Check for updates

Journal of KIIT. Vol. 22, No. 2, pp. 1-8, Feb. 28, 2024. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2024.22.2.1

데이터 전처리를 간소화한 자연어처리 기반의 악성코드 탐지모델

신동환*, 김한석**, 이수진***

Malware Detection Model-based on Natural Language Processing with Simplified Data Preprocessing

Dong-Hwan Shin*, Han-Seok Kim**, and Soo-Jin Lee***

요 약

머신러닝 혹은 딥러닝 모델을 학습시켜 악성코드 탐지에 활용하기 위해서는 모델 학습 이전에 데이터 전처리가 반드시 필요하다. 그리고 새로운 악성코드가 나타나면 전처리 과정을 업데이트하고 다시 모델에 적용해야 한다. 이러한 비효율성을 해결하기 위해 본 논문에서는 자연어처리 기반의 악성코드 탐지모델을 제안하였다. CIC-MalMem-2022 데이터세트가 가지는 55개의 속성을 공백으로 구분된 하나의 문장으로 병합한 후, 대표적인 Transformer 모델인 BERT, DistilBERT 및 BART에 학습시키고 분류성능을 분석하였다. 그 결과 이진 분류에서는 기존 연구들과 유사한 탐지성능을 보였고, 다중 분류에서도 84.69%, 85.10%, 84.44%의 정확도를 달성하여 전처리 과정을 간소화하고도 탁월한 성능을 발휘할 수 있음을 확인하였다.

Abstract

In order to train a machine learning or deep learning model to detect malware, it is essential to preprocess the data before training the model. And when new malware appears, the preprocessing process must be updated and applied to the model again. To solve these inefficiencies, this paper proposes a malware detection model based on natural language processing. After merging 55 attributes of the CIC-MalMem-2022 dataset into a single sentence separated by spaces, we trained the Transformer model, BERT, DistilBERT, and BART, and analyzed the classification performance. As a result, the detection performance of binary classification was similar to that of previous studies, and the accuracy of multi classification was 84.69%, 85.10%, and 84.44%, confirming that it can achieve excellent performance with a simple preprocessing process.

Keywords

preprocessing, transformer, BERT, DistillBERT, BART, natural language processing, CIC-MalMem-2022

* 해병대 연평부대 정보통신과장
- ORCID: https://orcid.org/0009-0000-4257-1184
** 육군본부 지능정보기술단 작전소프트웨어개발장교
- ORCID: https://orcid.org/0009-0008-2365-4230
*** 국방대학교 국방과학학과 교수(교신저자)
- ORCID: https://orcid.org/0000-0002-4117-407X

· Received: Nov. 08, 2023, Revised: Dec. 14, 2023, Accepted: Dec. 17, 2023
· Corresponding Author: Soo-Jin Lee
Dept. of Defense Science, Korea National Defense University, 1040 Hwangsanbeol-ro, Yangchon-myeon, Nonsan-si, Chungcheongnam-do, Korea
Tel.: +82-41-831-5378 Email: cyberkma@gmail.com

Ⅰ. 서 론

악성코드는 점점 더 지능적으로 진화하면서 정교해지고 있어 기존 탐지기법들만으로는 탐지가 쉽지 않은 상황에서 악성코드 탐지기법의 개선 필요성이 증대되고 있다. 이에 최근에는 악성코드 탐지성능을 향상시키기 위해 인공지능 기술을 적용하는 시도가 활발하게 진행되고 있다.

그러나 머신러닝 혹은 딥러닝 기법을 적용하여 악성코드를 탐지를 시도한 대부분의 연구는 메타데이터를 활용하고 있다. 때문에 탐지모델을 생성하기 위해서는 학습시킬 데이터를 분석한 후 전처리하는 과정이 필수적이다. 즉 범주형(Categorical) 데이터를 수치형(Numeric) 데이터로 변환하는 원-핫 인코딩(One-hot encoding)이나, 학습 속성의 범위가 달라 학습이 왜곡되는 것을 방지하기 위한 표준화 혹은 정규화 등을 반드시 수행해야 한다. 그리고 이러한 전처리 과정은 데이터에 종속적이기 때문에, 추가로 새로운 악성코드 데이터가 출현하면 전처리 과정을 업데이트 후 모델에 다시 학습시켜야 하는 번거로움이 발생한다.

이에 본 논문에서는 이러한 데이터 전처리의 비효율성을 극복하기 위해 여러 속성을 하나의 문장으로 인식하게 하여 모델에 학습시킨 후 악성코드를 탐지하는 자연어 처리[1] 기반의 악성코드 탐지기법을 제안하였다. 제안된 기법의 효율성을 검증하기 위해서 문장 분류에 강점을 보이는 대표적인 Transformer 모델 3종(BERT(Bidirectional Encoder Representations from Transformers)[2], DistilBERT[3] 및 BART(Bidirectional and Auto-Regressive Transformers)[4])을 활용하여 악성코드 탐지모델을 생성하였다. 그리고 메모리 분석을 기반으로 난독화된 악성코드를 탐지하는데 활용하기 위해 생성된 CIC-MalMem-2022 데이터세트[5]를 대상으로 탐지성능을 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 실험에 사용된 CIC-MalMem-2022 데이터세트에 대해 설명하고, 선행연구를 정리한다. 3장에서는 제안하는 악성코드 탐지모델의 구축절차와 실험 방법을 설명하고, 실험 결과를 분석한다. 마지막으로 4장에서 연구결과를 요약하고 결론을 맺는다.

Ⅱ. 관련 연구

2.1 데이터 전처리의 문제점

서론에서 전술한 바와 같이 메타데이터를 활용해 악성코드 탐지모델을 생성하기 위해서는 그림 1과 같은 복잡한 전처리 과정을 반드시 거쳐야만 한다. 원-핫-인코딩 및 정규화 등의 전처리를 통해 메타데이터를 새로운 특성(Feature) 세트로 변환해야 모델이 올바르게 학습을 실시하고 분류 등의 동작에서 원하는 결과를 달성할 수 있기 때문이다.

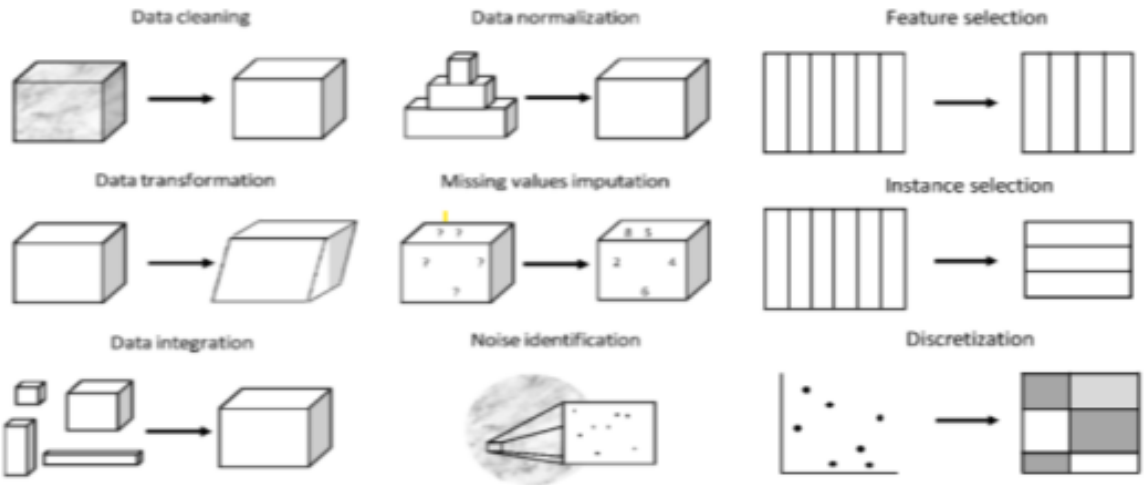

그림 1. 데이터 전처리[6]
Fig. 1. Data preprocessing[6]

그러나 데이터 전처리는 상당한 시간이 필요한 작업이다. 포브스(Forbes)의 발표자료[7]에 따르면, 데이터 전처리 과정에서 소요되는 시간이 데이터의 처리를 완료하는데 소요되는 전체 시간의 60%를 차지한다. 따라서 새로운 학습데이터를 탐지모델에 추가로 학습시키기 위해 전처리 과정을 반복하는 것은 지극히 비효율적이다.

2.2 CIC-MalMem-2022 데이터세트

CIC-MalMem-2022 데이터세트는 메모리 분석을 통해 난독화된 악성코드의 탐지를 테스트하기 위해 생성되었으며, 실제로 흔히 발견되는 악성코드를 사용하여 발생할 수 있는 실제 상황을 잘 표현할 수 있도록 제작되었다.

데이터세트는 29,298개의 악성코드 메모리 덤프(Dump)와 29,298개의 정상행위 메모리 덤프를 포함 총 58,596개의 데이터로 구성되어 있으며, 데이터세트에 포함된 악성코드의 유형별 세부 데이터 개수는 표 1에서 보는 바와 같다[8].

표 1. CIC-MalMem-2022 dataset 구성
Table 1. Configuration CIC-MalMem-2022 dataset

Category	Families	Count
Benign	-	29,298
Trojan horse (9,487)	Zeus	1,950
	Emotet	1,967
	Refroso	2,000
	scar	2,000
	Reconyc	1,570
Spyware (10,020)	180Solutions	2,000
	Coolwebsearch	2,000
	Gator	2,200
	Transponder	2,410
	TIBS	1,410
Ransomware (9,791)	Conti	1,988
	MAZE	1,958
	Pysa	1,717
	Ako	2,000
	Shade	2,128

악성코드 메모리 덤프는 그림 2에서 보는 바와 같은 2,916개의 악성코드 샘플을 VirusTotal에서 획득한 후, 2GB의 메모리를 가진 가상머신(VM)에서 실행시키면서 수집하였다. 최대한 실제 환경과 유사하도록 윈도우즈 10 환경에서 악성코드를 실행하였으며, 15초 간격으로 10개의 메모리 덤프를 캡처하여 29,298개의 악성코드 메모리 덤프를 추출하였다. 정상행위 메모리 덤프는 가상머신의 일부 응용프로그램을 수행한 후 메모리를 덤프하고 악성코드 메모리 덤프와 균형을 맞추기 위해 SMOTE를 사용하여 오버샘플링을 수행하였다[8].

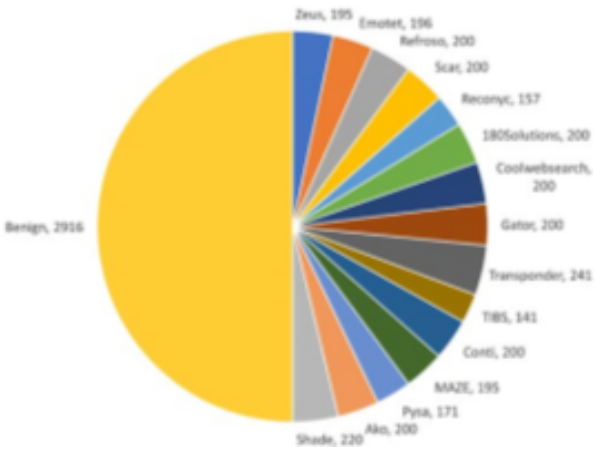

그림 2. 데이터세트 생성에 이용된 악성코드 현황[5]
Fig. 2. Malware used to generate the dataset[5]

2.3 선행연구 고찰

메모리 분석을 통해 악성코드를 탐지하는 연구는 다양하게 진행되고 있다. 본 절에서는 선행연구 중 본 연구와 동일한 데이터세트를 사용하여 악성코드 분류를 시도했던 연구와 난독화된 악성코드 탐지를 시도했던 연구들에 대해서만 간략하게 정리한다.

S. S. Shafin et al.[9]은 난독화된 메모리 악성코드를 탐지하기 위해 CNN이 가지는 특성 학습 능력과 Bi-LSTM(Bidirectional LSTM)이 가지는 시간적 모델링의 장점을 결합하여 CompactCBL(Compact CNN-BiLSTM)과 RobustCBL(Robust CNN- BiLSTM) 2종의 하이브리드 탐지모델을 제안하였다. CIC-MalMem-2022 데이터세트를 대상으로 이진 분류, 다중 분류 및 개별 분류 성능을 분석한 결과, CompactCBL 모

델은 각각 99.92%, 84.22%, 71.42%의 정확도를 달성했고, RobustCBL 모델은 각각 99.98%, 84.56%, 72.60%의 정확도를 달성하였다. 또한, RobustCBL모델은 정확도가 높고, CompactCBL 모델은 탐지 속도가 빠른 것으로 확인되었다.

M. A. Talukder et al.[10]은 머신러닝과 딥러닝 기법을 동시에 활용하는 하이브리드 모델을 제시하였다. SMOTE(Synthetic Minority Oversampling Technology) 및 XGBoost(Extreme Gradient Boosting)를 활용하여 데이터 불균형 문제를 해결하고 차원을 효율적으로 축소하였다. KDD CUP'99와 CIC-MalMem-2022 두 가지의 데이터세트를 대상으로 이진 분류 실험을 진행한 결과, KDD CUP'99 데이터세트에 대해서는 99.99%, CIC-MalMem-2022 데이터세트에 대해서는 100%의 정확도를 달성하였다.

M. Dener et al.[11]은 9가지의 머신러닝 및 딥러닝 기법을 사용하여 악성코드 탐지모델을 구축하고, CIC-MalMem-2022 데이터세트를 대상으로 이진 분류를 실시하였다. 그 결과 Gradient Boost Trees 모델과 Random Forest 모델이 가장 좋은 성능을 발휘하였으며, 정확도는 99.97%로 확인되었다.

D. Smith et al.[12]은 클러스터링 알고리즘으로 데이터세트를 분석한 후 7종의 분류 모델을 적용하여 CIC-MalMem-2022 데이터세트에 대한 이진 분류를 시도한 결과 99.99%의 정확도를 달성하였다.

M. Jerbi et al.[13]은 IMMU-Det 알고리즘을 제안하였다. IMMU-Det은 2단계 프로세스로 구성되어 있는데 먼저 MA(Memetic-Artificial immune systems)를 사용하여 악성 API 호출 벡터를 생성한 후, 복제 선택 알고리즘의 입력으로 사용하는 탐지모델을 생성하였다. CIC-MalMem-2022 데이터세트에 대한 이진 분류 정확도는 97.67%로 확인되었다.

M. H. L. Louk et al.[14]은 난독화된 악성코드 탐지모델을 구체적으로 제시하지는 않고, 다양한 모델의 일반화 가능성 평가에 중점을 둔 연구를 진행했다. 실험은 XGboost, Catboost, GBM, LightGBM 및 Random Forest 등 다양한 트리 기반 앙상블 모델을 활용하였으며, BODMAS, Kaggle 및 CIC-MalMem- 2022 데이터세트 등의 공개 데이터세트를 대상으로 성능분석을 실시하였다. 그 결과 하이퍼파라미터가 적절하게 잘 구성된 경우에는 모델 간 성능 차이가 통계적으로 유의미하게 나타나지는 않았지만 기존 탐지모델들보다 탐지성능이 뛰어나다는 점은 확인하였다.

2.4 BERT, Distill BERT, BART

BERT는 자연어처리 분야에서 활용되는 대표적인 딥러닝 모델 중 하나로, 2018년에 Google에서 발표하였다. 언어 이해와 관련된 다양한 자연어처리 작업에서 탁월한 성능을 보이며, 이후 출현한 다양한 언어모델의 기반으로 사용되기도 하였다. 주요 특징으로는 Transformer 아키텍처를 기반으로 하며, 입력 시퀀스를 양방향으로 처리한다. 그리고 대규모 말뭉치(Corpus)를 사용하여 문장 내 단어 간의 관계를 학습하고 문맥을 이해하며, MLM(Masked Language Model) 및 NSP(Next Sentence Prediction) 같은 자기지도 학습을 통해 언어 이해능력을 향상시킨다. 또한, 파인튜닝(Fine-tuning)을 통해 정확도를 좀 더 향상시킬 수 있고, 다양한 언어에 대한 자연어처리에도 적용할 수 있는 언어모델이다.[2]

DistilBERT는 BERT 모델의 경량화 버전으로, Hugging Face에서 개발하였다. BERT의 효과적인 성능은 유지하면서 모델 크기와 계산 비용을 줄이는 것이 DistilBERT의 주목적이었기 때문에, 저장공간이나 메모리 사용량이 감소하였다. 또한, BERT보다 적은 파라미터를 가지고 있어 학습 소요 시간 역시 짧고, 자원을 효율적으로 활용할 수 있다[3].

BART는 고품질 텍스트 생성 및 요약을 수행하는 딥러닝 모델로서, Facebook AI에서 개발하였다. 현재 다양한 자연어처리 분야에서 활용되고 있고, 특히 문장 및 문서 생성, 요약, 번역, 질문 응답 및 언어 이해 등과 같은 작업에서 탁월한 성능을 발휘한다. 양방향 및 자기회귀적 학습능력을 가지고 있으며, BERT와 마찬가지로 파인튜닝이 가능하다[4].

III. 제안 연구

3.1 Data converting

CIC-MalMem-2022 데이터세트는 55개의 속성값으로 구성되어 있다.

본 연구는 자연어처리 모델을 활용하기 때문에, 학습을 위해 그림 3에서 보는 바와 같이 55개의 속성값을 하나의 문장으로 병합하는 과정을 먼저 수행한다. 이때, 각 속성은 띄어쓰기(공백문자)로 구분하였고, 이러한 과정은 Python을 이용하여 구현하였다.

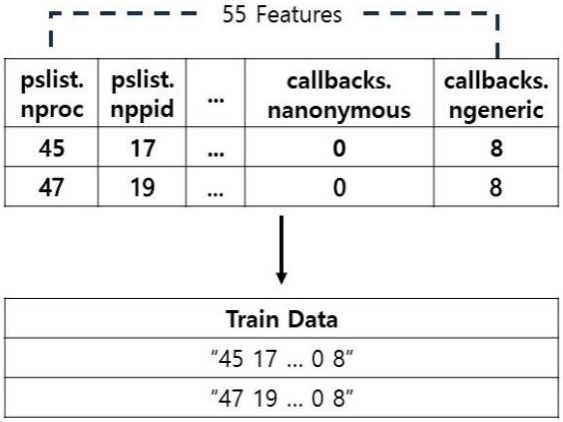

그림 3. 데이터 변환 예시
Fig. 3. Data converting example

이러한 변환과정은 메타데이터를 기반으로 모델 학습을 실시했던 선행연구들이 원-핫-인코딩, 특성 선택 및 정규화 등의 복잡하고 어려운 데이터 전처리를 위해 투입해야만 했던 노력과 시간을 획기적으로 감소시킬 수 있다. 또한, 새로운 메타데이터를 이용해 모델을 업데이트해야 하는 경우 데이터 전처리를 재수행하는 번거로움도 제거할 수 있다.

3.2 실험 설계

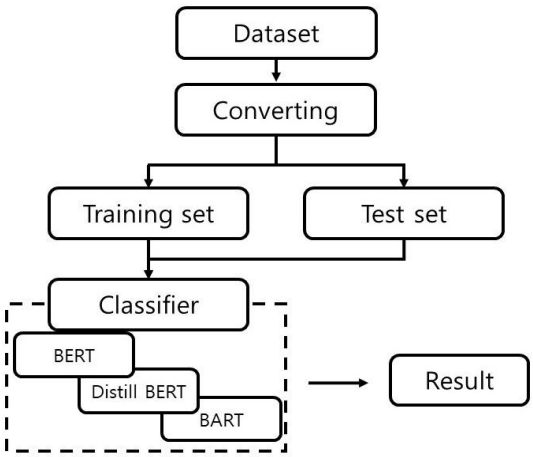

그림 4. 실험 설계
Fig. 4. Experimental design

하나의 문장으로 병합이 완료된 데이터세트를 BERT, DistilBERT 및 BART 모델에 직접 학습시켜 성능평가를 수행하였다. 학습(Train)과 테스트(Test) 데이터세트는 8대2의 비율로 구분했으며, 실험과정은 그림 4에서 보는 바와 같다.

실험은 구글에서 제공하는 Colab pro 환경(언어: Python 3.10 / 하드웨어: T4 GPU, 51GB RAM)에서 진행되었다.

3.3 실험 결과 및 분석

하나의 문장으로 병합된 데이터세트를 3종의 자연어처리 모델에 학습시킨 후 이진 분류 및 다중 분류 성능을 분석하였다.

이진 분류 성능평가 결과는 그림 5와 표 2를 통해 확인할 수 있다. 총 5500번의 학습을 진행하였고, 그 이상이 되면 비슷한 결과가 반복 도출되었다. 3종의 모델 중 BART가 99.93%로 가장 높은 정확도를 보여주었고, BERT의 정확도가 99.85%로 가장 낮았다.

그리고 선행연구들과의 비교는 표 2에서 확인할 수 있다. 3종 모델의 평가지표가 모두 99% 이상을 달성하기는 했지만, 선행연구 대비 미세하게 낮은 것으로 확인되었다. 그러나 복잡하고 시간이 많이 소요되는 데이터 전처리 과정을 간소화하고도 선행연구와 유사한 성능을 발휘했다는 점에서는 유의미한 성과를 달성했다고 볼 수 있다.

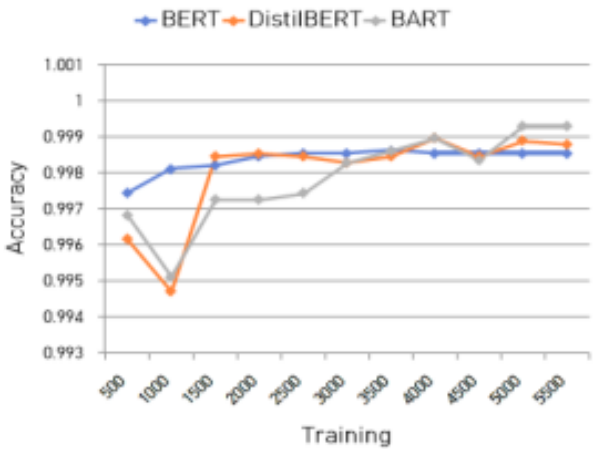

그림 5. 이진 분류 성능 비교
Fig. 5. Performance comparison of binary classification

표 2. 실험 결과(이진 분류)
Table 2. Experimental results(Binary classification)

Approach		Accuracy	F1-score	Precision	Recall
[10]		100	100	100	100
[11]		99.97	99.97	99.98	99.97
[12]		99.99	100	100	100
[13]		97.67	97.65	97.31	97.79
[9]	R-CBL	99.96	100	100	100
	C-CBL	99.92	100	100	100
Our approach	BERT	99.85	100	99.85	99.86
	Distil BERT	99.90	100	99.88	99.91
	BART	99.93	100	99.97	99.90

다중 분류 성능을 평가하기 위해 결과는 총 7000회까지 학습을 진행하였으며, 결과는 그림 6에서 보는 바와 같다. BERT는 5500회, DistilBERT는 7000회, 그리고 BART는 6500회에서 가장 높은 성능을 달성하였고, 정확도는 각각 84.69%, 85.10%, 84.44%로 확인되었다.

다중 분류를 시도한 선행연구와의 성능 비교 결과는 표 3에서 보는 바와 같다. BERT와 DistilBERT가 선행연구에서의 정확도 84.56% 대비 각각 0.13%P, 0.54%P 향상된 성능을 보였으나, BART는 0.12%P 낮은 성능을 보였다. DistilBERT는 모든 지표에서 높은 성능을 보였고, 다중 분류에서 선행연구 대비 성능 향상 효과도 뛰어났다. BERT는 이진 분류 성능이 가장 성능이 낮게 나타났으나, 다중 분류에서는 미세하게 향상된 결과를 보여주었다. Accuracy와 F1-score는 선행연구 대비 미세하게 향상되었다. BART는 이진 분류에서 이전 연구와 유사한 결과를 보였고, 다중 분류에서는 성능이 미세하게 낮지만 마찬가지로 유사한 결과를 나타냈다.

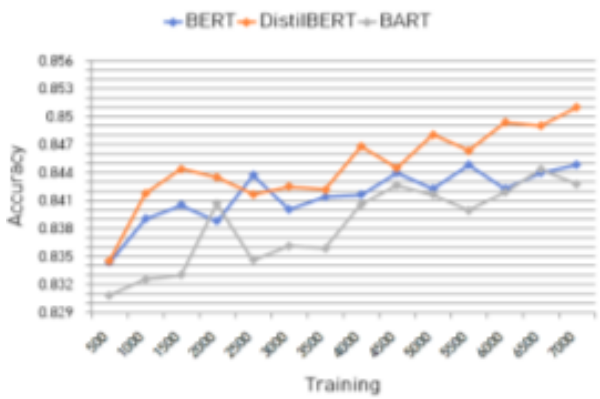

그림 6. 다중 분류 성능 비교
Fig. 6. Performance comparison of multi-class classification

표 3. 실험 결과(다중 분류)
Table 3. Experimental results(Multi-class classification)

Approach		Accuracy	F1-score	Precision	Recall
[9]	R-CBL	84.56	84	85	85
[9]	C-CBL	84.22	84	84	84
Our approach	BERT	84.69	84.69	84.75	84.69
	Distil BERT	85.10	85.07	85.06	85.10
	BART	84.44	84.35	84.37	84.44

3종 모델의 다중 분류에 대한 혼동행렬은 그림 7, 그림 8 및 그림 9에서 확인할 수 있다. 이진 분류에서 99% 이상의 분류성능을 달성했던 것처럼 다중 분류에서도 정상 트래픽은 오분류가 거의 없이 비교적 정확하게 분류하였다. 그러나 3종의 악성코드 클래스 사이에서는 오분류가 다수 발생하여 정확도가 떨어지는 결과를 나타내었다.

3종 모델이 가장 잘 분류하는 악성코드의 클래스 역시 상이하였다. BERT(그림 7)는 Spyware 클래스, DistilBERT(그림 8)는 Ransomware 클래스, BART(그림 9)는 Trojan 클래스를 가장 잘 분류하였다. 이러한 결과가 도출된 이유를 현시점에서 정확하게 판단하기는 힘들지만, 향후 연구에서는 서로 다는 클래스에 대해 가장 높은 분류성능을 달성한 모델을 적용하는 하이브리드 접근방법을 적용한다면 개별 클래스에 대한 분류 정확도가 보다 개선될 수 있을 것으로 판단된다.

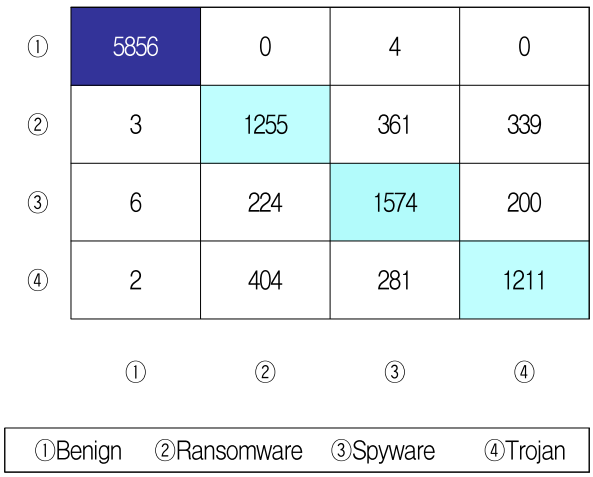

그림 7. 다중 분류 혼동 행렬(BERT)
Fig. 7. Confusion matrix of multi-class classification(BERT)

①	5853	1	5	1
②	1	1371	272	314
③	4	291	1491	218
④	1	447	239	1211
	①	②	③	④

①Benign ②Ransomware ③Spyware ④Trojan

그림 8. 다중 분류 혼동행렬(DistilBERT)
Fig. 8. Confusion matrix of multi-class classification(DistilBERT)

①	5855	1	4	0
②	1	1295	299	363
③	8	229	1528	239
④	2	364	236	1296
	①	②	③	④

①Benign ②Ransomware ③Spyware ④Trojan

그림 9. 다중 분류 혼동행렬(BART)
Fig. 9. Confusion matrix of multi-class classification(BART)

Ⅳ. 결론 및 향후 과제

악성코드 탐지를 위한 인공지능 기술의 활용이 증가하고 있지만, 탐지모델 구축 과정에서 데이터 전처리를 위해 많은 시간과 노력이 필요하다. 또한, 새로운 악성코드가 출현할 경우 전처리를 재수행하여 모델에 학습시켜야 하는 번거로움이 발생한다. 이러한 문제를 해결하기 위해 본 연구에서는 자연어처리 기법을 활용하여 전처리 과정을 간소화하면서 효율적으로 악성코드를 탐지하는 방안을 제안하였다.

데이터세트는 CIC-MalMem-2022를 사용하였고, 55개의 속성을 공백문자로 구분하면서 하나의 문장으로 병합한 후 BERT, DistilBERT 및 BART 3가지 모델에 학습시키고 이진 분류 및 다중 분류 성능을 분석하였다. 그 결과 이진 분류에서 각각 정확도 99.85%, 99.90%, 99.93%를 달성했고, 다중 분류에서는 각각 84.69%, 85.10%, 84.44%의 정확도를 달성하였다. 그리고 이진 분류에서는 BART, 다중 분류에서는 DistilBERT 모델의 성능이 가장 우수한 것으로 확인되었다.

이상과 같은 결과를 통해 자연어처리 기반의 악성코드 탐지가 선행연구 대비 월등한 성능 향상을 달성하지는 못했지만 복잡한 전처리 과정에 투입되는 비용을 획기적으로 감소시키면서도 우수한 성능을 발휘할 수 있다는 가능성은 충분히 확인하였다. 그리고 본 연구에서 제안하는 접근방법은 하루가 다르게 급증하는 신종 악성코드에 효율적으로 대응하기 위해 필수적인 탐지모델의 신속한 업데이트 측면에서도 훌륭한 대안이 될 수 있을 것으로 판단된다.

향후에는 CIC-MalMem-2022 데이터세트를 대상으로 다중 분류에서의 정확도를 개선하고, 다른 형태의 데이터세트를 대상으로 한 자연어처리 모델의 적용 가능성도 분석하면서 연구를 확장해 나갈 예정이다.

References

[1] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing", IEEE Computational Intelligence Magazine, Vol. 13, No. 3, pp. 55-75, Jul. 2018. https://doi.org/10.1109/mci.2018.2840738.

[2] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding", arXiv preprint, arXiv:1810.04805, pp. 4171-4186, Oct. 2018. https://doi.org/10.48550/arXiv.1810.04805.

[3] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", arXiv preprint, arXiv:1910.01108, pp. 1-5, Apr. 2020. https://doi.org/10.48550/arXiv.1910.01108.

[4] M. Lewis, et al.,, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension",

arXiv preprint, arXiv:1910.13461, pp. 1-8, Oct. 2019. https://doi.org/10.48550/arXiv.1910.13461.

[5] Canadian Institute for Cybersecurity, "CIC-MalMem-2022", 2022. https://www.unb.ca/cic/datasets/malmem-2022.html [accessed: Sep. 07, 2023]

[6] S. García, S. Ramírez-Gallego, J. Luengo, J. M. Benítez, and F. Herrera, "Big data preprocessing: methods and prospects", Big Data Analytics, Vol. 1, No. 9, pp. 1-22, Nov. 2016. https://doi.org/10.1186/s41044-016-0014-0.

[7] G. Press, "Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says", Forbes, Mar. 2016.

[8] S. S. Shafin, G. Karmakar, and I. Mareels, "Obfuscated Memory Malware Detection in Resource-Constrained IoT Devices for Smart City Applications", Sensors, Vol. 23, No. 11, pp. 5348, May 2023. https://doi.org/10.3390/s23115348.

[9] T. Carrier, P. Victor, A. Tekeoglu, and A. H. Lashkari, "Detecting Obfuscated Malware using Memory Feature Engineering", In Proc. of the ICISSP, pp. 177-188, Feb. 2022. https://doi.org/10.5220/0010908200003120.

[10] M. A. Talukder, K. F. Hasan, M. M. Islam, M. A. Uddin, A. Akhter, M. A. Yousuf, and M. A. Moni, "A dependable hybrid machine learning model for network intrusion detection", Journal of Information Security and Applications, Vol. 72, pp. 103405, Feb. 2023. https://doi.org/10.1016/j.jisa.2022.103405.

[11] M. Dener, G. Ok, and A. Orman, "Malware detection using memory analysis data in big data environment", Applied Sciences, Vol. 12, No. 17, pp. 8604, Aug. 2022. https://doi.org/10.3390/app12178604.

[12] D. Smith, S. Khorsandroo, and K. Roy, "Supervised and unsupervised learning techniques utilizing malware datasets", IEEE 2nd International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, pp. 1-7, Feb. 2023. https://doi.org/10.1109/ICAIC57335.2023.10044169.

[13] M. Jerbi, Z. C. Dagdia, S. Bechikh, and L. B. Said, "Immune-Based System to Enhance Malware Detection", In IEEE 2023 Congress on Evolutionary Computation, pp. 1-8, Apr. 2023.

[14] M. H. L. Louk and B. A. Tama, "Tree-Based Classifier Ensembles for PE Malware Analysis: A Performance Revisit", Algorithms, Vol. 15, No. 9, pp. 332, Sep. 2022. https://doi.org/10.3390/a15090332.

저자소개

신 동 환 (Dong-Hwan Shin)

2012년 2월 : 한국해양대학교 데이터정보전공(공학사)
2024년 2월 : 국방대학교 컴퓨터공학과(공학석사)
2024년 1월 ~ 현재 : 해병대 연평부대 정보통신과장
관심분야 : 머신러닝, 침입 탐지 시스템, 자연어처리, 인공지능

김 한 석 (Han-Seok Kim)

2014년 2월 : 육군사관학교 국제관계학과(문학사)
2024년 2월 : 국방대학교 컴퓨터공학과(공학석사)
2024년 1월 ~ 현재 : 육군본부 지능정보기술단 작전소프트웨어개발장교
관심분야 : 머신러닝, 침입 탐지 시스템, 자연어처리, 인공지능

이 수 진 (Soo-Jin Lee)

1992년 : 육군사관학교 컴퓨터공학과(공학사)
1996년 : 연세대학교 컴퓨터공학과(공학석사)
2006년 : 한국과학기술원(KAIST) 컴퓨터공학과(공학박사)
2006년 ~ 현재 : 국방대학교 컴퓨터공학과 교수
관심분야 : 국가 사이버 보안 정책, 침입 탐지 시스템, 모바일 네트워크 보안, 머신러닝, 암호화 이론 및 응용