Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.5, Oct. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.5.1019

악성코드 대응을 위한 신뢰할 수 있는 AI 프레임워크*

신 경 아,[1†] 이 윤 호,[3] 배 병 주,[4] 이 수 항,[5] 홍 희 주,[5] 최 영 진,[5] 이 상 진[2‡]
[1,2]고려대학교 정보보호대학원 (대학원생, 교수), [3,4,5](주)에프원시큐리티 (대리, 과장, 사원)

Trustworthy AI Framework for Malware Response*

Kyounga Shin,[1†] Yunho Lee,[3] ByeongJu Bae,[4] Soohang Lee,[5] Heeju Hong,[5]
Youngjin Choi,[5] Sangjin Lee[2‡]
[1,2]School of Cybersecurity, Korea University (graduate student, Professor),
[3,4,5]F1Security (assiatant manager, manager, staff)

요 약

4차 산업혁명의 초연결사회에서 악성코드 공격은 더욱 기승을 부리고 있다. 이러한 악성코드 대응을 위해 인공지능기술을 이용한 악성코드 탐지 자동화는 새로운 대안으로 주목받고 있다. 그러나, 인공지능의 신뢰성에 대한 담보 없이 인공지능을 활용하는 것은 더 큰 위험과 부작용을 초래한다. EU와 미국 등은 인공지능의 신뢰성 확보방안을 강구하고 있으며, 2021년 정부에서는 신뢰할 수 있는 인공지능 실현 전략을 발표했다. 정부의 인공지능 신뢰성에는 안전과 설명가능, 투명, 견고, 공정의 5가지 속성이 있다. 우리는 악성코드 탐지 모델에 견고를 제외한 안전과, 설명가능, 투명, 공정의 4가지 요소를 구현하였다. 특히 외부 기관의 검증을 통해 모델 정확도인 일반화 성능의 안정성을 입증하였고 투명을 포함한 설명가능에 중점을 두어 개발하였다. 변화무쌍한 데이터에 의해 학습이 결정되는 인공지능 모델은 생명주기 관리가 필요하다. 이에 인공지능 모델을 구성하는 데이터와 개발, 서비스 운영을 통합하는 MLOps 프레임워크에 대한 수요가 늘고 있다. EXE 실행형 악성코드와 문서형 악성코드 대응 서비스는 서비스 운영과 동시에 데이터 수집원이 되고, 외부 API를 통해 라벨링과 정제를 위한 정보를 가져오는 데이터 파이프라인과 연계하도록 구성하였다. 클라우드 SaaS 방식과 표준 API를 사용하여 다른 보안 서비스 연계나 인프라 확장을 용이하게 하였다.

ABSTRACT

Malware attacks become more prevalent in the hyper-connected society of the 4th industrial revolution. To respond to such malware, automation of malware detection using artificial intelligence technology is attracting attention as a new alternative. However, using artificial intelligence without collateral for its reliability poses greater risks and side effects. The EU and the United States are seeking ways to secure the reliability of artificial intelligence, and the government announced a reliable strategy for realizing artificial intelligence in 2021. The government's AI reliability has five attributes: Safety, Explainability, Transparency, Robustness and Fairness. We develop four elements of safety, explainable, transparent, and fairness, excluding robustness in the malware detection model. In particular, we demonstrated stable generalization performance, which is model accuracy, through the verification of external agencies, and developed focusing on explainability including transparency. The artificial intelligence model, of which learning is determined by changing data, requires life cycle management. As a result, demand for the MLops framework is increasing, which integrates data, model development, and

Received(08. 22. 2022), Modified(09. 27. 2022), Accepted(09. 27. 2022)
* 이 논문은 서울시 산학연 협력사업 2021년도 인공지능(AI) 기술사업화 지원사업(CY210066)의 지원을 받아 수행된 연구임.
† 주저자, fruitnara@korea.ac.kr
† 교신저자, sangjin@korea.ac.kr(Corresponding author)

service operations. EXE-executable malware and documented malware response services become data collector as well as service operation at the same time, and connect with data pipelines which obtain information for labeling and purification through external APIs. We have facilitated other security service associations or infrastructure scaling using cloud SaaS and standard APIs.

Keywords: malware, trustworthy AI, XAI, AI lifecycle, MLOps

I. 서 론

1.1 연구배경 및 목적

2019년 COVID-19 발생 이후로 전 세계적으로 피싱 공격, 악성코드, 개인정보 탈취를 포함한 사이버 공격이 급격히 상승하는 추세이며 관련된 피해 사례가 속출하고 있다.

악성코드는 기하급수적인 증가 및 다형성에 의한 자동 변형 생성 등으로 기존 분석가의 수작업에 의한 대응은 한계를 보인다.

이러한 악성코드 대응을 위해 인공지능 기술을 이용한 악성코드 탐지 자동화는 새로운 대안으로 주목받고 있다[1]. 머신러닝을 통한 악성코드 분석은 보안 전문가의 분석 시간 대비해 2~10배 이상 빠르다. 또한 빅데이터 분석을 통해 새롭게 추가되는 악성코드에 대한 대응을 강화할 수 있다.

그러나 인공지능의 신뢰성이 확보되지 않는다면 더 큰 위험과 부작용을 초래할 수 있다. 2021년 5월 과학기술정보통신부에서는 사람이 중심이 되는 인공지능을 위한 신뢰할 수 있는 인공지능 실현 전략을 발표했다[2]. EU, 미국 등은 인공지능 신뢰성을 인공지능 윤리 실천의 핵심 요소로서 강조하고 제도, 윤리, 기술 측면에서 신뢰성 확보방안을 강구하고 있다[2].

본 논문에서는 빅데이터 중심의 인공지능 학습은 데이터 분석과 오탐 원인을 분석하고 예측 과정을 설명하고자 하였다. 기존에는 악성코드 탐지에 인공지능 알고리즘을 적용하거나 학습 모델 생성 프레임워크를 개발하여 자동화하는 사례는 있지만, 악성코드 탐지 모델을 위한 학습 데이터 분석이나 오탐 분석 및 검증, 피드백 분석 등을 제공하지 못하였다. 본 논문에서는 학습 데이터 샘플링을 위한 데이터 정제부터 다차원 데이터 분석 및 오탐 분석과 전문가 피드백 분석을 결합하여 신뢰성 있는 인공지능을 구현하고자 하였다.

AI 프로젝트에서 모델링은 아주 작은 업무에 불과하며, 비즈니스 데이터와 비즈니스 활용이 훨씬 더 크고 복잡한 업무이다[23]. AI 모델의 수정 불가능, 학습과 운영 데이터의 불일치로 인한 예측 오류, 입출력 데이터의 해석 불가능 등으로 비즈니스에 활용할 수 없는 AI 모델은 무용지물이 된다.

신뢰성 있는 AI 모델의 지속적인 관리와 안정적인 운영을 목적으로 MLOps 개념을 도입하여, 모델의 생명주기 관리를 위한 데이터와 모델, 비즈니스의 통합 프레임워크[3]를 제안한다.

본 연구의 내용은 다음과 같다.

- 실행 파일과 문서 파일을 대상으로 인공지능 기반 악성코드 여부를 판정하고 세부 유형으로 분류하는 모델 생성
- 데이터 수집부터 정제, 피처 엔지니어링, 단일/복합 모델 생성, 데이터 및 오탐 분석과 피드백 통합 분석 등 머신러닝 전 과정을 자동화하는 지능화 프레임워크 개발
- 통계적 데이터 분석과 오탐 분석 및 시각화를 통한 설명 가능한 인공지능 구현
- 모델 개발과 데이터 파이프라인 연계. 보안 서비스를 통합한 MLOps 프레임워크 개발

본 논문은 다음과 같이 구성된다. 1장은 서론, 2장은 관련 연구, 3장은 신뢰성 있는 인공지능 기술로 모델의 생성과 평가, 검증 분석에 포함된 신뢰성 요소를 설명한다. 4장. 악성코드 탐지 AI 프레임워크에서는 인공지능 모델의 생명주기 관점에서 모델 생성과 평가, 검증 분석의 프레임워크를 설명한다. 5장은 결론으로 끝맺는다.

II. 관련 연구

2.1 악성코드 탐지 AI 모델

악성코드 분석은 EXE 바이너리를 이용한 정적분석[4-7]과 EXE를 실행시켜 악성 행위를 분석하는 동적 분석[8]으로 나눌 수 있다. 정적분석은 EXE 바이너리를 이용한다는 점에서 간단하게 데이터 추출이 가능하지만, 난독화로 더 복잡한 분석을 요구한다. 난독화 상태에서도 메타 정보는 난독화의 영향이

없어 악성코드 탐지에 유용하다[6]. EXE 정적 메타 정보와 동적 분석 정보를 결합한 하이브리드 AI 모델 연구가 이루어졌다[9].

악성코드 탐지 모델은 초기에는 데이터마이닝[10]을 이용한 시그니처 자동 생성에서 시계열 정보를 이용한 워드 임베딩[11][12]과 딥러닝[13] 기술로 발전하고 있다. 자동화된 피처 엔지니어링을 위해 빈도수[4, 14]나 이미지[15 - 17], 해시, 엔트로피 등을 이용한 연구와 워드 임베딩[18 - 21]을 이용한 딥러닝 연구가 이루어졌다.

기존 연구는 특정 모델을 심층 연구하였다면 우리는 악성코드 탐지 모델에 다양한 방법을 활용하였다. PE 메타 정보를 이용한 모델, 이미지 모델, IG(엔트로피) 등을 이용한 빈도수 통계 모델, 워드 임베딩 모델 등 다양한 방법으로 정적, 동적 분석 데이터를 활용하여 모델링이 가능한 프레임워크를 구축하였다.

2.2 인공지능의 신뢰성

2021년 과학기술정보통신부에서 발표한 인공지능의 신뢰성은 안전과 설명 가능, 투명, 견고, 공정의 5가지 속성으로 구성된다[2]. 안전(safety)은 인공지능의 판단과 예측 결과로 인한 시스템 동작과 기능 수행이 사람과 환경에 악영향을 미치지 않도록 예방할 수 있는 상태를, 설명 가능(explainability)은 인공지능의 판단과 예측의 근거와 결과에 이르는 과정이 사람이 이해할 수 있는 방식으로 제시되거나, 문제 발생 시 결과 도출과정을 분석할 수 있는 상태를, 투명(transparency)은 인공지능의 판단과 예측 등 작동과정과 이를 구현하기 위한 구성 요소를 이용자가 인지하고 확인 검사가 가능한 상태를, 견고(robustness)는 인공지능이 외부의 간섭 및 극한적인 운영 환경의 영향 없이 본래의 성능 및 기능을 유지하는 상태를, 공정(fairness)은 인공지능이 데이터를 처리하는 과정에서 특정 그룹에 대한 차별이나 편향이 없도록 하는 기능성을 의미한다[2].

신뢰성 연구는 신뢰성 기준을 수립하거나, XAI 알고리즘 연구가 대부분이나 우리는 프레임워크의 다양한 모델에 신뢰성 요소를 활용하는 것을 목적으로 한다. 견고를 제외한 4가지 신뢰성 요소를 악성코드 탐지를 위한 인공지능 모델에 적용하고 입증하였다.

2.3 MLOps(Machine Learning Operations)

MLOps는 데이터 관리 및 ML 시스템 개발과 서비스 운영(Operations)을 통합해 안정적으로 서비스를 제공할 수 있도록 신속 · 유연한 개발을 추구하는 협업 방식을 말한다.

인공지능 개발에 필요한 데이터 구축, 모델 개발 및 훈련, 모델 배포 등의 과정이 매끄럽지 못하거나, 파이프라인이 제대로 구축되지 않으면, 비용이 증가하고 프로젝트가 지연될 확률이 높다. 인공지능 모델 도입에 많은 기업이 실패하는 이유는 AI 모델이 프로젝트의 완성이라고 생각하고, 데이터 의존적인 모델의 수명주기 복잡성을 관리하지 못했기 때문이다. 즉 AI 운영 단계에서 모니터링과 지속적 개선을 간과했기 때문이다[22]. 인공지능 개발이 성공하려면, 일관된 데이터 흐름과 자동화된 모델 배포, 더불어 인공지능 운영 서비스를 이해하고 데이터와 모델과 서비스를 통합할 수 있어야 한다.

현재 MLOps의 대표 기업은 Azure ML Service의 마이크로소프트, SageMaker의 아마존 웹서비스(AWS), 구글 클라우드가 있으며, 알고리드미아, 셀던 등 다양한 스타트업에서도 MLOps 서비스를 개발하고 있다[23].

인공지능 개발의 효율성과 생산성을 개선할 수 있는 MLOps 개념을 적용하여 본 논문에서는 악성코드 대응 서비스를 위한 AI 프레임워크를 제안한다.

2.4 악성코드 탐지 및 분류를 위한 인공지능 프레임워크

그동안 연구된 악성코드 탐지와 분류의 인공지능 프레임워크 연구는 대부분 모델 생성에 집중되어 있다.

[24]는 정적 및 동적 악성코드 분석 기법의 하이브리드 분석 기법을 활용해 다단계 분석을 수행하고 학습 모델을 검증하는 탐지 프레임워크를 연구하였다. [25]은 사전 훈련된 네트워크 모델을 최적화된 방식으로 통합하는 새로운 아키텍처 프레임워크를 연구하였다. 전이 학습을 이용하여 딥러닝 모델을 결합하였다. [26]은 머신러닝 분류를 위한 정적 프레임워크에 대한 연구를 하였고, [27]은 에너지 소비로 통신의 은닉 여부를 판단하고 신경망을 사용해 탐지하는 측정 프레임워크에 대한 연구를 하였다. 악성코드 AI 프레임워크에서는 모델 생성 자동화를 위해

여러 모델을 결합하거나 파라미터를 변경하는 기능이 연구되었다. 그러나, 인공지능 모델의 신뢰성 연구나 모델의 생명주기 관점의 데이터와 서비스 연계의 통합 프레임워크에 대한 연구는 이루어지지 않았다. 본 논문은 실효성 있는 AI 모델 개발을 위하여, 설명가능 관점의 다차원 데이터 분석과 신뢰할 수 있는 인공지능 모델 개발, 데이터와 서비스를 결합한 통합 프레임워크에 대한 연구를 하고자 한다.

III. 신뢰성 있는 인공지능 기술

3.1 개요

본 논문에서는 인공지능의 신뢰성 요소 중 견고를 제외한 안전과 투명, 설명 가능, 공정을 구현하였고 특히 외부 기관의 검증을 통해 일반화 성능의 안전성을 입증하였고 투명을 포함한 설명 가능에 중점을 두어 개발하였다[2].

Fig.1.은 데이터 정제, 피처 엔지니어링, 모델 학습, 평가, 데이터 분석, 검증 등의 AI 개발 과정을 보여준다. 단계별 수행 작업과 신뢰성 요소가 어떻게 반영되었는지를 나타냈다. 악성코드 탐지 모델의 안전성은 정확도 보장을 통하여, 악성코드 오탐을 줄이고 보안사고를 예방하는 것이다. 정확도 높은 우수한 모델을 만드는 과정은 학습 데이터 선별과 벡터화, 모델링 과정에 해당한다. 공정성은 모델 평가에 있다. 학습 데이터와 테스트 데이터의 분리와 적정한 데이터의 수량과 비율을 통해 평가의 공정성을 유지해야 한다. 다차원 분석을 통한 데이터 투명성과 모델 검증을 통해 진행된다. 모델의 추론을 설명하기 위해 모델 검증과 오탐을 분석하는 기능을 제공하였다. 이상과 같은 안전, 투명, 설명 가능과 공정을 적

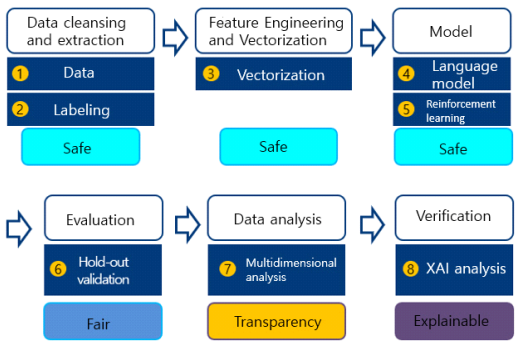

Fig. 1. Core technology for each stage

Table 1. Technology Description

Skill	Description
Good Data	Sampling data with data cleansing
Labeling	Generate binary class/multi-class labels with detection types from Global Vaccine
Vectorization	Various vectorization algorithms (Inst2Vec, One-hot Encoding, etc.) and independently developed algorithms (Binary Level & Difference)
Language model	Using BERT to learn API Sequence and cuckoo signature
Reinforcement learning	Quickly detect malicious code variants (new malicious code, etc.) by learning PE static/dynamic extracts with DQN
Hold-out Validation	Test with sufficient test sets across the entire dataset
Multi-dimensional analysis	Analyze data with statistical analysis, outliers analysis, etc
XAI analysis	Model verification and analysis on features using SHAP and PDP

용하여 신뢰성 있는 인공 지능 기술을 구현하였다.

인공지능 모델을 위한 기술적 특징은 Table 1과 같다.

3.2 데이터 수집

글로벌화 되는 악성코드에 대응하기 위해 글로벌 유료 데이터를 구매하였고, 글로벌 백신사의 악성 판정 값을 구하여 정제와 레이블 처리를 진행하였다.

본 연구는 exe 파일 중 PE 악성코드와 문서 악성코드 중 PDF, hwp, MS Office 형식의 악성코드를 대상으로 한다. exe의 경우 VirusShare[28]와 VirusTotal[29]에서 악성코드 및 정상 파일을 유료로 수집하여 총 112만 개의 데이터셋으로 실험을 진행하였다. 또한 자사의 보안 서비스를 통해 확보한 최신 악성코드를 활용하여 모델의 일반화 성능 검증에 활용하였다. 문서의 경우 exe와 마찬가지로 VirusTotal에서 악성 파일과 정상 파일을 유료 수집하여 PDF, hwp, MS office와 같이 3종류의 문서 유형 데이터셋을 수집하였다.

3.3 데이터 정제 및 추출

수집된 데이터셋은 AI 모델에 사용하기 위해 정제 및 추출 과정을 거쳐야 한다. 정제된 exe 파일은 동적 및 정적분석을 통한 다양한 데이터 추출을 진행하였고, 효과적인 학습을 위하여 피처 엔지니어링과 벡터화와 자체 개발한 빈도수 기반의 BRD(Binary Relevance & Difference)를 이용하였다.

3.3.1 데이터 정제

데이터 정제는 VirusTotal의 score를 이용하여 수행하였다. VirusTotal은 70여 개 백신사의 분석 결과를 토대로 악성 여부를 판정하고, 악성이라고 판정하고 백신사의 개수를 score로 회수한다. 우리가 수집한 데이터셋의 VirusTotal score의 분포도는 Fig.2.와 같다. 백신사의 정상과 악성 판정이 혼재하는 30 이하 구간의 데이터를 제거하고 다수 백신사가 악성이라 판정한 파일의 score와 카스퍼스키 판정 결과에 가중치를 부여하여 데이터 샘플링을 진행하였다. 데이터 정제 이후, Table 2와 같이 API 3-gram의 예측 성능이 8%가 향상되었고, Precision, Recall, F1-Score가 모두 고른 분포를 보였다. (Table 2는 초기 모델의 정확도로 이후 모델 개선을 통해 정확도는 상승함.)

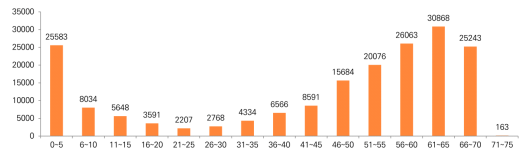

Fig. 2. VirusTotal Score histogram of exe dataset. Since score 1 to 30 are ambiguous sections to determine whether the file is malicious or benign, we removed these data from the dataset.

Table 2. Upgraded performance after cleansing: about 8% improvement

Cleansing	Accuracy	Precision	Recall	F-score
Before	85.7%	83.7%	88.5%	86.0%
After	93.8%	93.5%	94.2%	93.8%

3.3.2 데이터 추출

데이터 정제가 끝난 후 샘플링 된 데이터를 사용하여 AI 모델 학습에 사용될 정보를 추출해야 한다. exe 파일의 경우 exe 파일에서 PE 정보, byte 정보, 동적 분석 정보, API sequence 정보를 추출하였다. 문서 파일은 문서 구조에서 스트림을 추출하였다(Table 3).

Table 3. Extraction method by file type

File Type	Analysis Type	Extract Method
exe	Static analysis	PE
		Byte
	Dynamic analysis	Dynamic info.
		API sequence
Document	Static analysis	Stream keywords

3.3.2.1 PE 정보 추출

exe 파일의 경우 Python 라이브러리를 이용하여 정적분석 데이터를 추출할 수 있다. PE 헤더 및 각 영역의 정보를 추출 및 가공하여 데이터셋을 구성하였다.

3.3.2.2 Byte 정보 추출

PE 파일의 byte 정보를 추출하여 Fig.3.과 같이 파일 전체 바이트를 이미지화하고 리사이징을 진행하였다.

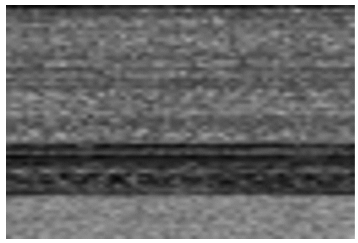

Fig. 3. Image of exe byte data. It is an byte image from an exe file, it showed malware families sometimes have a certain pattern.

3.3.2.3 동적 분석 데이터 추출

동적 분석 데이터는 가상 환경에서 PE 파일을 직접 실행시켜 얻은 정보이다. Cuckoo Sandbox의 가상 환경에서 파일을 실행하고 네트워크 정보, 프로세스 정보, 레지스트리 정보 등의 동적 분석 데이터를 추출하였다.

3.3.2.4 API sequence 추출

API sequence는 PE 파일의 API 호출 정보로, PE 파일의 정확한 행위를 알 수 있는 정보이다. 이는 동적 분석 데이터 추출 결과물에서 API call sequence 데이터 부분을 수집하였다.

3.3.2.5 문서 데이터 추출

문서형 악성코드의 OLE(Object Linking and Embedding) 문서 구조에서 스트림 추출 후 빈도수를 이용하여 키워드를 선별하고 데이터를 추출하였다.

3.4 피처 엔지니어링 및 벡터화

추출이 끝난 데이터는 AI가 효과적으로 학습할 수 있도록 피처 엔지니어링 작업이 필요하고, API sequence 같은 문자열 데이터는 벡터화 작업이 필요하다.

3.4.1 API sequence 가공

API sequence는 문자열 데이터이므로, AI가 학습할 수 있도록 수치로 바꿔주는 벡터화 작업이 필요하다. 벡터화 방식에는 빈도수 통계방식의 N-gram 벡터와 문맥 의미를 표현하는 워드 임베딩의 Word2Vec을 사용하였다. 병행하여 두 방식의 성능을 비교하기 위하여 CountVec과 LabelEncoder 벡터를 만들었다. Word2Vec을 이용한 기존 연구에는 JAVA와 C의 API를 대상으로 API의 사용 유사성과 의미 유사성을 표현하는 벡터 연구 API2Vec (Fig.4.)[20]과, opcode를 벡터화한 Inst2Vec[21] 사례가 있다. N-gram 벡터화에서는 상호관련성을 구하는 빈도수 통계방식을 적용하였다. N-gram에서는 feature selection 알고리즘인 MI, IG(엔트로피), CHI 등을 이용하여 피처 가중

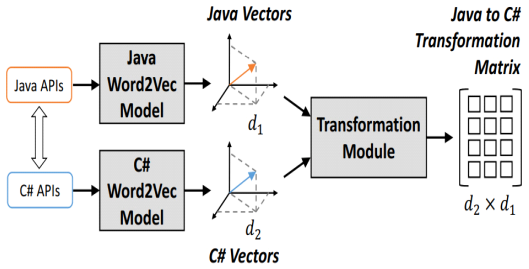

Fig. 4. API2Vec architecture[20]

	A	B	C	D	E	F	G	H
1	filename	regqueryv	getsystemi	ldrgetproc	ntclose	nt	outputdeb	process32	regenumk
2	d7dd5584	0	0.438964	0	0.608187	0	0	0
3	d2c55c6da	10.78931	3.16469	1.668112	1.021074	0	0	0.431867
4	c0697de93	0	0	0	0.304094	0	0	0
5	8727b636c	3.451602	0	0.194909	0.815014	0	0	0
6	b97fee317	5.197663	0.695741	1.375137	0.575585	0	0	0
7	fd2a02509	0	0	0	0	0	0	0
8	b41fff596d	9.503962	0	0.698741	0	0	0	0.431867
9	8d81b4fc5	10.89786	1.928069	1.377243	1.226441	0	0	0.431867
10	835e6e0f8	0	0	0	0	0	0	0

Fig. 5. N-gram Vectorization Results

치를 구하고 가중치와 빈도수를 이용하여 벡터화하는 자체 개발한 BRD 알고리즘을 사용하였다. 벡터화 결과는 Fig.5.와 같다.

3.4.2 Feature Selection

AI 학습을 위해서는 방대한 양의 데이터로부터 의미 없는 피처는 버리고 유용한 피처만 찾는 작업인 feature selection[30, 31]이 필요하다.

Feature selection 방법에는 Wrapper, Embed, Filtering 등이 있으며, 이 연구에서는 wrapper 방법 중 stepwise selection과 filtering을 주로 사용하여 최적의 feature set을 찾아내었다.

3.5 모델 생성

AI 모델은 다양한 알고리즘이 존재한다. 데이터의 유형에 따라 적절한 알고리즘을 사용해야 하고, 성능 향상을 위해 여러 모델을 결합하여 사용하기도 한다. 이 연구에서는 하나의 알고리즘만 사용하는 단일 모델, 단일 모델을 결합하여 만든 복합 모델, 시퀀스 데이터 학습을 위한 언어 모델, feature selection을 위한 강화학습 모델을 사용하였다. 이후 모델 리스트에서는 각 모델에 사용된 데이터와 알고리즘에 대한 정보를 제공하여 투명성을 확보할 수 있게 하였다.

3.5.1 단일 모델

단일 모델은 LightGBM, XGBoost, Random Forest와 같은 머신러닝 모델을 사용하였고, 딥러닝 모델로는 CNN, RNN, LSTM 등의 모델을 사용하였다. 이 중 CNN은 byte 데이터와 API sequence 데이터를 위해 사용하였고, RNN과 LSTM은 API sequence 데이터를 사용하여 Word2Vec을 이용하는 워드 임베딩 모델로 사용하였다.

3.5.2 복합 모델

복합 모델은 여러 단일 모델을 결합하는 방법으로 Multi Modal, Ensemble 등의 기법이 있다. 이 연구에서는 Fig.6.와 같이 PE 정적 모델과 동적 API 모델의 예측 score를 기반으로 높은 score를 나타내는 모델의 결과를 반영하도록 복합 모델을 만들었다. 또한, Fig. 7과 같이 초기 학습 후 학습된 모델들을 다시 학습하여 학습 중에 모델을 합치는 알고리즘을 사용하였다.

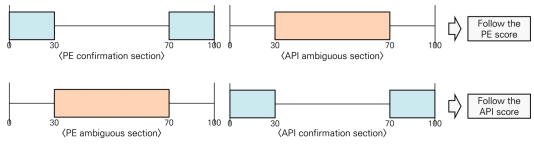

Fig. 6. Combined model using regression score

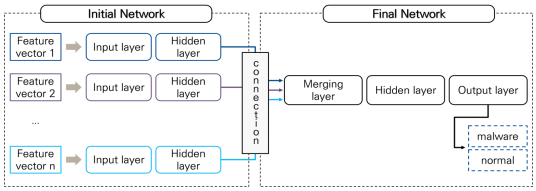

Fig. 7. Composite model combining several single model

3.5.3 언어 모델

API sequence 데이터와 같은 시퀀스 데이터는 시계열적 특성을 이용하여 학습하는 것이 좋다. 많은 언어 모델이 존재하는데 이 연구에서는 그 중 BERT를 사용하였다. BERT를 사용하여 API sequence 데이터를 학습하면 의미와 순서 정보를 포함해서 학습할 수 있으며, 대량의 말뭉치로 학습한 pre-trained 모델이 배포되어 있으므로 그것을 사용하여 API sequence 데이터를 fine-tuning 하였다.

3.5.4 강화학습 모델

강화학습 모델은 agent가 action에 따라 보상(reward)을 주면서 학습시키는 알고리즘이다. 이 연구에서는 강화학습 중 DQN을 사용하여 PE 데이터의 최적 feature set을 찾는 작업을 수행하였다.

3.6 데이터 통계 분석

다차원 분석을 진행하여 통계 분석, 이상치 분석, XAI 분석 등으로 데이터를 분석하여 인공지능 모델에게 영향을 주는 데이터를 찾아내는 등의 과정을 진행한다. 정상 파일과 악성 파일의 데이터 분포를 통계적 관점에서 분석하여 데이터 현황 및 결측값, 이상치 등을 빠르게 찾고 처리할 수 있다 (Table 4 참고).

Table 4. Statistical data for data analysis

Type	Description
count	Number of data
mean	mean
std	Standard Deviation
min	minimum value
25%	1st quartile
50%	2nd quartile, median
75%	3rd quartile
max	maximum value

3.7 시각화 분석

시각화는 빅데이터를 이용한 탐색적 분석에서 많은 양의 데이터를 한눈에 파악할 수 있는 필수적인 요소이다. 전문지식 없는 사용자에게도 쉽게 설명하고 전달할 수 있다. 시각화를 통해 데이터 분석, 모델 분석, 오탐 분석을 수행하여 모델 예측에 대한 투명성과 설명 가능성을 제공하였다. 전체적인 흐름을 시각적으로 파악하고 데이터와 모델에 대한 통찰을 발견할 수 있어 의사 결정을 도와준다.

Table 5. Single model performance comparison

File type	Analysis type	Extraction method	Accuracy					Runtime	Resource utilization
			LGB	XGB	RF	CNN	LSTM		
EXE	Static	PE	93.55%	94.55%	95.21%	-	-	low	low
		Byte	-	-	-	84.92%	-	high	high
	Dynamic	Dynamic info	93.08%	93.68%	94.12%	-	-	low	low
		API sequence	95.89%	96.29%	97.59%	97.44%	96.93%	middle	high
Document	Static	PDF	99.98%	99.95%	99.98%	-	-	low	low
		hwp	99.92%	99.98%	99.98%	-	-	low	low
		MS Ofiice	99.98%	100%	100%	-	-	low	low

데이터의 범위와 분포를 볼 수 있는 히스토그램, 파이 그래프 등으로 시각화된 분석이 가능하고, 이상치와 데이터 범위를 표현하여 신규, 변종 데이터를 식별할 수 있는 박스플롯을 시각화 하였다.

본 논문에서 제안한 프레임워크는 다양한 시각화 유형을 데이터의 종류나 특성별로 비교할 수 있도록 제공한다.

3.8 XAI 분석

XAI를 사용하면 인공지능 모델을 설명할 수 있어 신뢰성 있는 인공지능을 만들 수 있다[32, 33]. 이 연구에서는 모델 분석 및 오탐 분석에 XAI를 적용하여 블랙박스인 인공지능 추론 모델을 검증하고 오탐 원인을 찾아 분석한다.

XAI는 심층 설명학습, 해석 가능한 모델, 모델 귀납으로 나눌 수 있다. 심층 설명학습은 심층 신경망이 설명 가능한 특징들을 학습할 수 있게 하여 은닉층을 통해 모델의 판단 근거를 확인할 수 있고, 해석 가능한 모델은 구조화되고 해석 가능한 인과관계가 있는 모델을 만들어 모델 판정 근거를 찾는 것이다.

모델 귀납은 임의의 블랙박스 모델을 설명 가능한 모델과 상호대조 및 추론하는 기술로, 본 연구는 모델 귀납 방법 중 SHAP과 PDP를 사용한다. SHAP은 피처에 따른 예측 변화를 측정하고, 피처의 중요도, 예측에 영향을 준 피처와 그 값을 파악할 수 있으며, PDP는 피처 값의 변화에 따른 예측 변화를 측정할 수 있다.

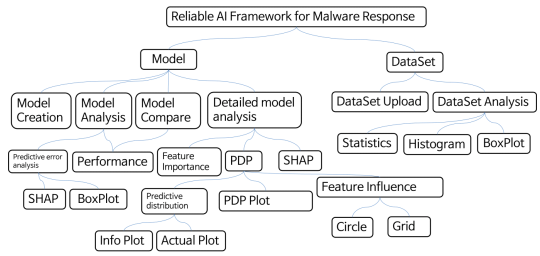

Fig. 8. Trustworthy AI framework for malware response

3.9 최종 분석

신뢰할 수 있는 모델을 만들기 위해 데이터 셋 분석과 XAI에 의한 모델 검증과 오탐 분석 시각화는 Fig.8.과 같이 구성하였다.

다양한 방법의 학습 데이터 추출과 학습 알고리즘을 적용하여 모델을 생성해 보았다. 프레임워크를 위한 적절한 학습 데이터 추출 방법과 최적의 모델을 선별하는 작업이 필요하다. 모델의 선택 기준은 정확도, 수행 시간, 리소스 사용률을 고려하여 선택하였다.

단일 모델에 대해서 각 추출 방법 및 모델 유형에 따른 실험 결과는 Table 5와 같다.

먼저 exe 파일에 관한 결과를 보면, PE와 동적 분석 추출 방식은 실행 시간도 적고, 자원 사용률도 낮아 적절하다. 그러나, byte 정보 추출의 경우에는 학습하는 데 시간이 오래 걸리고, 이미지 학습인 만큼 자원 사용률이 높은 것에 비해 정확도가 저조하다. 또한, byte 정보는 exe 파일이 패킹된 경우 데이터가 오염되어 학습 데이터로 적절하지 않다. API sequence의 경우에는 머신러닝 모델인 LGB,

XGB, RF에서는 학습 시간이 빨랐지만, 딥러닝 모델인 CNN과 LSTM에서는 학습이 오래 걸렸다. 또한, 필요한 sequence 길이가 최소 2,000이다 보니, 자원 사용률이 높았다.

문서 파일 PDF, HWP, MS-Office 모델은 빠른 실행 시간과 낮은 자원 사용률, 높은 정확도를 보였다.

모델의 평가 결과를 바탕으로 단일 모델의 프레임워크에 적용된 추출 방법은 PE, 동적 분석, API Sequence, 문서 파일의 스트림 키워드 추출 방법이고, 빠르게 학습할 수 있는 LGB, XGB, RF 알고리즘을 적용하였다.

Table 6. Multi model performance comparison

Model type	Before	After	Runtime	Resource utilization
Score model	93.55%	95.28%	low	low
	93.83%			
DL model	84.92%	88.64%	high	high

복합 모델에 대한 결과는 Table 6과 같다. Fig.7.과 같이 모델을 병합하는 방법은 딥러닝을 이용하기 때문에 학습 시간이 많이 소요되며, 여러 제약 사항들이 존재한다. 또한, Fig.6.의 방법은 개별 모델을 생성할 수 있기 때문에, 서로 다른 두 가지 머신러닝 모델의 regression score를 사용하여 모델을 통합하는 방법을 프레임워크에 적용하였다.

Table 7. BERT model performance comparison

Epoch	Accuracy	Runtime	Resource utilization
1	88%	high	high
2	90%		
3	91%		
4	91%		
Test	92%		

언어 모델인 BERT에 대한 성능은 Table 7과 같다. BERT 모델의 성능은 90% 초반의 성능으로 다른 모델보다 성능이 우수하지 않고, 학습을 위한 토크나이징 등의 전처리 과정과 학습 과정이 오래 걸리기 때문에 프레임워크에 적용하기에는 적합하지 않은 모델이라고 판단하였다. 따라서 BERT에 대한 기능은 추후 연구를 통하여 최적화 방법을 찾기로 한다.

Table 8. Reinforcement learning model performance comparison

	Feature count	Accuracy	Runtime	Resource utilization
Before	144	95.21%	high	middle
After	10	94.33%		

강화학습을 적용한 결과는 Table 8와 같다. 기본 feature set과 강화학습을 이용하여 찾은 feature set에 대한 비교이다. 원래 피처 개수인 144개에 비해 10개의 피처로도 상당히 좋은 성능을 보이고 있지만, 학습 시간이 매우 길어 강화학습을 이용해 피처를 찾는 것보다는 XAI 분석을 통한 방법이 더 적절하다고 판단하였다. 따라서 강화학습은 현재 프레임워크에는 적용하지 않고 추후 최적화 연구를 진행하여 적용할 예정이다.

Table 9. XAI model performance comparison

	Feature count	Accuracy	Runtime	Resource utilization
Before	144	95.21%	middle	middle
After	90	95.52%		

XAI를 이용하여 오탐 분석을 진행한 후 측정한 성능은 Table 9와 같다. SHAP과 PDP를 이용한 시각화 분석을 통해 오탐을 분석하면, 사용자가 오탐이 일어난 원인 파악을 쉽게 수행하게 할 수 있다. 또한, 직관적인 그래프로 각 피처의 모델에 대한 영향력을 파악할 수 있어, 영향력이 높은 피처를 쉽게 찾을 수 있다. 오탐 원인 파악이나 피처의 영향도는 모델 성능을 개선시켜 신뢰성 있는 인공지능을 제공할 수 있다.

IV. 악성코드 탐지 AI 프레임워크

4.1 개요

본 연구에서는 신뢰성 있는 AI 모델의 지속적인 관리와 안정적인 운영을 목적으로 MLOps 개념을 도입하여 프레임워크를 구성하였다. MLOps가 기존 DevOps(개발)와 다른 점은, 개발 외에 데이터 의존성이 강한 AI 모델의 특성을 반영하여 데이터 파

이프라인을 연계하고, 급변하는 시장 상황에 신속한 대응이 가능하도록 서비스 운영을 연계한다는 점이다(Fig. 9)[22].

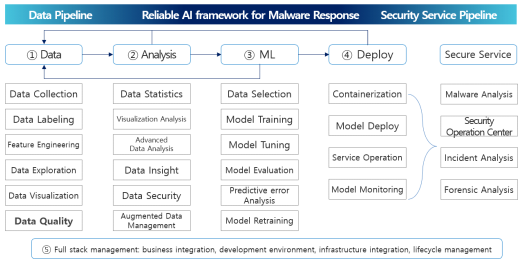

Fig. 9. MLOps for malware response

PE 악성코드와 문서형 악성코드 대응 서비스를 위하여 MLOps의 데이터와 운영을 통합한 신뢰할 수 있는 악성코드 대응 AI 프레임워크(Fig. 10.)를 설계한다. 개발, 통합, 테스트, 출시, 배포, 인프라 관리 등 ML 시스템 구성의 모든 단계에서 모델 재학습 자동화 및 모델 모니터링을 지원한다.

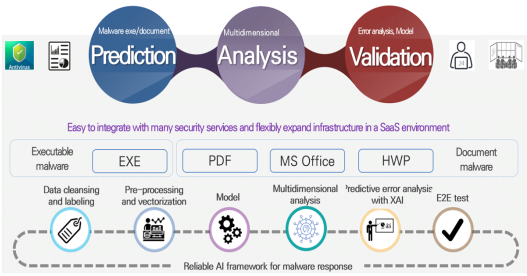

Fig. 10. AI framework for malware detection, analysis and verification

AI 프레임워크의 인프라(Fig. 11.)는 빅데이터 처리를 위한 NoSQL 서버, 악성코드 분석 서버, 인공지능 서버, 서비스 운영을 위한 SaaS 웹 서버, 웹 서비스용 관계형 DB(Maria DB)로 구성된다. 악성코드의 실시간 동적 분석을 위하여 가상 머신을 병렬 구성하였으며, 외부 API와 연결하여 지속적인 데이터 수집과 데이터 정제의 파이프라인을 구성하였다. 또한, 클라우드 SaaS 방식과 표준 API를 사용하여 다른 보안 서비스와 연계되도록 하였으며, 클라우드 방식으로 인프라 확장을 용이하게 하였다.

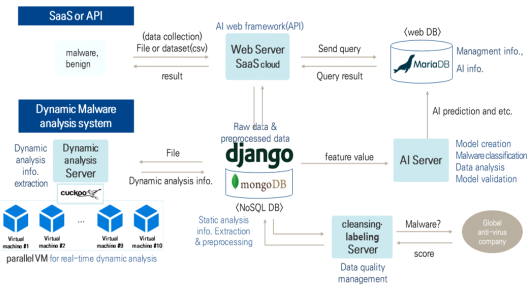

Fig. 11. Infrastructure of AI framework

악성코드 대응 AI 프레임워크는 데이터 수집부터 정제, 피처 엔지니어링, 단일/복합 모델 생성, 데이터 및 오탐 분석과 피드백 통합 분석 등 머신러닝 전 과정을 자동화하는 지능화 프레임워크로 동작한다(Fig. 10.). 프레임워크에 적용된 기술은 Fig. 12.와 같다.

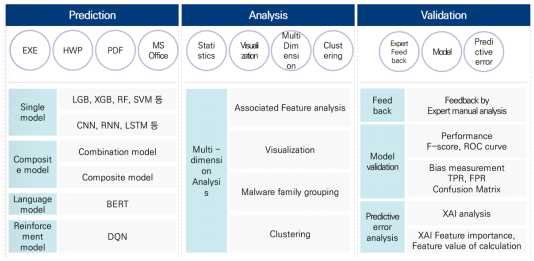

Fig. 12. AI model for malware detection

4.2 데이터

데이터 관련 섹션은 크게 데이터 업로드와 데이터 분석으로 나뉘고, 데이터 분석 섹션은 통계, 히스토그램, 박스 플롯 등을 시각화한다(Fig.13.)

4.2.1 업로드

분석, 모델 생성, 모델 테스트 등에 사용할 데이터셋을 업로드한다.

4.2.2 데이터 분석

업로드된 데이터셋을 분석하여 최솟값, 최댓값, 평균값, kurt, Q1, Q2, Q3, skew, 이상치 비율 등을 통계, 박스 플롯, 히스토그램(Fig.13. data analysis) 등의 형태로 출력한다.

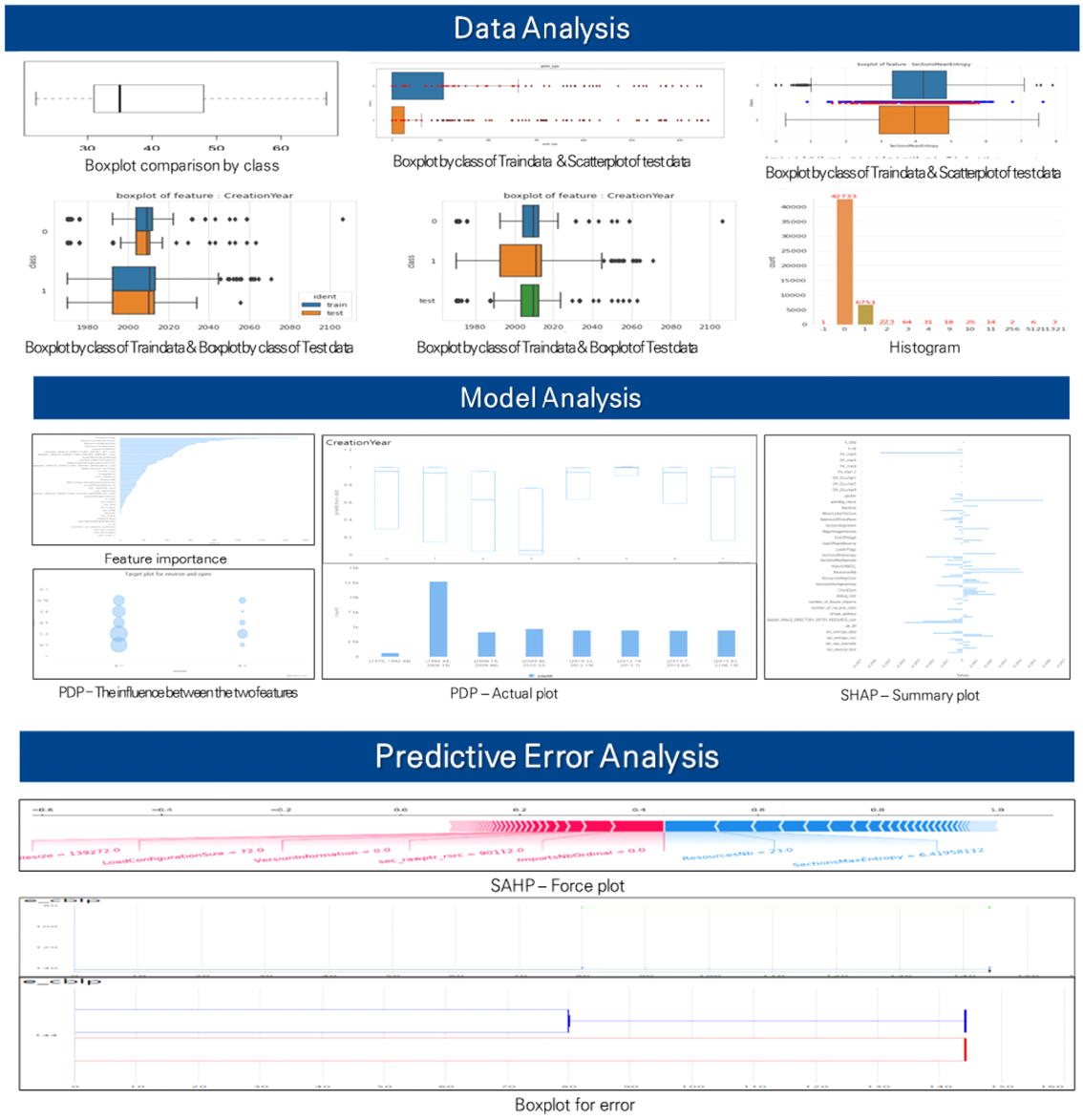

Fig. 13. Visualization for data analysis, model analysis, and predictive error analysis

4.3 모델

모델 관련 섹션은 크게 모델 생성, 모델 비교, 모델 분석, 모델 상세 분석으로 나뉜다(Fig.13. model analysis).

4.3.1 모델 생성

모델 종류와 학습할 피처를 선택하여 모델을 생성한다. 데이터 종류별로 본 연구에서 구한 최적의 피처 셋으로 학습할 수 있고, 원하는 피처를 선택하여 모델을 학습할 수도 있다.

4.3.2 모델 분석

모델 섹션에서는 모델 생성과 모델 비교, 모델 분석, 모델 상세 분석으로 나뉜다(Fig.13. model analysis). 모델 생성에서는 모델 종류와 학습할 피

처를 선택하여 모델을 생성한다.

모델 분석에서는 Fig.13. model analysis와 같이 모델 분석을 위한 다양한 시각화를 제시한다. 생성된 모델의 feature importance를 가져와서 피처 중요도를 확인할 수 있고(Fig.13. model analysis 왼쪽 상단 그림), XAI 기법 중 PDP를 이용하여 두 피처 간의 모델에 끼치는 영향도를 확인할 수 있으며(Fig.13. model analysis 왼쪽 하단 그림), 피처 값의 구간에 따라 피처 영향도를 확인할 수 있다(Fig.13. model analysis 중앙 그림). 또한, XAI 기법 중 SHAP을 이용하여 모델에 사용된 피처의 기여도를 확인할 수 있다(Fig.13. model analysis 오른쪽 그림).

4.4 테스트

4.4.1 테스트

테스트는 테스트와 오탐 분석으로 구성하였다. 테스트는 이전 단계에서 생성된 모델을 이용하여 테스트를 진행한다. 성능지표로 Accuracy와 Precision, Recall, F1-score 등을 사용해 데이터 편향성, 과적합 등을 확인할 수 있다.

오탐 분석은 테스트에서 발생한 예측 오류를 분석한다(Fig. 13. predictive error analysis). 박스플롯과 사분위수를 이용하여 이상치가 발생한 피처 개수를 확인했고 SHAP을 이용하여 각 피처의 기여도를 확인할 수 있다.

Table 10. Assessment result by Certification body

Type	Assessment item	Accuracy	
Malware detection accuracy	MS Kaggle challenge	99.49%	
	KISA challenge	99.00%	
	exe from Global	97.93%	
	exe from Domestic	95.59%	
	Packing and Anti-VM	95.58%	
	hwp	95.57%	
	PDF	99.52%	
	MS Office	99.85%	
Predicted time	exe predicted time	127.66s	
	Document predicted time	HWP	8.66s
		PDF	2.33s
		Office	14s

4.5 AI 악성코드 탐지 모델과 검증

exe, PDF, hwp, MS Office 형식의 파일들을 대상으로 악성코드 탐지 모델을 설계, 개발하였다. 또한 데이터 분석과 모델 분석을 위한 시각화와 분석 기능을 추가하였다(Table 11.).

Table 11. Visualization and Analysis

Type	Assessment item	Measurement conditions/ environment
Inference (detection model)	Local pattern and time series pattern combination algorithm	Combining API n-gram with API sequence model
	Model type	Number of EXE malware detection models
		Number of document malware detection models
Multi-dimensional analysis	Data analysis type	Number of data analysis types that can be analyzed
	Visualization graph type	Number of visualization graphs available

4.5.1 모델 검증

신뢰할 수 있는 AI 모델을 입증하기 위하여 다양한 데이터셋을 이용하여 데이터 독립적인 모델의 일반화 성능을 검증하고자 하였다. 모델 검증은 탐지 정확도로 측정하였으며(Table 10 참고), 탐지 정확도 목표에는 세계적인 악성코드 탐지 대회인 MS 캐글 데이터로 최고 성능 99%, 국내 KISA 대회 데이터로 최고 성능 98%, 국내외 범용데이터 및 최신 데이터를 활용하여 93%, 난이도가 높은 패킹 및 난독화 악성코드 탐지율 90%로 설정하였다. 또한,

상용화를 위한 실시간 탐지 속도로 exe 예측 시간 180초와 문서 예측 시간 120초, 정적 데이터 추출 성공률 98%를, 동적 데이터 추출 성공률 80%를 목표로 설정하였다. 모든 성과지표는 국내 안티바이러스 성능 평가기관을 통해 평가의 공정성을 확보하였다.

V. 결 론

본 논문에서는 악성코드 대응 자동화를 위해 신뢰할 수 있는 인공지능 모델을 만들고, 모델의 생명주기 관리를 위한 데이터와 모델, 비즈니스의 통합 프레임워크를 제안하였다.

데이터 추출, 피처 선택, 모델 생성, 데이터 및 오탐 분석을 자동화하여 사용자의 편리성을 높이고, AI 전문지식이 없어도 직접 악성코드 탐지 모델을 만들고 분석할 수 있도록 하였다.

신뢰할 수 있는 인공지능 모델 개발을 위하여 본 논문에서는 견고를 제외한 안전과, 설명 가능, 투명, 공정의 4가지 요소를 구현하였다. 특히 외부 기관의 검증을 통해 일반화 성능의 안정성을 입증하였고 투명을 포함한 설명 가능에 중점을 두어 개발하였다.

클라우드 SaaS 방식과 표준 API를 사용하여 다른 보안 서비스 연계나 인프라 확장을 쉽게 하여서 exe 실행형 악성코드와 문서형 악성코드 대응 서비스를 구축하였다. 이렇게 운영되는 보안 서비스는 데이터 수집원이 되어 다양한 데이터를 수집할 수 있게 하고, 외부 API를 통해서 라벨링과 정제를 위한 정보를 가져오는 데이터 파이프라인을 구성하였다.

본 논문에서 제시하는 신뢰할 수 있는 인공지능 프레임워크를 통해 안정적인 악성코드 대응 서비스뿐만 아니라, AI 생명주기 관리를 통해 인공지능 도입의 시행착오를 줄이는 계기가 되기를 바란다.

References

[1] Gibert, D., Mateu, C., & Planes, J., "The rise of machine learning for detection and classification of malware: Research developments, trends and challenges," Journal of Network and Computer Applications, vol. 153, pp. 102526, Jan. 2020.

[2] K. P. Briefing, "Trusted artificial intelligence realization strategy," https://www.msit.go.kr/bbs/view.do?sCode=user&mId=113&mPid=112&pageIndex=&bbsSeqNo=94&nttSeqNo=3180239&searchOpt=ALL&searchTxt=, Sep. 2022.

[3] C. P. SungIk Jo, WoongSik Yoo, "Implementation strategies for machine learning operations (mlops)," Institute for Information & communication Technology Planning & evaluation (IITP), 1985(1), pp. 2-16, Feb, 2021.

[4] Kang, B., Yerima, S. Y., Sezer, S., & McLaughlin, K., "N-gram opcode analysis for android malware detection," arXiv preprint arXiv:1612.01445. 2016.

[5] Santos, I., Brezo, F., Ugarte-Pedrero, X., & Bringas, P. G., "Opcode sequences as representation of executables for data-mining-based unknown malware detection," Information Sciences, vol. 231, pp. 64-82, May. 2013.

[6] Saxe, J., & Berlin, K., "Deep neural network based malware detection using two dimensional binary program features," In 2015 10th International Conference on Malicious and Unwanted Software (MALWARE), pp. 11-20, Oct. 2015.

[7] Shabtai, A., Moskovitch, R., Feher, C., Dolev, S., & Elovici, Y., "Detecting unknown malicious code by applying classification techniques on opcode patterns," Security Informatics, vol. 1, no. 1, pp. 1-22, Feb. 2012.

[8] Kumar, N., Mukhopadhyay, S., Gupta, M., Handa, A., & Shukla, S., "Malware Classification using Early Stage Behavioral Analysis," IEEE Conference Publication | IEEE Xplore., pp. 16-23, Aug. 2019.

[9] Han, W., Xue, J., Wang, Y., Huang, L., Kong, Z., & Mao, L., "MalDAE: Detecting and explaining malware based on correlation and fusion of static and dynamic characteristics," Computers & Security, vol. 83, pp. 208-233, Jun. 2019.

[10] Fuyong, Z., & Tiezhu, Z., "Malware detection and classification based on n-grams attribute similarity," In 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC), vol. 1, pp. 793-796, Jul. 2017.

[11] Mikolov, T., Chen, K., Corrado, G., & Dean, J., "Efficient estimation of word representations in vector space," arXiv preprint arXiv:1301.3781, 2013.

[12] Pascanu, R., Stokes, J. W., Sanossian, H., Marinescu, M., & Thomas, A, "Malware classification with recurrent networks," In 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1916-1920, Apr. 2015.

[13] Hardy, W., Chen, L., Hou, S., Ye, Y., & Li, X., "DL4MD: A deep learning framework for intelligent malware detection," In Proceedings of the International Conference on Data Science (ICDATA). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), pp. 61-67, 2016.

[14] Jung, B., Kim, T., & Im, E. G., "Malware classification using byte sequence information," In Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems, pp. 143-148, Oct. 2018.

[15] Gibert, D., "Convolutional neural networks for malware classification," University Rovira i Virgili, Tarragona, Spain, pp. 1-98, Oct. 2016.

[16] Kolosnjaji, B., Eraisha, G., Webster, G., Zarras, A., & Eckert, C, "Empowering convolutional networks for malware classification and analysis," In 2017 International Joint Conference on Neural Networks (IJCNN), pp. 3838-3845, May. 2017.

[17] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., Nicholas, C. K., "Malware detection by eating a whole exe," In Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence, pp. 268-276, Jun. 2018.

[18] "Word2vec: Google's New Leap Forward on the Vectorized Representation of Words," Medium, 2. Jun. 2018.

[19] Le, Q., Boydell, O., Mac Namee, B., & Scanlon, M, "Deep learning at the shallow end: Malware classification for non-domain experts," Digital Investigation, vol.26, pp. S118-S126, Jul. 2018.

[20] Trong Duc Nguyen, Anh Tuan Nguyen, Hung Dang Phan, and Tien N Nguyen, "Exploring api embedding for api usages and applications," In 2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE), pp. 438 - 449, Jul. 2017.

[21] Lee, Y., Kwon, H., Choi, S. H., Lim, S. H., Baek, S. H., & Park, K. W., "Instruction2vec: efficient preprocessor of assembly code to detect software weakness with CNN," Applied Sciences, vol. 9, no. 19, pp. 4086, Sep. 2019.

[22] Cho, S., Yoo, W., Pyo, C., "Machine Learning Operations (MLOps)

Implementation Strategy," https://kosen.kr/file/down/FILE_00000000000051750/1, IITP, weekly tech trend, pp.2-16, Feb. 2021.
〔23〕 KIAT〔Agile〕, "Driving innovation in AI development, what is 'mlops'?" https://www.kiat.or.kr/commonfile/fileidDownLoad.do?file_id=58516, Aug. 2022.
〔24〕 S. Poudyal and D. Dasgupta, "Ai-powered ransomware detection framework," in 2020 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1154 - 1161, Dec. 2020.
〔25〕 O. Aslan and A. A. Yilmaz, "A new malware classification framework based on deep learning algorithms," Ieee Access, vol. 9, pp. 87936 - 87951, Jun. 2021.
〔26〕 C. Urcuqui-L ´opez and A. N. Cadavid, "Framework for malware analysis in android," Sistemas y Telem´atica, vol. 14, no. 37, pp. 45 - 56, Aug. 2016.
〔27〕 L. Caviglione, M. Gaggero, J.-F. Lalande, W. Mazurczyk, and M. Urba ´nski, "Seeing the unseen: revealing mobile malware hidden communications via energy consumption and artificial intelligence," IEEE Transactions on Information Forensics and Security, vol. 11, no. 4, pp. 799 - 810, Dec. 2015
〔28〕 Virusshare, "Forensics, C.: Virusshare", https://virusshare.com/, Sep. 2022.
〔29〕 VirsuTotal, "Virustotal.com", https://www.virustotal.com/, Sep. 2022.
〔30〕 Khalid, S., Khalil, T., & Nasreen, S., "A survey of feature selection and feature extraction techniques in machine learning," In 2014 science and information conference, pp. 372-378, Aug. 2014.
〔31〕 Peng, H., Long, F., & Ding, C., "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on pattern analysis and machine intelligence, vol. 27, no. 8, pp. 1226-1238, Jun. 2005.
〔32〕 A. B. Arrieta, N. D´ıaz-Rodr´ıguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garc´ıa, S. Gil-L´opez, D. Molina, R. Benjamins, et al., "Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai," Information fusion, vol. 58, pp. 82 - 115, Jan. 2020.
〔33〕 Lim, H., Kim, S., "The need and research trends for explainable AI (XAI)," Journal of Information Technology and Applied Engineering (JITAE), 12(1), pp. 51-57, Apr. 2022.

〈저자소개〉

신 경 아 (Kyounga Shin) 종신회원
2019년 2월: 고려대학교 정보보호대학원 정보보호학과 박사 수료
2016년 2월: 에프원시큐리티 보안서비스본부 상무
현재: 포테이토넷 대표
〈관심분야〉 정보보호, 악성코드, 위협인텔리전스, 인공지능

이 윤 호 (Yunho Lee) 정회원
2022년 3월: 고려대학교 정보보호대학원 정보보호학과 석사 입학
2020년 5월~2022년 8월: 에프원시큐리티 보안서비스본부 대리
〈관심분야〉 정보보호, 인공지능

배 병 주 (Byeongju Bae) 정회원
2019년 2월: 서울호서전문학교 정보보호학 학사
2018년 1월~현재: 에프원시큐리티 보안서비스본부 과장
〈관심분야〉 정보통신, 인공지능

이 수 항 (Suhang Lee) 정회원
2021년 2월: 서울호서전문학교 정보보호학 학사
2020년 9월~2022년 8월: 에프원시큐리티 보안서비스본부
〈관심분야〉 정보보호, 인공지능

홍 희 주 (Heeju Hong) 정회원
2021년 6월: Northern Alberta Institute of Technology 졸업
2020년 5월~2022년 9월: 에프원시큐리티 보안서비스본부
〈관심분야〉 정보통신, 인공지능

최 영 진 (Youngjin Choi) 정회원
2020년 2월: 충훈고등학교 졸업
2022년 2월~현재: 에프원시큐리티 보안서비스본부
〈관심분야〉 정보통신, 인공지능

이 상 진 (Sangjin Lee) 종신회원
1989년 2월~1999년 2월: 한국전자통신연구원 선임 연구원
1999년 2월~현재: 고려대학교 교수
2008년 3월~현재: 고려대 정보보호연구원 디지털포렌식연구센터장
2017년 3월~현재: 고려대학교(정보보호대학원) 원장
〈관심분야〉 대칭키 암호, 정보은닉이론, 컴퓨터 포렌식