 값으로, 문서 내에서 단어 중요도를 파악한다. 수식은 다음과 같다.

$$TFIDF(t,d,D) = TF(t,d) \cdot idf(t,D) \quad (3)$$

본 연구에서는 scikit-learn의 TFidfVectorizer를 활용하여 TF-IDF 벡터화를 진행하였다. 단어 단위에만 초점을 맞추어 벡터화를 진행하였으며, 이를 통해 로그 데이터의 단어들을 TF-IDF 가중치로 변환하여 특징을 추출하였다. 로그 데이터에서 특정 용어 또는 패턴의 중요도를 파악할 수 있다.

3.3 SVD(Singular Value Decomposition)

TF-IDF 적용한 데이터는 원본 데이터보다 더 높은 차원을 가지다보니, 분석을 수행함에 있어 다소 어려움이 따른다. 이를 해결하기 위해 고차원을 저차원으로 변환해주는 차원축소를 수행하였다. 본 연구에서는 차원을 축소하는 방법으로 SVD(Singular Value Decomposition)를 활용한다. SVD는 데이터를 주성분(principal component)으로 분해하여 차원을 축소하는 기법이며, 다음과 같은 수식(4)으로 표현된다.

$$A = U \Sigma V^T \quad (4)$$

- 여기서 A는 $m \times n$ 크기의 데이터 행렬을 나타낸다. U와 V는 각각 $m \times m$과 $n \times n$ 크기의 직교행렬이며, Σ는 $m \times n$ 크기의 대각 행렬(특이값이 대각선 상의 위치)을 나타낸다. TF-IDF를 통해 변환된 데이터에서 SVD를 적용함으로써, 데이터의 중요한 feature를 추출하였다. 데이터를 저차원으로 변환함으로써 데이터의 크기를 줄임과 동시에 중요한 정보를 유지할 수 있게 되었다. 이는 AutoML 및 딥러닝 모델의 연산 복잡도를 낮추고 학습 속도를 향상시키는 장점을 가진다.

3.4 NLTK를 활용한 ALBERT(A Lite Bidirectional Encoder Representations from Transformer) 모델 학습

NLTK는 자연어 처리 작업을 수행할 수 있는 파이썬 라이브러리로, 본 연구에서는 웹 방화벽 로그의 공격 payload를 데이터를 전처리하였다. NLTK에서 지원하는 기능 중 토큰화, 어근 추출, 불용어 제거 기능을 사용하여 payload 데이터를 전처리하였고, 학습 모델은 ALBERT 모델을 선정하였다. 전처리된 'payload' 데이터는 ALBERT를 사용하여 공격 패턴을 예측하였다. ALBERT는 구글의 BERT 모델을 기반으로 하여, 'Masked Language Model', 'Next Sentence Prediction' 등의 방법을 통해 대규모 텍스트 데이터를 학습하고 문맥을 이해하여 자연어 처리 작업을 수행하는 경량화된 모델이다. NLTK를 이용해 전처리된 데이터를 대상으로 하여, 본 논문에서는 ALBERT 모델을 이용해 우리의 데이터를 학습시켰을 때 분류 성능에 대해 분석한다.

IV. 방법 및 결과

4.1 데이터셋

본 연구에서는 2023년 사이버보안 AI 빅데이터 챌린지에서 배포한 웹 방화벽 로그 데이터와 CSIC2012 데이터셋 중 공격 데이터를 활용하였다[14], [15]. 두 데이터셋은 서로 다른 형태와 특징을 가지고 있지만, payload를 공통된 feature로 가지고 있어 이를 활용하여 지도 학습을 수행하였다. 사이버보안 AI 빅데이터 챌린지 로그 데이터는 각각 45,000개의 데이터로써, Log_Number, payload, Label_Action의 세 가지 컬럼으로 되어있으며, 8가지의 공격 유형을 포함한다. 그러나, 데이터셋 분석 과정에서 중복데이터가 상당히 많았으며, 같은 공격이 여러번 기록되었음에도 불구하고 다른 Label_Action으로 분류된 경우가 있었다. 이러한 경우 학습 모델의 정확도에 부정적인 영향을 미칠 수 있기 때문에 중복 데이터를 제거하고 일관된 Label_Action을 부여하는 전처리 과정을 수행하였다. 반면, CSIC2012 데이터셋은 네트워크 공격 식별에 특화된 데이터셋으로, 웹 페이로드로 재구성된 49,311개의 로그 데이터로, 사이버 보안 AI 빅데이터 챌린지 데이터셋과 유사한 형태로 재구성되었다. 재구성된 공격 유형은 XXS, XPath, SSI, SQLi, LDAPi, FormatString, CRFi, BufferOverflow인 8개의 공격 라벨로 본 연구에 사용하기 위해 비슷한 공격 유형을 가진 데이터를 Cross Site Scripting, SQL Injection, System Command Execution으로 재분류하였다. 이 과정을 거쳐 우리는 최종적으로 52,722개의 데이터를 확보하였고, payload를 학습 Feature로 사용, Label Action을 학습 Label로 지정하여 지도 학습을 수행하였다. [표 1]은 Label Action에 따른 데이터의 개수를 나타낸 것이다.

Table 1. Web Attack Payload Dataset

Label Action	Count
SQL Injection	21227
Vulnerability Scan	9969
System Cmd Execution	8229
Cross Site Scripting	5222
Path Disclosure	3796
HOST Scan	2201
Cross Site Scripting	1036
Automatically Searching Information	667
Directory Indexing	247
Leakage Through NW	128

4.2 AutoML을 통한 웹 공격 분류

AutoML은 데이터 전처리, 모델 선택, 하이퍼파라미터 조정 등의 과정과 최적화를 자동으로 수행하는 기술이다. 웹 공격 페이로드 분류 과정은 AutoML을 활용하여, 3절에서 설명된 전처리 방법을 로그 데이터에 적용함으로써 시작한다. 이 과정에서는 AutoML의 모듈인 TabularPredictor를 사용해 여러 머신러닝 알고리즘을 수행하였다. 그 결과, WeightedEnsemble_L2 모델 94.80%의 높은 정확도를 달성하였다. 또한, 다른 머신러닝 알고리즘들을 비교했을 때, [표 2]처럼 Tree 계열 알고리즘에서 비교적 높은 정확도를 보이고 있음을 확인할 수 있다.

Table 2. Performance and time comparison of various machine learning models in AutoML using the proposed preprocessing

Model	Accuracy (train)	Accuracy (valid)	Prediction Time (test)/min	Prediction Time (valid)/min	Learning time/min
RandomForestGini	0.9745	0.8600	0.6555	0.2482	7.9848
ExtraTreesGini	0.9744	0.8592	0.6372	0.2162	2.6157
ExtraTreesEntr	0.9744	0.8592	0.7280	0.2556	2.2242
RandomForestEntr	0.9744	0.8576	0.5910	0.2183	16.3105
KNeighborsDist	0.9737	0.8484	1.6658	0.1932	0.1820
WeightedEnsemble_L2	0.9507	0.8760	16.4137	1.5983	345.5796
LightGBM	0.9480	0.8608	0.7474	0.0461	13.1548
LightGBMLarge	0.9473	0.8608	0.5976	0.0304	18.7651
XGBoost	0.9444	0.8648	0.7330	0.0615	14.2201
LightGBMXT	0.9254	0.8596	0.9078	0.0516	14.8419
CatBoost	0.9190	0.8624	0.1558	0.0233	92.8109
NeuralNetTorch	0.8975	0.8692	11.5835	0.7691	155.3104
NeuralNetFastAI	0.8926	0.8688	0.8345	0.0616	72.4990
KNeighborsUnif	0.8845	0.8384	1.7069	0.2721	0.1736

4.3 CNN, RNN 구현 및 실험

본 절에서는 딥러닝을 활용한 웹 공격 분류 모델인 CNN과 RNN의 구현 및 실험 결과를 기술한다. 각 모델의 입력은 3절에서 언급한 전처리된 데이터를 기반으로 적용하였으며, 학습용과 테스트용 데이터셋은 8:2의 비율로 분할하였다. 클래스 간의 비율을 유지하기 위해 Scikit-learn의 stratify 옵션을 사용하였다.

딥러닝 알고리즘으로는 CNN과 RNN 모델을 구현하였으며, Pytorch 프레임워크를 사용했다. 모델의 학습에는 Stochastic Gradient Descent (SGD) 최적화 알고리즘을 사용한다. 이 알고리즘은 모델의 파라미터를 최적화하여 정확도를 구한다.

학습 데이터는 TensorDataset과 DataLoader를 사용하여 배치 크기를 20으로 설정하였고, 모델의 출력과 실제 라벨 간의 차이는 Cross Entropy 손실 함수로 측정하였다. 각 epoch마다 학습 데이터와 검증 데이터에 대한 손실 및 정확도(f1-macro)를 기록하였다. 과적합을 방지하기 위해 검증 손실이 더 이상 개선되지 않을 경우 조기 종료 기능을 구현하였다. CNN 모델은 Conv1d 연산을 통해 1차원 합성곱 레이어를 구성하였다. 합성곱 레이어는 입력 데이터의 차원을 1로 설정하고, 출력 채널을 120으로 설정하였다. 이후 Linear 연산을 사용하여 완전 연결 레이어를 구성했으며, 완전 연결 레이어에서는 클래스의 개수와 동일한 수의 출력 뉴런을 설정했다. RNN 모델은 입력 크기, 숨겨진 유닛의 크기, 레이어의 수, 그리고 출력 클래스의 수를 매개변수로 받는다. RNN 레이어에서는 초기 은닉 상태를 0으로 설정하고, 이후 순환 연산을 통해 각 시간 단계에서의 은닉 상태를 업데이트한다. Fully Connected Layer는 RNN 레이어로부터의 출력을 입력받아 최종 분류 결과를 도출한다. 실험 결과, CNN, RNN 모델과의 학습 결과는 〔표 3〕과 〔표 4〕와 같다. 두 모델 모두 학습을 진행하면서 검증 손실이 감소하고 검증 정확도가 증가하는 것을 확인할 수 있다.

학습 결과를 바탕으로, CNN과 RNN 모델은 웹 공격 분류에 효과적으로 사용할 가능성을 확인하였다. 단, 추후 연구에서는 모델의 성능을 더욱 개선하

Table 3. Learning results of CNN model

epoch	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
1	1.5706	1.2352	0.4612	0.5993
10	0.5847	0.5755	0.7677	0.7711
20	0.4711	0.4678	0.8146	0.8156
30	0.4269	0.4264	0.8278	0.8297
40	0.4038	0.4055	0.8355	0.8367

Table 4. Learning results of RNN model

epoch	Train Loss	Validation Loss	Train Accuracy	Validation Accuracy
1	1.4706	1.1201	0.4932	0.6251
10	0.4578	0.4555	0.8171	0.8193
20	0.3361	0.3525	0.8696	0.8653
30	0.2806	0.3281	0.8888	0.8774
40	0.2507	0.3070	0.9004	0.8864

기 위해 하이퍼파라미터 튜닝 및 다른 딥러닝 알고리즘 적용을 논의 중이다.

4.4 ALBERT(A Lite BERT)를 통한 웹 공격 분류

ALBERT 모델 학습을 위해선 자연어 처리를 통한 전처리가 필요하다. 웹 공격 분류를 위해서 데이터의 URL 인코딩된 문자열의 디코딩, HTML 태그 제거 작업, NLTK 라이브러리를 활용하여 불용어 처리 등의 페이로드의 핵심 단어들이 두드러지도록 전처리하였다. 전처리된 웹 로그 데이터를 ALBERT 모델은 각 로그를 고차원 임베딩 벡터로 변환한다. 이 임베딩 벡터는 웹 로그의 문맥적 의미를 띄고, 이를 기반으로 웹 공격을 분류하는 데 사용된다. ALBERT 모델의 출력은 Softmax 함수를 거쳐 각 클래스에 대한 확률값으로 변환된다. 이 확률값이 가장 높은 클래스가 해당 웹 로그의 예측된 공격 유형이 된다. 모델의 성능 평가는 교차 엔트로피 손실 함수를 사용하여 진행하였다. 〔그림 2〕는 AutoML과 ALBERT 모델을 통해 웹 공격 분류를 수행한 결과이다.

4.5 실험 평가

본 절에서는 데이터 전처리 기법과 머신러닝 알고리즘을 적용한 실험 결과를 종합적으로 분석한다. 실험에서는 3.1절에서 소개한 데이터 전처리 과정을 거친 데이터와 거치지 않은 데이터를 사용하여, TF-IDF 피처와 TF-IDF와 SVD를 결합한 피처를 생성하고, 이를 Autogluon의 머신러닝 알고리즘으로 학습시켰다. 또한, 데이터 전처리를 거친 딥러닝

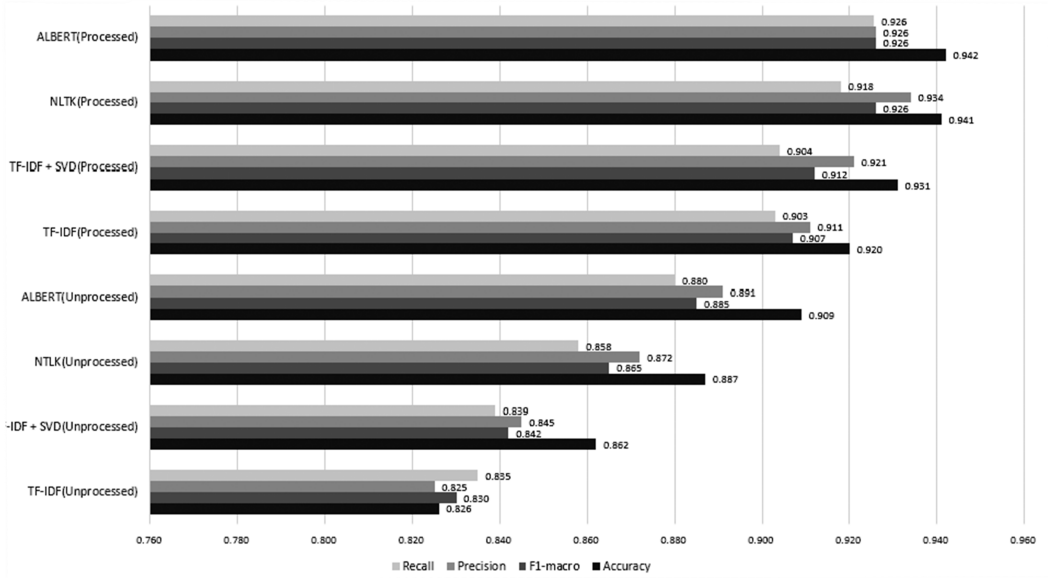

Fig. 2. Performance Evaluation of Data Preprocessing and Machine Learning/Deep Learning Models

알고리즘(CNN, RNN, ALBERT) 모델 중 어떤 알고리즘이 가장 높은 정확도를 보이는지 실험하였다. 웹 공격 분류 정확도, F1-매크로, Recall 기준으로 데이터 전처리 전후 및 TF-IDF와 SVD의 피처 결합 효과를 분석하였다. 실험 결과는 〔표 4〕에 요약되어 있다.

3.1절의 전처리 과정을 거치지 않은 경우에도 비교적 준수한 성능을 보였으나, 중복 데이터 빈도, 특수 문자, 불용어 등에 노출되어 모델 성능에 부정적인 영향을 미쳤다. 전처리 후 불용어를 제거한 정제된 데이터에 TF-IDF를 적용한 결과, 정확도가 향상되었다. 3.1절의 데이터 전처리 없이 단독으로 사용한 TF-IDF와 비교했을 때, SVD와 결합할 경우 전처리 전에는 성능이 떨어지지만, 전처리 후에는 유사한 성능을 보였다. SVD는 TF-IDF의 불필요한 정보를 줄이고 중요한 정보를 추출하는 데 도움을 줄 수 있으나, 학습 속도를 고려할 때 전처리 과정이 더욱 중요함을 나타낸다. 이러한 이유로 ALBERT 모델에서는 3.1절의 데이터 전처리와 NLTK로 불용어를 처리했다. 또한, WordPiece로 페이로드를 토크나이징하고 ALBERT 모델에 입력했다. 이러한 요소를 제거한 웹 페이로드의 문맥적 의미를 잘 반영하여, 단순한 단어 빈도 기반의 TF-IDF보다 더 나은 성능을 보였다.

V. 결 론

웹 로그 데이터는 다양한 형식과 복잡한 구조를 가지고 있어 이상 징후 탐지를 위한 데이터 분석에 매우 까다로운 특징을 가지고 있다. 본 연구에서는 머신러닝이나 이러한 문제를 해결하기 위해 데이터 클리닝 수행 및 세밀한 데이터 전처리 과정을 거쳐 머신러닝, 딥러닝 알고리즘에 적용하였다. 사용한 하드웨어 장비의 사양은 다음과 같다. CPU: Intel(R) Xeon(R) Silver 4210, RAM: 400 GB, GPU: Telsa P40 TF-IDF, SVD를 이용하여 전처리한 데이터를 AutoML, CNN, RNN에 적용한 결과와 NLTK를 사용하여 데이터를 전처리하는 ALBERT 모델의 결과를 비교 분석하였다. TF-IDF와 AutoML을 이용하였을 때, Closed-World 환경에서 약 95%의 정확도를, Open-World 환경에서 약 91%의 f1-macro로 유의미함을 확인할 수 있었다. Closed-World 환경은 이미 알려진 공격 패턴만을 고려하는 환경이며, Open-World 환경은 알려지지 않은 공격 패턴도 고려하는 환경을 의미한다.

본 연구를 통해 웹 방화벽 로그 분석에 있어 ALBERT 모델이 가장 높은 성능을 보임을 확인하였다. ALBERT 모델은 성능 면에서 매우 뛰어나지만, 높은 자원 소모와 느린 학습 속도라는 한계가 있다. 반면에, TF-IDF와 SVD를 병합한 모델은 효율적인 선택이 될 수 있다. 최종적으로, 웹 방화벽 로그 분석을 위해 BERT 기반 모델이 가장 적합하며, 특히 ALBERT 모델이 최적의 선택이 될 수 있지만, 상황에 따라 학습 속도가 빠른 TF-IDF와 SVD를 사용한 AutoML 모델도 고려해야 한다. CNN과 RNN 모델도 분석에 포함되었으나, ALBERT 모델에 비해 성능이 낮았다. 하지만 이들 모델은 상대적으로 가벼운 자원 소모와 빠른 학습 속도를 제공하므로, 실시간 분석이 필요한 상황에서 유용할 수 있다.

향후 연구에서는 ALBERT 모델의 자원 소모를 줄이기 위한 최적화 방안과 TF-IDF + SVD 모델의 성능을 더욱 향상시킬 수 있는 방법에 대한 추가 연구가 필요할 것이다. 이러한 연구 결과는 웹 로그 데이터를 이용한 공격 분류 및 이상 징후 탐지 기술의 발전에 기여할 것으로 기대된다. 향후 연구에서는 다양한 웹 로그 데이터를 수집하고, 이를 더욱 세밀하고 다양한 전처리 과정을 거쳐 분석하는 것이 필요하다. 이를 통해 분류 및 이상 징후 탐지의 정확도와 신뢰도를 더욱 향상시킬 수 있을 것이다.

References

〔1〕 Musa, D.S, "Advanced Persistent Threat-APT." https://www. academia. edu/6309905/Advanced_Persistent_Threat_-_APT, 2014.

〔2〕 Igloo Security, "Example of control application using security orchestration", July. 2022

〔3〕 J. Holland, P. Schmitt, N. Feamster, and P. Mittal, "New directions in automated traffic analysis," Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 3366-3383, Nov. 2021.

〔4〕 A. Brown, M. Gupta, and M.

Abdelsalam. "Automated machine learning for deep learning based malware detection." Computers & Security vol. 137 p.103582, 2024.

[5] D. Xin, E.Y. Wu, D.J.L. Lee, N. Salehi, and A. Parameswaran, "Whither automl? understanding the role of automation in machine learning workflows," Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, pp. 1-16, May 2021.

[6] M. Feurer, A. Klein, K. Eggensperger, J. Springenberg, M. Blum, and F. Hutter, "Efficient and robust automated machine learning," Advances in Neural Information Processing Systems, vol. 28, 2015.

[7] H. Jin, Q. Song, and X. Hu, "Auto-keras: An efficient neural architecture search system," Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 1946-1956, July 2019.

[8] C. Cortes, X. Gonzalvo, V. Kuznetsov, M. Mohri, and S. Yang, "Adanet: Adaptive structural learning of artificial neural networks," International conference on machine learning, pp. 874-883, July 2017.

[9] N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola, "Autogluon-tabular: Robust and accurate automl for structured data," arXiv preprint arXiv: 2003. 06505, 2020.

[10] Jaewoo Park, Minsu Kim, and Heejun Roh, "On the Effectiveness of nPrint to an Encrypted Malware Traffic Dataset," Proceedings of the Korean Information Science Society Conference, pp. 1938-1940, 2023.

[11] A. Razzaq, A. Hur, H.F. Ahmad, and M. Masood, "Cyber security: Threats, reasons, challenges, methodologies and state of the art solutions for industrial applications," Proceedings of the 2013 IEEE Eleventh International Symposium on Autonomous Decentralized Systems (ISADS), pp. 1-6, March 2013.

[12] Y. Gao, Y. Ma, and D. Li, "Anomaly detection of malicious users' behaviors for web applications based on web logs," Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT), pp. 1352-1355, Oct. 2017.

[13] J. Zhan, X. Liao, Y. Bao, L. Gan, Z. Tan, M. Zhang, ... & J. Lu, "An effective feature representation of web log data by leveraging byte pair encoding and TF-IDF," Proceedings of the ACM Turing Celebration Conference-China, pp. 1-6, May 2019.

[14] KISA, "Cyber security AI Big Data Challenge 2023, A track," https://ai-bigdatasec.kr/, Oct. 2023.

[15] Torpeda, "CSIC2012 Dataset(Attacks)", ISI-CSIC, https://www.tic.itefi.csic.es/torpeda/datasets.html, Sept. 2012.

〈저자소개〉

조 영 복 (Youngbok Jo) 학생회원
2024년 8월: 고려대학교 인공지능사이버보안학과 수료
2024년 5월: KISTI 한국과학기술정보연구원 과학기술보안연구센터 융합보안연구팀 재직
〈관심분야〉 AI 보안, 네트워크 보안, 침해사고 대응

박 재 우 (Jaewoo Park) 학생회원
2024년 2월: 고려대학교 인공지능사이버보안학과 졸업
2024년 3월~현재: KAIST 정보보호대학원 석사과정
〈관심분야〉 네트워크 보안, 침해사고 대응, 보안관제

한 미 란 (Mee Lan Han) 종신회원
2002년 2월: 동덕여자대학교 컴퓨터과학과 졸업
2004년 8월~2012년 3월: ㈜넥슨 메이플스토리 해외사업본부 책임연구원
2015년 8월: 고려대학교 정보보호대학원 석사 졸업
2020년 8월: 고려대학교 정보보호대학원 박사 졸업
2020년 9월~2021년 8월: 고려대학교 정보보호연구원 연구교수
2021년 9월~2022년 8월: 고려대학교 인공지능사이버보안학과 산학협력중점교수
2022년 9월~현재: 고려대학교 인공지능사이버보안학과 조교수
〈관심분야〉 이상징후탐지 및 식별, CTI 연구, 사이버범죄 행위분석, 물리계층보안