정 보 보 호 학 회 지
제 31 권 제5호, 2021. 10

정보보호 분야의 XAI 기술 동향

김 홍 비*, 이 태 진**

요 약

컴퓨터 기술의 발전에 따라 ML(Machine Learning) 및 AI(Artificial Intelligence)의 도입이 활발히 진행되고 있으며, 정보보호 분야에서도 활용이 증가하고 있는 추세이다. 그러나 이러한 모델들은 black-box 특성을 가지고 있으므로 의사결정 과정을 이해하기 어렵다. 특히, 오탐지 리스크가 큰 정보보호 환경에서 이러한 문제점은 AI 기술을 널리 활용하는데 상당한 장애로 작용한다. 이를 해결하기 위해 XAI(eXplainable Artificial Intelligence) 방법론에 대한 연구가 주목받고 있다. XAI는 예측의 해석이 어려운 AI의 문제점을 보완하기 위해 등장한 방법으로 AI의 학습 과정을 투명하게 보여줄 수 있으며, 예측에 대한 신뢰성을 제공할 수 있다. 본 논문에서는 이러한 XAI 기술의 개념 및 필요성, XAI 방법론의 정보보호 분야 적용 사례에 설명한다. 또한, XAI 평가 방법을 제시하며, XAI 방법론을 보안 시스템에 적용한 경우의 결과도 논의한다. XAI 기술은 AI 판단에 대한 사람 중심의 해석정보를 제공하여, 한정된 인력에 많은 분석데이터를 처리해야 하는 보안담당자들의 분석 및 의사결정 시간을 줄이는데 기여할 수 있을 것으로 예상된다.

Ⅰ. 서 론

ML 및 AI에 기반한 기술의 도입이 활발히 진행되고 있다. 이러한 기술을 편리성 및 높은 정확도로 인해 널리 사용되고 있으며, 보안 분야에서도 악성코드 탐지 등에서 많이 사용되고 있다. McAfee의 2019년 최신 보고서에 따르면 ransomware 공격은 118%, PowerShell 공격은 460% 증가했으며 XMB exploit traffic의 고유한 소스가 400개 이상 발견되었다[1]. 점차 증가하는 위협에 대응하기 위해 전문가들은 다양한 대응 기술에 대해 연구를 진행하고 있다. 악성 위협에 대응하기 위한 방법 중 하나인 signature 기반의 탐지 방법은 높은 정확도로 악성코드의 탐지가 가능하지만 알려진 공격에 대해서만 정확한 탐지가 가능하다는 한계점이 있다. signature 기반 탐지 방법의 한계점을 보완하기 위해 AI가 도입되고 있으며 알려지지 않은 공격에 대한 탐지뿐만 아니라 자동화 특성으로 인해 정보보호 분야에서도 매우 인기를 얻고 있다.

그러나 AI는 많은 매개 변수로 인해 모델이 매우 복잡하고 결과 산출의 이유에 대한 이해 및 이를 추적하는 것조차 어렵다는 문제가 있다. AI의 급속한 확산은 ML의 진보를 기반으로 스스로 인식하고, 학습하고, 결정할 뿐 아니라 행동할 수 있는 자율시스템의 등장을 불러와 유익함을 안겨줄 것으로 예상하고 있으나 한편으로는 이러한 시스템은 사용자에게 스스로의 결정과 행동을 설명할 수 없기 때문에 그에 따른 부작용도 발생해 효율적인 사용에 장애물이 될 것으로 분석하고 있다[2]. 모델의 예측에 대한 이해가 부족한 경우 신뢰성 및 안정성이 중요한 보안 분야에서 광범위하게 적용할 수 없게 된다. 이와 같은 문제를 해결하기 위해 XAI 방법론에 대한 연구가 활발히 진행되고 있다. 그러나 보안 분야에서의 XAI 방법론의 활용은 아직 도입 단계이며, 더 많은 연구가 필요하다. 이에 따라 본 논문은 대표적인 XAI 방법론 및 보안 분야에서의 적용 사례에 대해 연구한다.

2장에서는 XAI의 필요성 및 개념, 용어에 대한 더 자세히 설명한다. 3장에서는 XAI 기술의 정보보호 분야 적용 사례를 소개하며, 4장에서는 XAI 기술의 평가 방법에 따른 결과를 설명한다. 5장에서는 본 논문의 결론 및 향후 기대 효과를 설명한다.

2021년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-00026, 지능화된 악성코드 위협으로부터 ICT 인프라 보호)
* 호서대학교 정보보호학과(대학원생, tlp3a1@gmail.com)
** 호서대학교 정보보호학과(교수, kinjecs0@gmail.com)

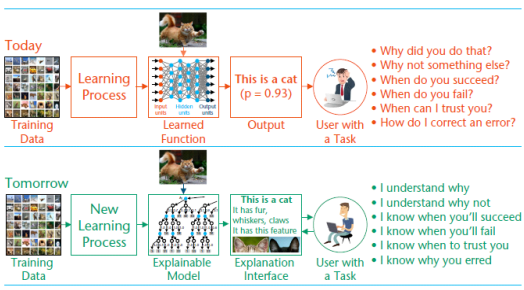
Ⅱ. XAI 개념 및 필요성

XAI는 AI 설명성을 제공한다. AI는 자동화 및 높은 정확도로 인해 많은 관심을 받고 있지만, black-box 특성으로 인해 실질적인 도입에 어려움이 있다. AI는 본질적으로 비선형 회귀모델로 입력과 출력 간의 관계가 선형적이지 못해 입력이 출력에 어떻게 영향을 주는지 직접적으로 알기 어려우며, 기술이 발전함에 따라 AI가 자체적으로 학습에 필요한 feature를 잘 선택할 수 있게 되었기 때문에 분석가는 더 이상 feature를 생성하지 않아도 되어 AI 내부에 대한 해석이 더욱 어렵게 되었다. 이는 분석가에게 편리성을 가져다 주었다는 점에서 매우 유용하였지만, 해석이 가능하지 않아 신뢰성 및 안정성을 보장할 수 없게 되어 결국 분석가는 AI 모델의 대응 후에도 결과에 대한 분석을 진행해야 했다. 이에 AI에 대한 해석을 제공할 수 있는 XAI 기술이 주목받고 있다. XAI는 높은 수준의 성능을 유지하면서도 보다 설명 가능한 모델을 생성함으로써 분석가가 AI를 이해하고 신뢰할 수 있도록 해준다. 인터넷의 핵심 기술을 개발한 미국 국방성의 개발부서 DARPA(Defense Advanced Research Projects Agency)[3]은 XAI의 목적을 크게 세 가지로 정의하였다. 첫 번째는 모델의 복잡성 감소이며, 두 번째는 모델 예측의 신뢰성, 세 번째는 의사 결정을 위한 AI 모델의 활용이다. 그림 1은 DARPA가 제시한 XAI 개념을 나타내는 그림이다.

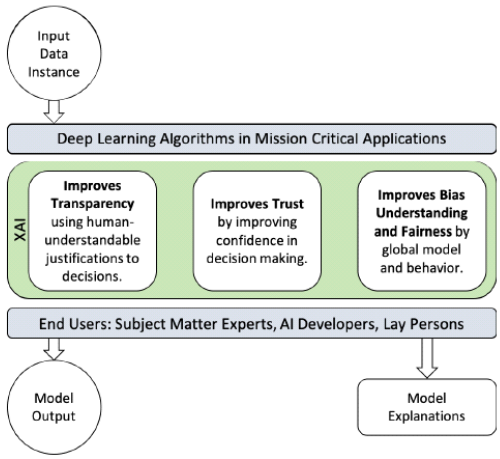
XAI가 필요한 이유[4]에 대해 자세히 살펴 보면 첫 번째로 예측에 대한 신뢰 제공 및 AI 모델의 개선을 위함이다. AI 모델은 수많은 매개 변수 및 복잡한 구조로 인해 black-box 특성을 가지며, 이로 인해 해석이 불가능하고 모델의 예측에 대한 이해가 어렵다, 모델이 어떻게 예측 결과를 산출했는지 해석할 수 있게 되면 예측 결과에 대해 신뢰가 가능하며, 현재의 AI 모델보다 더 향상된 성능으로 탐지가 가능할 수 있기 때문에 AI의 내부 구조에 대한 이해가 필요하다. 두 번째는 AI 모델에서 얻을 수 있는 정보의 활용을 위함이다. 분석가는 결과 예측 뿐만 아니라 AI 모델에서 숨겨진 법칙을 발견할 수 있고, 새로운 통찰력을 얻기 위해 AI 모델에서 정제된 지식을 추출할 수 있다. 하지만, AI 모델의 학습 과정을 상세히 알 수 없기 때문에 AI에서 얻을 수 있는 정보의 활용도 사실상 불가능하다. 세 번째는 AI 모델은 주어진 데이터에 대해서만 추론이 진행되기 때문에 결과가 특정하게 편향될 수 있다. AI가 결정한 최종 결과와 도출 과정에서 학습 데이터가 편향된 데이터를 기반으로 하거나 알고리즘의 악의적 조작이 개입될 경우 편향된 판단을 내릴 위험이 발생한다. 이는 잘못된 결과를 가져올 수 있으며, 예를 들어 정보보호 분야에서 악성코드 탐지 시 악성을 정상으로 탐지하는 등과 같은 부작용을 가져올 수 있다.

따라서, XAI는 크게 투명성, 신뢰성, 편향 개선의 기능을 제공하는 것을 목표로 한다. XAI는 AI가 어떻게 동작했는지에 대한 설명을 제공하여 AI를 투명하게 이해할 수 있도록 하며, AI 예측의 신뢰 정도를 판단할 수 있는 정보를 제공하고, 모델의 편향을 개선할 수 있도록 해준다. 그림 2는 XAI의 투명성, 신뢰성, 편향 개선에 대한 설명을 보여준다[5].

XAI 방법론에서 자주 등장하는 용어들을 다음과 같이 정의 가능하다[5]. “해석가능성 (Interpretability)”은



〔그림 1〕 XAI 개념[3]



〔그림 2〕 XAI 도입에 따른 기대효과[5]

알고리즘 작동 방식을 이해하기에 충분한 표현 데이터를 제공하는 알고리즘의 기능이나 특징이다. “해석(Interpretation)”은 ML 모델에 의해 생성된 출력과 같은 복잡한 영역을 인간이 이해할 수 있고 합리적인 의미 있는 개념으로 단순화한 표현을 의미한다. “설명(Explanation)”은 특정 예측 결과에 대한 입력 인스턴스의 feature 중요도 및 관련성을 설명하기 위해 외부 알고리즘 또는 ML 모델 자체에 의해 생성되는 추가적인 메타 정보를 의미한다. “설명가능성(Explainability)”은 인간과 의사결정자 간의 인터페이스의 개념으로서 의미를 설명하는 능력을 의미한다[6]. “해석가능성”과 “설명가능성”은 다른 의미를 가지는 용어로서 “해석가능성”은 결과를 스스로 이해하고 해석하는 능력을 의미하고, “설명가능성”은 다른 사람에게 결과를 설명하는 능력을 의미한다는 점에서 차이가 있다[7].

Ⅲ. 정보보호 분야의 XAI 방법론 적용 현황

본 장에서는 정보보호 분야에서의 XAI 방법론 적용 현황에 대해 소개한다. XAI 방법론에서 대표적인 방법들에 대해 간단히 설명하며, 정보보호 분야의 AI 도입 확산에 도움을 줄 수 있는 XAI 방법론들을 적용한 정보보호 분야의 사례를 알아본다.

3.1. Traditional method

기존 악성코드 탐지 모델들은 정적 및 동적 분석 방법을 통해 파일로부터 feature를 추출하여 학습을 진행하였다. 이때 추출된 feature들은 악성코드의 동작에 기반하여 생성된 feature들로서 분석가는 이를 통해 예측에 대한 이해가 가능하며 이를 이용하여 학습이 진행되기 때문에 결과에 대한 해석 또한 가능하다.

기존 탐지 모델 중 하나인 rule 기반 모델은 분석가의 이해가 가능한 rule을 기반으로 탐지를 진행하는 모델로서 예측을 해석하는 것은 분석가의 지식에 의해 쉽게 검증 가능하다. Zhen 외 1인은 Fuzzy Logic 기반 rule의 형태로 interpretability-oriented layer가 추가된 CNN(Convolutional Neural Network) 학습 구조를 제안했다[8]. 사용자는 DL(Deep Learning) 구조에서 언어적 Fuzzy logic 기반 rule을 추출하고 이 정보를 전처리를 통해 파생된 feature에 연결하여 전체 분류의 해석 가능성을 향상시킬 수 있다.

Alan 외 2인은 NODENS라는 경량 악성코드 탐지 시스템을 제안했으며, 더불어 예측 결과의 투명성을 제공하여 최종 사용자가 예측의 결과에 대한 이유를 알 수 있도록 하였다[9]. NODENS는 원시 데이터 분석에서 도출된 평가에 추가 가중치를 제공하고 이해하기 쉬운 출력 형식을 사용하여 분석에 도움이 되는 결정적인 해석 가능성을 제공하였다. 악성코드의 구별을 위해 이진 값을 활용하였으며 이진 값은 정상에서는 찾아볼 수 없는 명확한 동작 패턴을 제공하기 때문에 설명이 가능하다.

그러나, 이러한 방식은 알려진 공격에 대해 탐지 정확도가 높으며, 해석이 가능하다는 장점이 있지만 알려지지 않은 공격에 대해서는 탐지가 어렵다는 이면이 존재한다. 이에 AI 기반 탐지 모델들의 활용이 증가하고 있으며 이러한 모델들의 해석을 제공하기 위한 방법론들이 연구되었다.

3.2. LRP(Layer-wise Relevance BackPropagation)

LRP는 NN(Neural Network) 모델에서 결과를 역추적해 입력 데이터의 개별 feature에 대한 기여도를 계산하는 방법으로, 2015년에 Bach 등이 도입하였다[10]. DL 모델의 예측 결과를 분석하여 입력 데이터의 개별 feature에 대한 기여 점수를 도출하는데 사용된다. 각 입력에 대한 기여 점수는 결과 클래스 노드의 클래스 점수를 입력 계층으로 역전파하여 계산된다.

입력 인스턴스 x, 선형 출력 y 및 활성화 출력 z가 있는 간단한 NN을 고려하면 시스템은 다음 식(1)과 같이 설명될 수 있다[5].

$$y_j = \sum_i w_{ij} x_i + b_j$$
$$z_j = f(y_i) \quad (1)$$

이 방법은 예측에 대한 개별 입력변수의 기여도를 시각화하고 이해하는데 도움이 되는 방식으로 기여도는 top-down 방식으로 각 뉴런의 출력단에서 입력단 방향으로 재분배되며, DL 모델의 부분 모듈인 각 계층의 기여도를 히트맵 형태로 시각화하여 직관적으로 이해할 수 있다[2].

Aditya와 Nhien-AN은 사이버 보안 영역과 관련된 다양한 보안 속성 및 위협 모델을 다루는 설명 가능한 AI 방법에 대한 분류법을 제안하였다[11]. 3가지

Id	LRP
0	createdate_ts
1	creator_dot
2	creator_lc
3	creator_uc
4	moddate_ts
5	producer_uc
6	title_dot
7	title_lc
8	title_oth

Id	LRP
0	createdate_ts
1	creator_dot
2	creator_lc
3	creator_uc
4	moddate_ts
5	producer_uc
6	title_dot
7	title_lc
8	title_oth

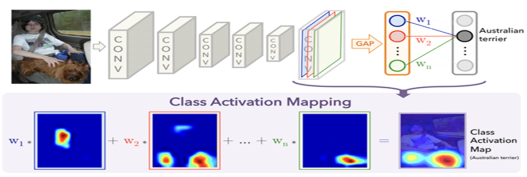
〔그림 3〕 악성 PDF 파일에 대한 explanation map (좌), Mimicus System에 대한 IC- 공격(우)

보안 관련 dataset 및 모델에 대해 제안된 시스템을 검증하였고 제안된 공격에 의해 XAI 방법의 보안 속성이 손상될 수 있음을 보여주었다. 공격에 대한 정성적 평가를 위해 설명법의 관련성 벡터를 시각화하였으며 그림 3이 이를 나타낸다. 모델 결과와 설명법의 결과가 일치하면 녹색, 일치하지 않으면 붉은색으로 표시되며 밝기는 feature의 중요성을 나타낸다.

3.3. CAM(Class Activation Mapping)

CAM은 Input-Conv-FC 계층로 이루어져 있는 CNN이 구조에서 FC-계층의 구조를 조금씩 바꾸면서 기존에 잃었던 위치 정보들을 얻어내는 기술이다[12]. 마지막 Conv-계층을 FC-계층으로 바꾸는 대신에, GAP(Global Average Pooling)을 적용하면 별다른 추가 지도학습 없이 CNN이 특정 위치들을 구별하도록 만들 수 있다는 것이다. GAP layer는 입력 이미지의 모든 값의 평균을 출력한다. 아래 그림 4는 CAM의 기본 구조를 나타낸다.

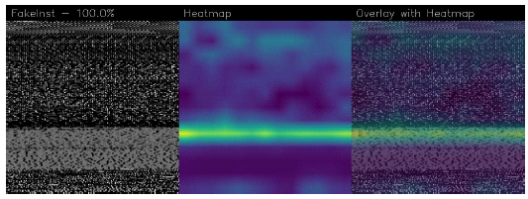
특정 예측에 대한 CAM은 결과 라벨을 식별하는 입력 이미지의 구별 영역을 보여주며, 간단히 계산할 수 있다는 이점이 있다. 그러나, GAP 계층을 사용해야만 한다는 한계점을 갖고 있다. GAP 계층을 사용할 경우 fine tuning 과정이 필수적이기 때문에 모델을 다시 설계하고 학습해야 하며, 마지막 Conv-계층에 대해서만 CAM을 추출할 수 있다.



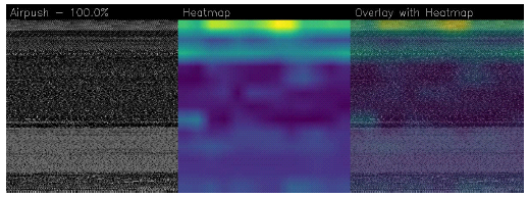
〔그림 4〕 CAM 구조

이러한 CAM의 한계점을 해결하기 위해 Grad-CAM(Gradient-weighted CAM)이 제안되었다. Grad-CAM은 CAM 마지막 Conv-계층 뒤에 GAP 구조가 필요 없어, CNN의 기본 구조를 변형하지 않고 그대로 사용 가능하다.

Giacomo 외 3인은 DL 기반 악성코드 패밀리 탐지기를 제안하였으며, 보안 분석가에게 예측 분류를 해석하고 예측 신뢰성을 검증을 제공하기 위해 Grad-CAM 기반 활성화 맵에 대해 연구했다[13]. 동일한 패밀리에 속하는 악성코드는 코드의 일부를 공유하므로 이미지 영역이 유사하며, Grad-CAM을 통해 생성 가능한 활성화 맵은 유사한 영역이 강조되기 때문에 예측 결과에 대한 해석이 가능하다. 그림 5는 안드로이드 악성코드 패밀리 FakeInstaller로 올바르게 분류된 FakeInstaller에 속하는 샘플의 활성화 맵 예시를 나타낸다. 진한 파란색 영역은 예측에 영향을 끼치지 않은 영역이며, 녹색 및 노란색 영역은 활성화 징후 영역으로 예측에 영향을 끼친 영역을 나타낸다. 그림 6은 Airpush에 속하는 샘플의 활성화 맵 예시를 나타낸다. 회색조 이미지가 Fakelnsfaller와 유사하게



〔그림 5〕 악성 안드로이드 FakeInstaller 패밀리 예시〔13〕. 악성 안드로이드 FakeInstaller 패밀리 예측 시 노란색(녹색)으로 표시된 부분이 해당 패밀리로의 예측에 영향이 크게 작용했음을 알 수 있으며, 악성 패밀리의 분류가 잘못된 경우 위와 같은 시각화를 통해 원인을 분석하여 모델 개선을 진행할 수 있음



〔그림 6〕 악성 안드로이드 Airpush 패밀리 예시〔13〕. 악성 안드로이드 FakeInstaller 패밀리 예측 시 노란색(녹색)으로 표시된 부분이 해당 패밀리로의 예측에 영향이 크게 작용했음을 알 수 있으며 Fakelnestaller 패밀리와 유사한 이미지이지만 전혀 다른 영역이 결과에 영향을 끼쳤음을 확인할 수 있음

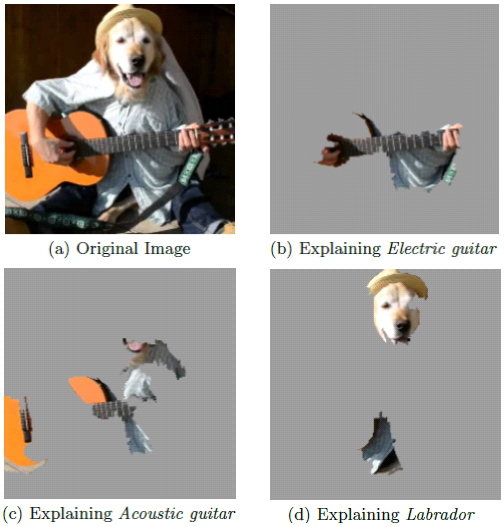
나타나지만 엄연히 다른 패밀리이며, Fakelnsfaller와 다른 영역에 초점을 맞춰 예측이 진행된 것을 확인할 수 있다.

3.4. LIME(Local Interpretable Model-Agnostic Explanations)

LIME은 M. T. Ribeiro 외 2인에 의해 제안되었으며 국지적 단위의 모델을 설명하는 기법이다[14]. 개별 예측의 결과를 설명하기 위해 학습 국지적 대리 모델에 초점을 맞춘다. LIME은 연속 경로의 존재 여부를 나타내는 이진 벡터 $x' \in 0,1$ 또는 클래스 출력에 대한 가장 높은 표현력을 제공하는 'superpixels'을 찾는다. LIME에서 제공하는 설명은 다음과 같은 식(2)을 통해서 주어진다. 여기서 g는 결정 트리, 선형 모델 또는 다양한 해석가능성을 가진 기타 모델들이 될 수 있다. $ohm(g)$는 설명 복잡도를 나타낸다.

$$\xi(x) = \arg(L(f,g,\pi_x) + ohm(g)) \quad (2)$$

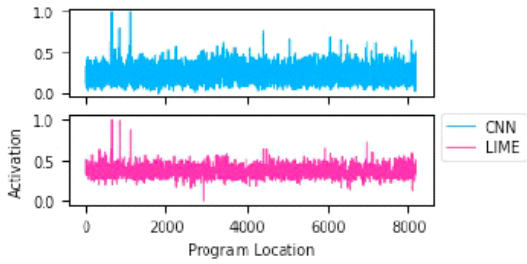
그림 7은 단일 인스턴스에 대한 LIME 알고리즘의 시각화 예시를 나타낸다. 여기에서 상위 3개 클래스는 "electric guitar"($p = 0.32$), "acoustic guitar"($p = 0.24$) 및 "labrador"($p = 0.21$)로 도출되었다. 분류기는 입력 이미지에서 'superpixels' 그룹을 선택하여 상위 예측 라벨에서 시각적 설명을 제공한다.



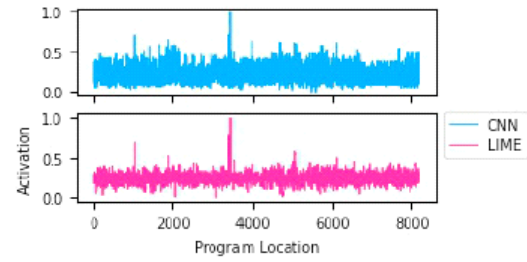
(a) Original Image (b) Explaining *Electric guitar*
(c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

〔그림 7〕 LIME 기반 이미지 분류예측에 대한 설명 예시〔14〕

Marin 외 3인은 안드로이드 악성코드 탐지에서 활용되고 있는 CNN이 예측을 실제로 설명하고 XAI 방법론들과 얼마나 잘 상관되는지에 대한 연구를 진행했다[15]. 악성코드 탐지에 기여하는 것으로 보이는 안드로이드 opcode sequence에서 CNN이 중요하다고 간주하는 위치와 LIME이 중요하게 간주하는 위치를 비교하여 해석을 진행했다. 결과적으로 CNN과 LIME이 중요하다고 간주하는 위치가 거의 일치한다는 것을 증명하였으며, CNN이 악성코드 탐지 작업에 대해 올바른 feature를 학습하고 있다는 확신이 높아지게 되었다. 그림 8과 그림 9는 해당 논문에서 진행한 실험의 일부를 나타낸다. 실험 결과는 CNN과 LIME이 유사한 위치에서 높은 활성화 결과가 산출된 것을 확인할 수 있다.



〔그림 8〕 악성 안드로이드 Gasms 패밀리 예측 시 CNN과 LIME이 중요하게 간주하는 위치 비교 예시〔15〕



〔그림 9〕 악성 안드로이드 Updtkiller 패밀리 예측 시 CNN과 LIME이 중요하게 간주하는 위치 비교 예시〔15〕

3.5. SHAP(SHapley Additive exPlanations)

SHAP은 1953년 Shapley가 처음으로 제안한 방안으로 게임 이론에 기반하며 모델에서 feature의 중요성에 대한 강력하고 통찰력 있는 해석을 제공한다[16]. 2017년에 Lundberg와 Lee가 LightGBM,

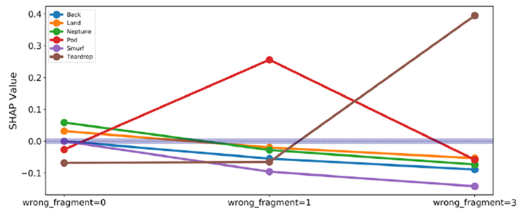
XGBoost, GBoost, CatBoost 및 Scikit-learn 트리 모델을 포함한 다양한 기술에 대해 SHAP를 계산할 수 있는 Python 패키지를 개발하였다[17]. 다음 식(3)은 SHAP 수식을 나타낸다.

$$\varnothing_i = \sum_{S \subseteq F \setminus j} \frac{|S|!(|F|-|S|-1)!}{|F|!} \cdot (v(S \cup j - v(S)) \quad (3)$$

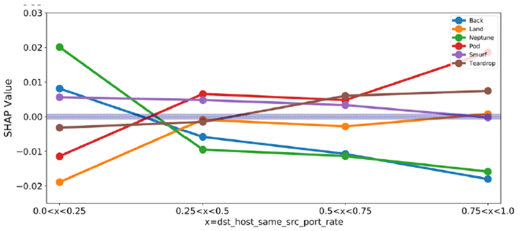
SHAP는 feature들을 추가 및 제거하는 dataset을 만들어 이를 선형 모델로 구성하고 이렇게 구성된 선형 모델의 가중치를 가지고 해석하는 방식으로 '특정 변수가 제거'되면 얼마나 예측에 변화를 주는지 살펴보고, 그에 대한 답을 SHAP 값으로 표현하는데 이때 SHAP 값은 한 예측에서 변수의 영향도를 방향과 크기로 표현한다[2].

Wang 외 2인은 IDS(Intrusion Detection System) 판단에 대한 국지적 및 전역적 설명을 제공하는 프레임워크를 제안했다[18]. NSL-KDD dataset를 이용하여 제안 프레임워크를 검증하였으며, 그림 10은 제안 프레임워크를 통한 DoS 공격에서의 feature 간의 관계를 시각화한 그래프이다. 두 가지 feature에 대해 시각화하였으며, 각 feature와 DoS 공격의 유형 간의 관계를 분석할 수 있도록 한다. 그래프 (a)의 선 그래프에서 wrong_blight=0인 경우 Pod(붉은색) 및 Teardrop(갈색)의 평균 Shapley 값이 음수로 나타났으며, 이는 wrong_fragment의 값이 0일 때 데이터가 Pod 또는 Teardrop이라는 예측 결과에 부정적인 영향을 미친다는 것을 의미한다. 반대로 wrong_fragment=1인 경우에는 Pod 값이 양수로 도출되어 데이터가 Pod라는 예측 결과에는 긍정적인 영향을 미친다고 볼 수 있다. 이와 같은 방법으로 다른 결과에 대해서도 해석이 가능하다. 해당 논문에서 제안된 프레임워크는 결과 예측 시 가장 영향도가 높았던 feature를 알 수 있어 IDS 환경에서의 공격 탐지 모델에 해석 가능성 및 신뢰성을 부여할 수 있다. 이를 통해 분석가가 IDS 판단을 더 잘 이해할수록 도움을 줄 수 있다.

Kim 외 3인[19]는 보안 분석가에게 AI 모델의 예측에 대한 신뢰성을 제공하기 위해 SHAP 기반 AI 통계분석 기법을 활용한 예측 신뢰성 지표를 제안했다. 제안된 방법은 XAI 기법 중 SHAP를 이용하여 대규모 위협을 효율적으로 분석하고 AI 모델 학습에 크게 영향을 끼친 feature를 분석가가 쉽게 이해할 수 있도록 FOS(Feature Outlier Score)라는 점수를 통해 계산된 지표를 제공한다. AI의 한계점으로 인해 분석가는 실 보안 환경의 위협 각각에 대해 직접적인 최종 확인이 필요했으나 일일 작업량이 제한되어 있어 모든 위협에 대한 분석이 불가능했다. 이를 해결하기 위해 신뢰 지표를 제안하였으며 분석가가 중요한 데이터에 집중하고 신속하게 AI 예측을 확인할 수 있도록 하였다. 결과적으로 AI의 해석가능성을 제공하면서 기존 AI 모델 대비 우수한 결과가 산출되는 것을 확인하였다. 그림 11은 IDS(Intrusion Detection System) dataset를 활용한 실험 결과를 나타낸다.

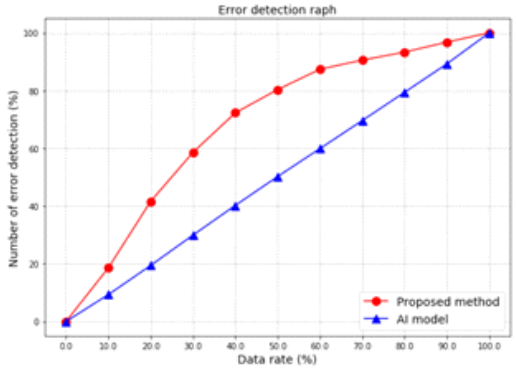


(a) *wrong_fragment*



(b) *dst_host_same_src_port_rate*

〔그림 10〕 feature 값과 공격의 특정 유형 간의 관계 예시〔18〕



〔그림 11〕 AI 모델 및 제안된 방법의 탐지된 오류 비율〔19〕

Ⅳ. XAI 평가 방법론

XAI가 설명을 적절하게 제공했는지에 대한 성능 평가는 중요한 영역임에도 XAI에 대한 전체 연구 중에5%에 불과하다. XAI 성능평가는 아직까지 완성된 방법론이 있지는 않지만, 연구가 진행중인 방법론에 대해 설명하고자 한다[20]. XAI를 평가 전 평가 기준을 분류하며, DL을 적용한 최근 4가지 보안 시스템에서 평가 방법론을 적용한 결과에 대해 설명한다.

4.1. Dataset

XAI 평가 방법을 검증하기 위해 DL을 적용한 최근 4가지 보안 시스템을 이용한다(표 1 참조).

첫 번째 시스템인 Drebin+ 시스템은 MLP를 사용하여 안드로이드 악성코드를 식별한다. Grosse 등[21]에 의해 제안되었으며, Arp[25]이 개발한 feature를 기반으로 한다. feature는 안드로이드 애플리케이션에서 정적으로 추출되며, 전체 129,013개 안드로이드 애플리케이션에서 75%를 학습에 사용하고 25%를 테스트에 사용한다. 두 번째 시스템인 Mimicus+ 시스템도 MLP를 사용하며, 악성 PDF 문서를 탐지한다. Guo 등[22]의 연구를 기반으로 다시 구현되었으며 Smutz 외 1인[26]이 도입한 feature를 기반으로 한다. PDF 문서에서 총 135개의 feature를 추출하여 진행되며 검증을 위해 5,000개의 정상 PDF 파일과 5,000개의 악성 PDF 파일이 포함된 원본 dataset를 사용하고, 이 중 75%를 학습에 25%를 테스트에 사용한다. 세 번째 시스템인 DAMD 시스템은 CNN을 사용하며 악성 안드로이드 애플리케이션을 식별한다. 시스템의 자세한 내용은 McLaughlin 등[23]의 논문에서 확인 가능하다. 시스템 검증을 위해 Malware Genome Project[27]의 데이터를 활용하였으며 이 dataset는 총 2,123개의 애플리케이션으로 구성되어 있고 869개의 정상 샘플과 1,260의 악성 샘플이 있다. 이 중 학습에 75%를 테스트에 25%를 사용하여 결과를 산출하였다. 네 번째 시스템인 VulDeepPecker 시스템은 RNN을 사용하며 소스 코드의 취약점을 발견한다[24]. LSTM cells[28]를 사용하였으며 word2vec embedding[29]를 적용하였다. 결과의 검증을 위해 취약점에 해당하는 10,444개의 가젯과 39,757개의 코드 가젯으로 구성된 CWE-119 dataset을 사용하였다. 이 중 학습에 80%를, 테스트에 20%를 사용하여 결과를 검증하였다.

〔표 1〕 4가지 보안 시스템 개요

System	Publication	Type	Layer
Drebin+	ESORICS'17 〔21〕	MLP	4
Mimicus+	CCS'18 〔22〕	MLP	4
DAMD	CODASPY'17 〔23〕	CNN	5
VulDeePecker	NDSS'18 〔24〕	RNN	5

4.2. 정확도 기반 XAI 평가

XAI를 평가하기 위한 첫 번째 방법인 정확도 평가 방법은 XAI가 예측과 관련된 feature를 얼마나 정확하게 추출하는지를 반영한다. feature와 예측 간의 관계를 직접 평가하는 것에는 한계가 있으므로, 간접적인 방법에 따라 가장 관련성이 높은 feature를 제거함으로써 NN의 예측이 어떻게 변하는지를 측정한다.

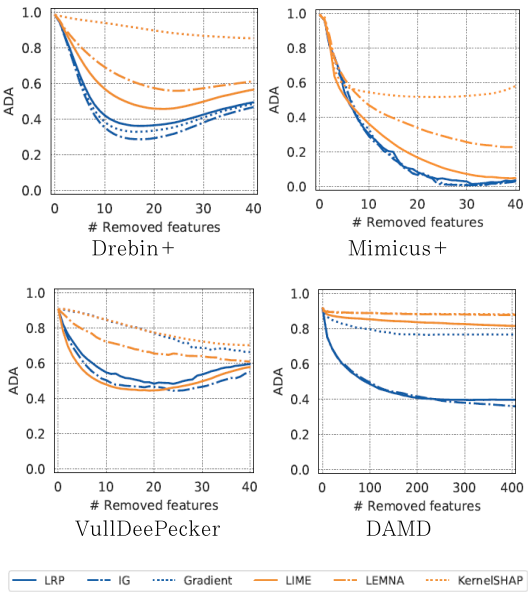
XAI의 정확도 평가를 위한 DA(Descriptive Accuracy)는 샘플 x가 주어지면 가장 관련성이 높은 k개의 feature $x_1,...,x_k$를 샘플에서 제거하고 결정 함수 f_N을 사용하여 새 예측을 계산한 후, k개의 feature 없이 원래 예측 클래스 c의 점수를 측정하여 계산한다. DA 식은 다음 식(4)와 같다.

$$DA_k(x,f_N) = f_N(x|x_1 = 0,...,x_k = 0)_c \quad (4)$$

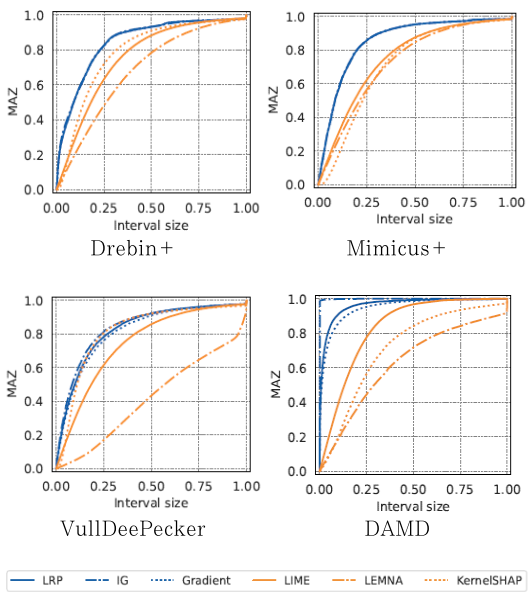
샘플에서 관련 feature를 제거하면 NN이 정확한 예측을 할 수 있는 정보가 적어지기 때문에 정확도가 낮아진다. XAI 성능이 높을수록 DA가 급격히 감소하며, XAI 성능이 비교적 낮을수록 DA가 점진적으로 감소한다.

그림 12는 4가지 보안 시스템에 대한 DA 실험 결과를 나타낸다. 전체적으로 IG 및 LRP 방법이 DA의 급격한 감소가 있음을 확인할 수 있다. VulDeepecker의 경우에서만 LIME의 방식이 미세한 차이로 가장 급격한 경사를 나타냄을 알 수 있다. 결과적으로 해당 실험 결과를 통해 LRP 및 IG 방법이 가장 우수함을 알 수 있다.

표 2는 그림 12의 DA 곡선에 대한 AUC(Area Under Curve)를 나타낸다. LRP와 IG는 DA 모든



〔그림 12〕 ADA(Average descriptive accuracy) 예시



〔그림 13〕 MAZ(Mass Around Zero) 예시

〔표 2〕 각 XAI 설명 방법론에 대한 Descriptive accuracy (DA)

Method	Drebin+	Mimicus+	DAMD	VulDeePecker
LIME	0.580	0.257	0.919	0.571
LEMNA	0.656	0.405	0.983	0.764
SHAP	0.891	0.565	0.966	0.869
Gradients	0.472	0.213	0.858	0.856
IG	0.446	0.206	0.499	0.574
LRP	0.474	0.213	0.504	0.625

dataset에서 최상의 방법으로 나타났으며, 다른 방법들에 평균적으로 최대 48% 우수하게 나타났다.

4.3. 희소성 기반 XAI 평가

예측에 영향을 미치는 feature에 높은 관련성을 할당하는 것은 우수한 설명 가능성을 제공하기 위해 필수 전제 조건이다. 그러나 분석가 인력은 제한적이기 때문에 feature의 분석 또한 제한적으로 처리할 수 있다. 희소성 평가는 이러한 경우를 위한 방법론으로 다음의 방법을 통해 측정된다. 관련성 값을 범위 [-1, 1]로 스케일링하고, 정규화된 histogram h를 계산하며, 다음 식(5)와 같이 정의된 MAZ(Mass Around Zero)를 계산하여 측정된다.

〔표 3〕 각 XAI 설명 방법론에 대한 희소성(MAZ)

Method	Drebin+	Mimicus+	DAMD	VulDeePecker
LIME	0.757	0.752	0.833	0.745
LEMNA	0.681	0.727	0.625	0.416
SHAP	0.783	0.716	0.713	0.813
Gradients	0.846	0.856	0.949	0.816
IG	0.847	0.858	0.999	0.839
LRP	0.846	0.856	0.964	0.827

$$MAZ(r) = \int_{-r}^{r} h(x)\,\mathrm{dx} \quad \text{for} \quad r \in [0.1] \tag{5}$$

희소한 설명을 제공하는 XAI는 대부분의 feature가 관련 집합이 없는 것으로 표시되기 때문에 MAZ가 0에서 가파르고 1 부근에서는 평평하며, 반대로 밀집된 설명을 제공하는 XAI는 0에서는 눈에 띄게 작은 기울기를 가지며 관련 feature의 집합이 더 많음을 나타낸다. 결과적으로 MAZ 분포가 0에서 가파를수록 XAI의 설명 방법이 우수함을 알 수 있다.

그림 13은 4가지 보안 시스템에 대한 희소성 실험 결과를 나타낸다. IG, LRP 및 Gradients 방법이 가장 가파른 기울기를 보여주고 대부분의 feature가 거의 관련성이 없다는 것을 알 수 있다. 대조적으로 다른 XAI 방법들은 더 넓은 범위의 관련성 값을 생성하고,

덜 희소하기 때문에 0에서 MAZ의 기울기가 비교적 가파르지 않은 것을 확인할 수 있다. DAMD의 경우, IG는 0에서 매우 가파른 기울기를 나타내고 있으며, 이는 거의 모든 feature가 관련성이 없는 것으로 볼 수 있다. DAMD dataset에는 최대 520,000개의 feature가 있는 샘플이 포함되어 있기 때문에 IG 방법을 사용할 경우 분석가는 어렵지 않게 분석이 가능하다.

표 3은 그림 13에 대한 AUC를 계산하여 MAZ의 성능을 요약한 표이다. 높은 AUC는 0에 가까운, 대부분의 관련성이 없는 feature가 더 많다는 것을 의미한다. 즉 설명이 더 희소하다는 것을 나타낸다. 그래프와 표를 통해 IG 방법이 다른 XAI 방법들 중 가장 우수함을 알 수 있다.

Ⅴ. 결 론

AI 기술은 많은 분야에서 널리 사용되고 있다. 보안 분야에서도 AI의 도입이 이루어지고 있으며, 높은 정확도와 자동화된 방식으로 인해 활용이 증가하고 있는 추세이다. 그러나 AI 기술은 복잡한 구조로 인해 'black-box' 특성을 가지고 있어 결과에 대한 해석이 불가하여 신뢰성이 낮다는 문제가 있다. 이에 AI에 해석가능성을 제공하기 위해 XAI 방법이 주목받고 있지만 아직 보안 분야에서의 XAI 기술의 적용 사례가 많이 출현하지 않은 상황이다. 본 논문에서는 XAI 개념 및 필요성, 정보보호 분야에서의 XAI 방법론 적용 사례, 평가 방법에 대해 소개하였다. 본 논문에서 소개한 XAI 방법론 및 결과를 통해 향후 보안 분야에서 AI 기술 도입 시 신뢰성에 도움을 줄 수 있을 것이라고 기대한다.

참 고 문 헌

[1] C. Beek. et al, "McAfee Labs Threat Report August 2019," *Mcfee Labs*. rep. 2019.

[2] 하연 편집부, *설명가능한 인공지능(XAI) 기술 동향과 데이터 산업의 시장 전망*. 하연. 2021.

[3] D. Gunning and D. Aha. "DARPA's explainable artificial intelligence (XAI) program." *AI Magazine*. 40(2). Jun. 2019.

[4] W. Samek, T. Wiegand, and KR. Müller. "Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models." *arXiv preprint arXiv:1708.08296*. Aug. 2017.

[5] A. Das, and P. Rad. "Opportunities and challenges in explainable artificial intelligence (xai): A survey." *arXiv preprint arXiv:2006.11371*. Jun. 2020.

[6] AB. Arrieta, et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI." *Information Fusion*. 58. pp. 82-115. Dec. 2019.

[7] J. Vaughan, et al. "Explainable neural networks based on additive index models." *arXiv preprint arXiv:1806.01933*. Jun. 2018.

[8] Z. Xi, and G. Panoutsos. "Interpretable Convolutional Neural Networks Using a Rule-Based Framework for Classification." *Intelligent Systems: Theory, Research and Innovation in Applications*. 864. 2020.

[9] A. Mills, T. Spyridopoulos and P. Legg. "Efficient and interpretable real-time malware detection using random-forest." *2019 International conference on cyber situational awareness, data analytics and assessment (Cyber SA)*. IEEE, pp. 1-8. Nov. 2019.

[10] S. Bach, et al. "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation." *PloS one*. 10(7). July. 2015.

[11] A. Kuppan, and NA. Le-Khac. "Black box attacks on explainable artificial intelligence (XAI) methods in cyber security." *2020 International Joint Conference on Neural Networks (IJCNN)*. IEEE, pp. 1-8. Sep. 2020.

[12] B. Zhou, et al. "Learning deep features for discriminative localization." *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 2921-2929. Dec. 2016.

[13] G. Iadarola, et al. "Evaluating deep learning classification reliability in android malware family detection." *2020 IEEE International Symposium on Software Reliability Engineering Workshops*

(ISSREW). IEEE, pp. 255-260. Oct. 2020.

[14] MT. Ribeiro, S. Singh, and C. Guestrin. ""Why should i trust you?" Explaining the predictions of any classifier." *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*. pp. 1135-1144. Feb. 2016.

[15] M. Kinkead, et al. "Towards Explainable CNNs for Android Malware Detection." *Procedia Computer Science*. 184. pp. 959-965. Mar. 2021.

[16] LS. Shapley. “17. A value for n-person games.” *Princeton University Press*, 2016.

[17] SM, Lundberg and S. Lee. "A unified approach to interpreting model predictions." *Proceedings of the 31st international conference on neural information processing systems*. pp.4768-4777. May. 2017.

[18] M. Wang, et al, ‘‘An explainable machine learning framework for intrusion detection systems,’’ *IEEE Access*, 8, pp. 73127–73141, Apr. 2020,

[19] H. Kim, et al. "Cost-Effective Valuable Data Detection Based on the Reliability of Artificial Intelligence." *IEEE Access*. pp. 108959-108974. Sept, 2021.

[20] A. Warnecke, et al. "Evaluating explanation methods for deep learning in security." *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, pp. 158-174. Sept. 2020.

[21] K. Grosse, et al. "Adversarial examples for malware detection." *European symposium on research in computer security*. Springer, Cham, pp. 62-79. Aug. 2017.

[22] W. Guo, et al. "Lemna: Explaining deep learning based security applications." *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. pp. 364-379. Oct. 2018.

[23] N. McLaughlin, et al. "Deep android malware detection." *Proceedings of the seventh ACM on conference on data and application security and privacy*. pp. 301-308. Mar. 2017.

[24] Z. Li, et al. "Vuldeepecker: A deep learning-based system for vulnerability detection." *arXiv preprint arXiv:1801.01681*. Jan. 2018.

[25] A. Daniel, et al. "Drebin: Efficient and explainable detection of android malware in your pocket”." *Proceedings of 21th Annual Network and Distributed System Security Symposium (NDSS)*. Feb. 2014.

[26] C. Smutz, and A. Stavrou. "Malicious PDF detection using metadata and structural features." *Proceedings of the 28th annual computer security applications conference*. pp. 239-248. Dec. 2012.

[27] Y. Zhou, and X. Jiang. "Dissecting android malware: Characterization and evolution." *2012 IEEE symposium on security and privacy*. IEEE, pp. 95-109. May. 2012.

[28] S. Hochreiter, and J. Schmidhuber. "Long short-term memory." *Neural computation*. 9(8). Nov. 1997.

[29] T. Mikolov, et al. "Efficient estimation of word representations in vector space." *arXiv preprint arXiv:1301.3781*. Sep. 2013

〈저자소개〉

김 홍 비 (Hongbi Kim)

정회원

2020년 2월 : 호서대학교 정보보호학과 졸업

2020년 3월~현재 : 호서대학교 정보보호학과 석사과정

<관심분야> 악성코드 분석, 정보보호, AI

이 태 진 (Taejin Lee)

종신회원

2003년 2월 : 포항공과대학교 컴퓨터공학과 졸업

2008년 2월 : 연세대학교 컴퓨터공학과 석사 졸업

2013년 1월~2017년 2월 : 한국 인터넷진흥원 팀장

2017년 2월 : 아주대학교 컴퓨터공학과 박사 졸업

2017년 3월~현재 : 호서대학교 정보보호학과 교수

<관심분야> 시스템 보안, 악성코드 분석, 침해사고 대응, AI