J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

AI 모델의 Robustness 향상을 위한 효율적인 Adversarial Attack 생성 방안 연구☆

A Study on Effective Adversarial Attack Creation for Robustness Improvement of AI Models

정 시 온[1] 한 태 현[1] 임 승 범[1] 이 태 진[1*]
Si-on Jeong Tae-hyun Han Seung-bum Lim Tae-jin Lee

요 약

오늘날 AI(Artificial Intelligence) 기술은 보안 분야를 비롯하여 다양한 분야에 도입됨에 따라 기술의 발전이 가속화되고 있다. 하지만 AI 기술의 발전과 더불어 악성 행위 탐지를 교묘하게 우회하는 공격 기법들도 함께 발전되고 있다. 이러한 공격 기법 중 AI 모델의 분류 과정에서 입력값의 미세한 조정을 통해 오 분류와 신뢰도 하락을 유도하는 Adversarial attack이 등장하였다. 앞으로 등장할 공격들은 공격자가 새로이 공격을 생성하는 것이 아닌, Adversarial attack처럼 기존에 생성된 공격에 약간의 변형을 주어 AI 모델의 탐지체계를 회피하는 방식이다. 이러한 악성코드의 변종에도 대응이 가능한 견고한 모델을 만들어야 한다. 본 논문에서는 AI 모델의 Robustness 향상을 위한 효율적인 Adversarial attack 생성 기법으로 2가지 기법을 제안한다. 제안하는 기법은 XAI 기법을 활용한 XAI based attack 기법과 모델의 결정 경계 탐색을 통한 Reference based attack이다. 이후 성능 검증을 위해 악성코드 데이터 셋을 통해 분류 모델을 구축하여 기존의 Adversarial attack 중 하나인 PGD attack과의 성능 비교를 하였다. 생성 속도 측면에서 기존 20분이 소요되는 PGD attack에 비하여 XAI based attack과 Reference based attack이 각각 0.35초, 0.47초 소요되어 매우 빠른 속도를 보이며, 특히 Reference based attack의 경우 생성률이 97.7%로 기존 PGD attack의 생성률인 75.5%에 비해 높은 성공률을 보이는 것을 확인하였다. 따라서 제안한 기법을 통해 더욱 효율적인 Adversarial attack이 가능하며, 이후 견고한 AI 모델을 구축하기 위한 연구에 기여할 수 있을 것으로 기대한다.

☞ 주제어 : 인공지능, 견고성, 적대적 공격

ABSTRACT

Today, as AI (Artificial Intelligence) technology is introduced in various fields, including security, the development of technology is accelerating. However, with the development of AI technology, attack techniques that cleverly bypass malicious behavior detection are also developing. In the classification process of AI models, an Adversarial attack has emerged that induces misclassification and a decrease in reliability through fine adjustment of input values. The attacks that will appear in the future are not new attacks created by an attacker but rather a method of avoiding the detection system by slightly modifying existing attacks, such as Adversarial attacks. Developing a robust model that can respond to these malware variants is necessary. In this paper, we propose two methods of generating Adversarial attacks as efficient Adversarial attack generation techniques for improving Robustness in AI models. The proposed technique is the XAI-based attack technique using the XAI technique and the Reference based attack through the model's decision boundary search. After that, a classification model was constructed through a malicious code dataset to compare performance with the PGD attack, one of the existing Adversarial attacks. In terms of generation speed, XAI-based attack, and reference-based attack take 0.35 seconds and 0.47 seconds, respectively, compared to the existing PGD attack, which takes 20 minutes, showing a very high speed, especially in the case of reference-based attack, 97.7%, which is higher than the existing PGD attack's generation rate of 75.5%. Therefore, the proposed technique enables more efficient Adversarial attacks and is expected to contribute to research to build a robust AI model in the future.

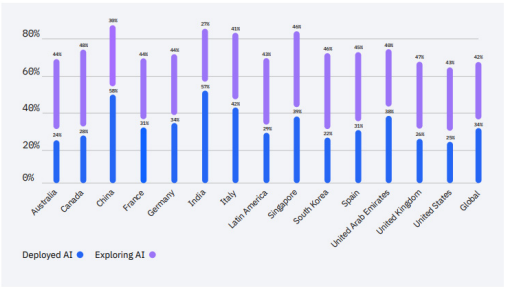
☞ keyword : Artificial Intelligence, Robustness, Adversarial attack

1 Department of Information Security, Hoseo University., Chungnam, 31499, Korea.
* Corresponding author (kinjecs0@gmail.com)
[Received 07 March 2023, Reviewed 01 April 2023(R2 21 June 2023), Accepted 03 July 2023]
☆ 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원(No.2022-0-00089, 사이버공격 대응을 위한 Life-cycle 기반 공격그룹 식별 및 유형 분석 기술 개발)과 대한민국 정부 산업통상자원부 및 방위사업청 재원으로 민군협력진흥원에서 수행하는 민군기술협력사업의 연구비 지원으로 수행되었습니다. 협약번호 (21-CM-EC-07).

1. 서 론

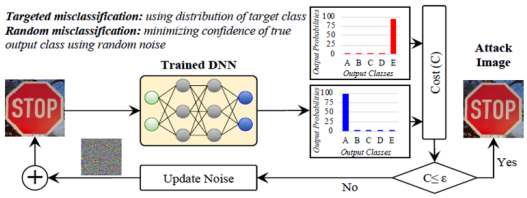
최근 보안 분야를 비롯하여 다양한 분야에 AI (Artificial Intelligence) 도입이 확산되면서 다양한 연구 및 기업에서 AI가 활용되고 있다. IBM이 2022년 5월경 발표한 '2022년 AI 도입 지수' 보고서에 따르면 그림 1과 같이 전 세계 7,500곳 이상의 기업 중 35%의 기업이 이미 AI를 도입하여 사용 중인 것으로 확인되었다[1]. 해당 보고서에 따르면 AI 도입률은 전년 대비 13%가 증가하였으며, 매년 꾸준한 추세로 증가하고 있다. AI 기술의 발전 및 도입의 확산에 따라 AI 알고리즘을 우회하는 다양한 공격 기법들이 발전되고 있다. AI를 활용하여 기존 악성 행위 탐지를 교묘하게 우회하는 지능화된 보안 위협이 발생되고 있다[2]. 이에 따라 AI의 안전을 확보하는 일에 대한 중요도와 관심이 상승하고 있다.

2014년 Goodfellow 등[3]에 의해 AI 모델의 오 분류와 신뢰도 하락을 유도하는 Adversarial attack이 소개되었다. Adversarial attack은 Adversarial attack 기법으로 생성된 Adversarial example을 통해 AI의 분류 체계를 무너뜨려 AI가 잘못된 결과를 산출하도록 조작하는 공격 기법을 말한다. 이는 머신러닝 알고리즘 내에 내재하고 있는 취약점에 의해 적대적 환경에서 발생할 수 있는 위협 기법이다. 학습 과정이 아닌 분류 단계에서 입력값의 미세한 조정을 통해 오 분류를 유도하며, 육안으로 공격 여부를 구별할 수 없다[4]. DNN 기반의 딥러닝 모델은 훈련에서의 불안정성 및 비 신뢰적인 과정으로 인해 Adversarial attack에 매우 취약하다는 단점이 존재한다. DNN은 입력의 작고 육안으로 쉽게 구별하기 힘든 미세한 변화(Perturbation)를 오 분류 할 수 있다[5][6]. Goodfellow 등[3]은 DNN 기반의 딥러닝 모델이 Adversarial example로 인해 높은 확률로 오 분류를 산출할 수 있다는 것을 증명하였다. 특히나 보안 분야에서 사용 중인 Malware Detection System(MDS) 및 Intrusion Detection System(IDS)의 경우 DNN 기반의 딥러닝 모델이므로 Adversarial attack에 매우 취약할 수밖에 없다. 또한 앞으로 등장할 공격들은 새로운 공격을 생성하는 것이 아닌 Adversarial attack처럼 이미 생성하였지만 분류 모델에 의해 탐지되었던 악성코드에 대해 조금의 변형을 주어 새로이 탐지 체계를 회피하는 방법으로 공격을 생성해나가는 방식이다[7]. 이러한 악성코드의 변종에도 대응이 가능한 견고한 모델을 만들어야 하는 것이 앞으로의 AI 보안의 목표이다. 하지만 일반적인 이미지 기반의 분류 모델에 비해 악성코드 분류 모델을 대상으로 하는 Adversarial attack에 관한 연구가 많지 않다. 따라서 악성코드 분류 모델을 대상으로 하는 Adversarial attack에 관한 연구가 필요하다. 이미지 분류 모델의 경우, 기존 Adversarial attack 기법을 통해 Adversarial example을 생성 시 높은 성공률을 나타낸다는 연구 결과는 많이 존재한다. 하지만, 악성코드 데이터 셋을 사용하여 기존의 Adversarial attack 기법으로 Adversarial example을 생성하는 연구는 거의 없다. 본 논문에서는 악성코드 데이터 셋을 사용하여 기존 Adversarial attack을 통해 Adversarial example을 생성한 후 생성률과 생성 시간, 원본과의 거리를 확인한다. 이후 XAI(eXplainable Artificial Intelligence) 기법 중 SHAP value 산출을 활용하여 Adversarial example을 생성하는 기법과 기존 Anomaly 데이터를 사용하는 Reference based explanation을 분류 모델에 적용시켜 Adversarial example을 생성하는 기법을 제안한다. 또한 기존의 Adversarial attack 기법과 제안하는 2가지 기법 간의 비교를 통해 악성코드 데이터 셋으로 생성 가능한 가장 효율적인 Adversarial example 생성 기법을 제안하고자 한다.



(그림 1) 전 세계 AI 도입률[1]
(Figure 1) AI adoption rates around the world[1]

본 논문의 구성은 다음과 같다. 2장에서는 기존 Adversarial attack을 통한 Adversarial example 생성 기법을 소개하며, 3장에서는 제안하는 XAI based attack 및 Reference based attack에 대한 동작 방식과 더불어 2가지 기법의 Adversarial example 생성 방법을 소개한다. 4장에서는 실험에서 사용한 데이터 셋에 대한 설명과 데이터 셋을 통해 학습한 AI 모델의 성능을 제시하며, 제안한 2가지 기법과 기존 Adversarial attack 기법을 통해 결과를 산출한 후 산출된 결과를 비교한다. 이후 5장에서 결론으로 마친다.

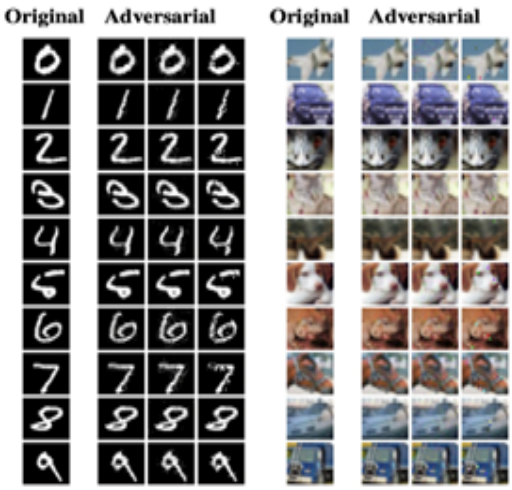


(그림 2) 적대적 공격[4]
(Figure 2) Adversarial attack[4]

2. 관련 연구

Adversarial attack이란 그림 2와 같이 머신러닝 알고리즘 내에 내재하고 있는 취약점에 의해 적대적 환경에서 발생할 수 있는 위협 기법으로, 원본 데이터에 육안으로 쉽게 구별하기 힘든 아주 작은 변화(Perturbation)를 주어 오 분류와 신뢰도 하락을 유발하는 공격 기법이다. 학습 과정이 아닌 분류 단계에서 입력값의 미세한 조정을 통해 오 분류를 유도하며, 육안으로 공격 여부를 구별할 수 없다[4]. Adversarial attack은 입력 데이터에 최소한의 Perturbation을 가해 AI 모델의 결정 경계는 넘어가면서 실제 라벨은 변경되지 않고 AI 모델의 판단만을 혼동시켜 공격을 공격이 아닌 정상으로 판단하도록 유도한다. 이에 따라 AI 모델의 결정 경계를 넘어가는 원본 데이터와 가장 가까운 데이터를 찾음으로써 Adversarial example을 생성하고, 공격을 진행하게 되는 것이다. Adversarial attack을 통해 생성된 Adversarial example은 그림 3과 같다.

Adversarial attack을 통해 Adversarial example을 생성하는 기법은 다양하다. FGSM(Fast Gradient Sign Method)[3]은 모델 내부의 가중치 값을 이용하여 Input data의 손실함수 기울기를 계산하고, 이를 바탕으로 입력값의 조작을 통해 Adversarial example을 생성하는 공격 기법이다. 이는 미분 값을 구하면 바로 실행되는 One-step attack 알고리즘이므로 생성 속도가 매우 빠르다는 장점이 있다. DeepFool[8]은 더욱 효율적이고 더욱 작은 변형으로 비선형 신경망 구조에 여러 번의 질의를 통해 무표적 공격을 수행한다. 기울기 계산이 아닌 여러 개의 점에서 Decision boundary에 대해 수직으로 투영하고, 그에 따라 적당한 변형 값을 추가하여 Adversarial example을 생성하게 된다. JSMA(Jacobian-based Saliency Map Attack)[9]는 입력값이 출력값에 미치는 변화를 행렬로 대응시켜 기울기 변화를 통해 모델이 오 분류 하도록 하는 Adversarial example을 생성하는 방식이다. Zoo(Zerosh Order Optinuzation



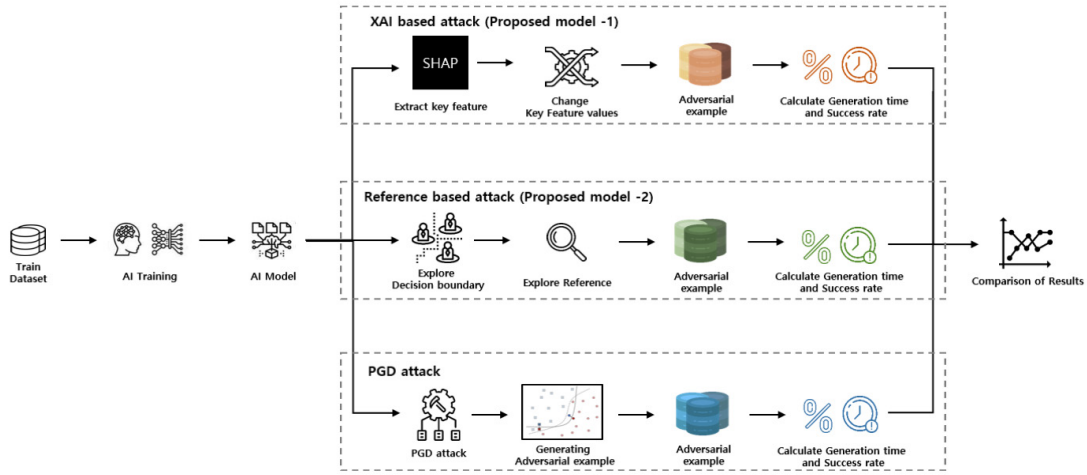
(그림 3) Adversarial example 생성 예시[11]
(Figure 3) Example of Adversarial example created[11]

based Black-Box) attack[10]은 기울기를 직접 가져와서 사용하는 기울기 기반 방식을 사용하지는 않지만 기울기 기반 방식과 유사하게 기울기를 추정하여 최적화 문제를 해결하며 Adversarial example을 생성한다. C&W(Carlini& Wagner) attack[11]은 Approximate 방법으로 원본과 Adversarial example 간의 왜곡과 공격 성공률의 가중치를 적절하게 찾도록 설계하여 최소한의 변형(Perturbation)을 담당하는 손실 함수와 공격 성공률을 높이는 손실 함수의 합을 최소화하여 최적의 Adversarial example을 생성한다. PGD(Projected Gradient Descent) attack[12]은 FGSM에서 발전한 공격 기법으로 지정한 수많은 Step을 반복하여 L-infinity norm 내에서 Adversarial example을 생성한다. 가장 일반적으로 잘 알려진 공격 기법으로, 많은 Adversarial attack에 대한 Defence 연구에서 사용되고 있다.

본 연구에서는 Adversarial attack을 통한 Adversarial example 생성을 위해 앞서 소개한 Adversarial attack 기법 중 가장 보편적으로 사용되며 강력한 White-box Adversarial attack인 PGD attack을 사용하였다. PGD attack에서 Adversarial example을 생성하기 위한 수식은 수식 1과 같다.

$$x^{t+1} = \Pi_{x+S}(x^t + \alpha sgn(\nabla_x L(\theta, x, y))) \quad (1)$$

먼저 Loss 값에 대하여 경사 값을 계산하고, 경사 값의



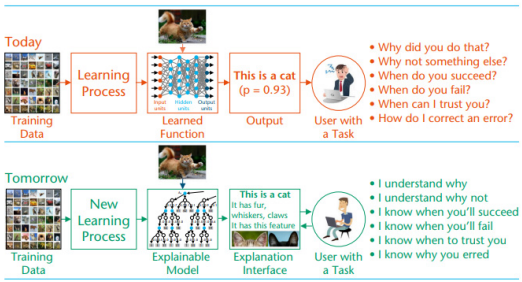
(그림 4) 제안 모델
(Figure 4) Proposed Model

Sign 값에 대해 α만큼 변경하여 원본 데이터(x^t)에 더해준다. S는 Learning rate를 의미하며 Learning rate를 지정하여 각 Step마다 Learning rate만큼 입력 데이터의 변형이 일어나도록 유도한다. x^{t+1}은 생성된 Adversarial example을 의미한다. 이러한 방법으로 악성코드 데이터 셋을 이용하여 PGD attack을 통해 원본과 가까운 Adversarial example을 생성한다. 하지만 PGD attack의 경우 많은 연산량을 요구하므로 많은 시간이 소요된다는 단점이 존재한다.

3. 제안 모델

이 장에서는 DNN 기반의 분류 모델에 대한 효율적인 Adversarial attack을 위해 2가지 기법을 제안한다. 제안하는 기법은 XAI 기법을 활용하여 기여율이 높은 Feature의 값을 변경하는 XAI based attack과 모델의 결정 경계 탐색을 통해 값을 변경하는 Reference based attack이다. 제안한 기법으로 생성된 Adversarial example의 성능 비교를 위해 기존 Adversarial attack 중 PGD attack을 사용한다. PGD attack으로 생성된 Adversarial example과 제안한 기법으로 생성된 Adversarial example의 성능 비교를 통해 제안한 2가지 기법을 통한 효율적인 Adversarial example 생성이 가능한지 확인하고자 한다.

제안하는 모델은 그림 4와 같으며, 해당 모델의 동작 방식은 다음과 같다. 기존에 수집한 학습 데이터 셋을 기반으로 AI 모델의 학습 진행을 통해 AI 모델을 구축한다. 이후 학습 데이터 셋을 대상으로 각 기법을 통해 Adversarial attack을 진행한다. 먼저, XAI based attack은 SHAP을 통해 각 Feature 별 기여도를 산출한 후 AI 판단에 가장 많은 기여를 한 상위 3개의 Feature에 대해 값 변경을 진행하여 Adversarial example을 생성한다. Reference based attack은 모델의 의사결정 경계를 찾아 원본과 가장 가까운 정상 범주의 데이터, 즉 Reference를 탐색하여 탐색한 Reference를 바탕으로 Adversarial example을 생성한다. 또한 기존 Adversarial attack과의 성능 비교를 위해 기존의 Adversarial attack 중 PGD attack을 통해 Adversarial example을 생성한다. 기존 Adversarial attack 기법인 PGD attack의 경우 이미지 데이터 셋의 입력을 기반으로 개발되었다. 따라서 PGD attack과 제안하는 XAI based attack은 CNN(Convolution Neural Network) 모델을 통해 AI를 구축하여 사용하였다. 또한 기존 악성코드 분류 모델과 같은 모델에서의 Adversarial example 생성의 성능을 확인하고자 Reference based attack은 악성코드 분류 모델과 동일한 DNN(Deep Neural Networks) 모델을 통해 AI를 구축하여 사용하였다. 이후 제안하는 2가지 기법과 기존 Adversarial attack 기법으로 생성된 총 세 종류의 Adversarial example에 대하여 성공률, 생성 시간을 산출하여 성능 비교를 진행한다.



(그림 5) XAI 동작 원리[13]
(Figure 5) Principle of XAI operation[13]

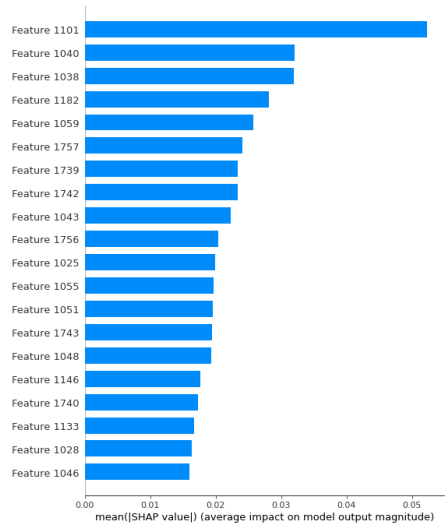
3.1 XAI 기반 공격

XAI(eXplainable Artificial Intelligence)[13]는 그림 5와 같이 기존 Black-box 모델의 특성상 판단 기준을 알 수 없었던 AI의 설명성을 제공하는 방법론으로, AI의 예측에 대해 사람이 이해할 수 있는 형태로 설명하고 제시할 수 있다. XAI는 AI 모델의 예측에 큰 영향을 준 특징값에 대한 해석을 제공한다. 대표적인 XAI 기법으로는 1개의 Input data를 확인할 수 있는 로컬 관점의 설명 제공이 가능한 LIME[14], 로컬 관점의 설명과 더불어 글로벌 관점의 설명을 함께 제공할 수 있는 SHAP(SHapley Additive exPlanation)[15] 등이 있다.

SHAP(SHapley Additive exPlanation)[15]은 로컬 설명을 기반으로 하며, 로컬 설명과 더불어 데이터의 전체적인 영역에 대한 해석, 즉 글로벌 해석을 제공하는 대표적인 XAI 기법 중 하나이다. SHAP은 Perturbation 기반의 Black-box 모델 설명 기법으로 Shapley value를 통해 AI 판단에서 각 특징값의 영향도에 대한 확인이 가능하다. 각 특징값의 기여도를 계산하여 관측치의 예측값을 설명하는 것이 목적이다. SHAP은 모든 가능한 특징값 순서쌍에 대하여 샘플링하고, 평균값을 계산하여 산출한다. SHAP value 산출을 통해 AI 판단에 가장 많은 기여를 하는 특징값에 대한 판단이 가능하다. SHAP은 수식 2와 같이 동작한다.

$$\Phi_i = \sum_{S \subseteq F \setminus i} \frac{|S|!(|F|-|S|-1)!}{|F|!}[f_{S\cup i}(x_{S\cup i}) - f_s(x_s)] \quad (2)$$

집합 F는 기여도를 구하고자 하는 i 번째 특징값을 포함한 전체 집합을 의미하며, 집합 S 는 전체 집합에서 i 번째 특징값이 빠진 모든 부분 집합을 의미한다. $f_{S\cup i}(x_{S\cup i})$ 는



(그림 6) SHAP Values 산출
(Figure 6) SHAP Values Calculation

i 번째 특징값을 포함한 전체 기여도를 나타내며, $f_s(x_s)$ 는 i 번째 특징값이 빠진 나머지 부분 집합의 기여도를 나타낸다. 이러한 수식을 통해 계산된 i 번째 데이터(i 번째 특징값)에 대한 Shapley Value 값(Φ_i), 즉 해당 특징값의 기여도를 구할 수 있다. SHAP은 모델의 특징에 따라 LIME과 Shapley value를 활용한다. 모든 모델에 적용이 가능한 Kernel SHAP, 의사결정 트리 기반 모델에 대한 SHAP 값을 계산하는 Tree SHAP, 딥러닝 기반 모델에 대한 SHAP 값을 계산하는 Deep SHAP 등이 존재하며, 본 논문에서는 딥러닝 기반의 모델을 사용하였으므로 Deep SHAP을 사용하여 실험을 진행하였다.

본 실험에서는 원본 데이터에서 분류 모델의 판단에 가장 많은 기여를 하는 Feature의 산출을 위해 XAI 기법 중 AI 판단에 가장 많은 기여를 하는 특징값의 판단과 각 Feature 별 기여율 확인이 가능한 SHAP을 사용한다. 분류 모델로는 딥러닝 기반의 CNN 모델을 구축하여 연구를 진행하였으므로, SHAP 값 산출에는 Deep SHAP을 사용하였다. 본 제안 모델의 동작 방식은 다음과 같다. 먼저 그림 6과 같이 SHAP value를 산출한다. 그림 6에서 y축은 Feature를 나타내며, x축은 AI 판단에 대한 Feature 별 기여도(SHAP value)를 나타낸다. 이후 각 열에서 모델 판단에 기여도가 높은 순으로 Feature를 정렬하였다. 학습 데이터의 라벨이 0인 정상 데이터에 대하여 Feature 별 가장 작은 값을 산출한다. 앞서 추출한 정렬된 항목을

기준으로 기여율이 높은 상위 3개의 Feature를 선별하였고, 선별한 Feature의 값을 앞서 산출하였던 정상 데이터에서의 가장 작은 값으로 변경하였다. 기여율이 높은 상위 Feature의 값을 변경한 데이터의 라벨이 변경되었는지 확인하고, 변경되었으면 이를 Adversarial example로 사용하게 된다. SHAP을 사용한 XAI based attack 기법이므로 Adversarial example 생성 과정에서 바뀐 Feature에 대한 해석 제공이 가능하다.

Algorithm 1-XAI based attack

```
SET Train, Test: Image Input Data
SET Model: CNN
SET Target: Image Input Data

import shap                #SHAP Extraction
shap.initjs()
    bachground=Train[np.random.choice(Train.shape[0],
1000, replace=False)]
explainer=shap.DeepExplainer(Model, background)
shap_values=explainer.shap_values(Target)

def get_shap_values(n):
    df=pd.DataFrame()
    for i in df.columns:
        {temp_df=
df[i].abs().sort_values(ascending=False)
temp_df=pd.DataFrame(temp_df).reset_index(drop=Fals
e).rename(columns={'index':'index_'+str(i)})}
        train_df=train_df[train_df['Label']==int(0)].index
  #Extract min value by feature
train_df=train_df.drop
          (columns=['FileName', 'Label'])
for i in train_df.columns:
    min_li.append(train_df[i].min())

setting=len(Target)        #XAI based attack
distance_df=pd.DataFrame()
for i in range(1):
    perturb_df=pd.DataFrame()
    for j in range(setting):
      for k in range(3):
        if Target_label[j].reshape(2,1)[0]==[1.]:
          importance_df=get_shap_values(j)
          for m in range
              {(in(len(importance_df.columns)/2))}:
            for n in range(feature_count):
              instance_index=importance_df[n]
              ori_value=new_df
                        [instance_index*42+i]
              perturb_value=normal_info_df
                    [instance_index*42+i]['min']
              cum_value +=
        normal_info_df[instance_index*42+i]['min']
              new+df['instance_index*42+i]=
                                  perturb_value
```

3.2 Reference 기반 공격

DNN은 높은 성능을 입증하였지만, 투명성과 해석성이 부족하여 보안 분야에서 사용되기 힘들다는 단점이 존재한다. DeepAID[16]의 논문에서는 Anomaly Detection의 결과 해석을 위해 Reference를 사용한다. 해당 논문에서 Reference란 Normal data 범주 내의 경계에 위치한 입력 데이터(Anomaly)로부터 가장 가까운 Normal data로 정의한다. 정확한 Reference를 찾기 위해 초기 Anomaly 값에 변화를 준 후 변화한 Anomaly 값으로부터 Loss 값을 산출하여 Loss 값이 감소하는 방향으로 Anomaly를 업데이트한다. 이후 Anomlay value와 Reference를 통해 Anomaly Detection의 결과 해석이 가능하다. 해당 논문의 목표는 Anomaly로부터 가장 가까운 Normal data를 찾는 것이다. Reference를 구하는 식은 수식 3과 같다.

$$argmin_{x^*} ReLU(\epsilon_R(x^*, f_R(x^*)) - (t_R - \varepsilon)) + \lambda \| x^* - x^\circ \|_2 \quad (3)$$

수식 3에서는 두 가지의 Loss 값을 사용한다. 하나는 Autoencoder에 의한 손실 함수이고, 다른 하나는 유클리디안 거리(L2-norm)이다. 두 Loss 값의 합이 최소가 되는 x^*를 찾는 것이 목적이다.

$$ReLU(\epsilon_R(x^*, f_R(x^*)) - (t_R - \varepsilon)) \quad (4)$$

수식 4는 x^*가 Normal data인지 판단하는 식이다. MSE Loss(ϵ_R)값이 Threshold(t_R)보다 작다면 Normal data이며 음의 값이 나오지 않도록 하였다. 따라서 손실 함수 값이 작아지는 방향으로 x^*가 업데이트되며, 이후 Normal data의 값을 찾게 된다. 이후 유클리디안 거리 측정 방식을 통해 계산한 원본 데이터와 Reference 사이의 거리 차이를 더한다. 해당 값의 감소는 원본 데이터와 가장 가까운 Normal data를 찾겠다는 의미이다. 더 이상 Loss 값이 감소하지 않을 때까지 Reference(x^*)의 업데이트를 반복한다. 이러한 방식으로 산출된 Reference는 Anomaly로부터 가장 가까운 Normal data 값이 되며, 이때 Reference와 Anomaly의 차이를 통해 Anomaly Detection의 결과를 신뢰할 수 있다.

본 연구에서는 해당 논문의 방식을 변형하여 악성코드 데이터 셋을 사용하여 분류 모델에서 Reference를 탐색하고, 이를 Adversarial example로 사용하고자 한다. 본 제안 모델의 동작 방식은 다음과 같다. Adversarial example을 생성하기 위해 분류 모델로는 DNN 모델을 구

축하여 학습을 진행하였으며, Reference 산출을 위해 Interpreter를 사용하였다. Interpreter에 입력된 원본 데이터는 업데이트되고 DNN 모델에 입력되기를 반복하여 최종적으로 Reference를 생성하고, 이를 Adversarial example로 사용한다. 분류 모델에서 Reference를 생성하는 식은 수식 5와 같다.

$$Loss1 = ReLU(final\,Layer - \varepsilon)$$
$$Loss2 = \lambda \| x^* - x^\circ \|_2 \quad (5)$$
$$Loss = Loss1 + Loss2$$

Reference 생성을 위해 두 개의 Loss 값을 사용하게 된다. Loss1은 DNN 모델의 Sigmoid 전 출력값이며, Loss2는 원본과 업데이트된 데이터 간의 유클리디안 거리이다. 원본 데이터에 대한 Adversarial example을 생성하기 위하여 원본 데이터의 Clone을 만들어 업데이트한다. 업데이트된 Clone은 다시 DNN 모델에 입력되어 변화한 예측값을 측정하며 동시에 Clone과 원본 데이터 사이의 거리를 측정한다. 이후 Loss1과 Loss2를 더한 최종 Loss값을 계산하여 Clone의 업데이트에 사용한다. Clone은 역전파 기법을 사용하여 최종 Loss가 감소하는 방향으로 업데이트되며, 이때 Clone의 일부 Feature만을 변형한다. 적절한 Learning rate를 설정하여 최종 Loss값을 조금씩 감소시켜 나가며 분류 모델이 Label을 잘못 판단하는 순간까지 반복한다. 마지막 Clone 값이 Reference가 되며, 이를 Adversarial example로 사용하게 된다. 표 1과 같이 Interpreter를 통해 AI 판단에 가장 많은 기여를 한 Feature 기반으로 값이 변경됨을 확인할 수 있으며, AI 판단에 많은 기여를 한 Feature 목록에 대한 제공 또한 가능하다.

(표 1) Reference 산출 결과
(Table 1) Reference Result

Feature Description	Value in Anomaly	comp.	Value in Reference
SizeOfInitializedData	0.2	>	0.044
Subsystem	0.3	>	0.149
AddressOfEntryPoint	0.2	>	0.069
SizeOfStackReserve	0.3	>	0.181
...	...	...	...

4. 실험 결과

이 장에서는 악성코드 데이터를 통해 제안 모델의 검증을 진행한다. 먼저 AI 모델을 구축한 후 제안한 기법인 XAI based attack과 Reference based attack을 통해 Adversarial example을 생성하였으며, 성능 검증을 위해 기존의 Adversarial attack 기법 중 가장 대중적으로 사용되는 PGD attack을 통해 Adversarial example을 생성하였다. 이후 2가지 비교 항목을 설정하여 제안하는 2가지 방법과 PGD attack 간의 비교를 통해 성능을 점검하였다.

4.1 데이터 셋

본 논문에서는 2019 KISA Datachallenge 악성코드 데이터 셋을 사용하였다. 데이터 셋의 구성은 표 2와 같다. 학습 데이터 셋은 악성 데이터 17,562개와 정상 데이터 11,568개로 총 29,130개를 사용하였으며, 테스트 데이터 셋은 악성 데이터 4,513개, 정상 데이터 4,518개로 총 9,031개를 사용하였다. 본 연구에서는 2019 KISA Data challenge Dataset 중 확인된 대표 AVClass 구성 확인을 통해 약 800가지 중 가장 많이 검출된 상위 5개(autoit, ramnit, scar, winactivator, zegost) AVClass의 학습 데이터를 사용하여 Adversarial example을 생성하였다. AVClass의 구성은 표 3과 같다. 학습 데이터 셋은 autoit 517개, ramnit 369개, scar 281개, winactivator 288개, zegost 198개로 총 1,653개를 사용하였다.

(표 2) 데이터 셋 구성
(Table 2) Dataset Configuration

Dataset	Malware	Normal	Total
Train	17,562	11,568	29,130
Test	4,513	4,518	9,031

(표 3) AVClass 구성
(Table 3) AVClass Configuration

AVClass	Train
autoit	517
ramnit	369
scar	281
winactivator	288
zegost	198
Total	1,653

PE(Portable Executable) Structure에는 실행파일을 실행하기 위한 다양한 정보가 기록되어 있다. 따라서 PE Structure의 정적 분석을 기반하여 Feature를 추출하였다.

PE Header에서 37개의 Feature를 추출하였고, DLL을 통해 512개, API를 통해 512개, PE section의 Entropy를 통해 128개의 Feature를 추출하였다. 또한 ASCII 문자열 분석을 통한 Feature hashing 기반 문자열 핵심 Feature 추출 방법을 통해 String 525개, Entry point 분석을 통한 핵심 Feature 추출 방법을 통해 Entry point 50개의 Feature를 추출하였다. 총 1,764개의 Feature를 추출하였으며, 이후 기존 Adversarial attack인 PGD attack과 XAI based attack에서는 CNN(Convolution Neural Networks) 모델을 사용하며, Reference based attack에서는 DNN(Deep Neural Networks) 모델을 사용하여 학습을 진행하게 된다.

4.2 AI 성능 평가

기존의 Adversarial attack 기법은 대부분 Feature의 범위가 일정한 이미지 데이터 셋의 입력을 기반으로 개발되었다. 하지만 악성코드 Feature의 경우 범위가 일정하지 않으므로 Feature의 정규화가 필요하다. 따라서 본 논문에서는 Min-max nomalisation 기법을 통해 악성코드 데이터 셋의 Feature 값을 0과 1 사이로 변환하여 사용하였다. 또한 기존 Adversarial attack 기법이 이미지 데이터 셋의 입력을 기반으로 개발되었으므로 생성률을 높이기 위해 이미지 변환이 필요하다. 따라서 Adversarial attack과 XAI based attack에서는 그림 7과 같이 42*42 크기의 이미지 변환 이후 이미지 데이터 처리에 성능이 우수한 CNN(Convolution Neural Networks) 모델을 통해 AI 모델을 구축하였다. 학습에 사용한 CNN Model의 구조는 표 4와 같으며, batch_size = 128, epochs = 187을 학습 파라미터로 지정하였다. CNN 모델의 학습 결과는 표 5와 같다. Reference based attack에서는 기존 악성코드 분류 모델과 동일한 DNN(Deep Neural Networks) 모델을 통해 AI 모델을 구축하였다. 데이터 셋은 앞서 이미지 변환 이전에 Min-max nomalisation 기법을 통해 악성코드 데이터 셋의 Feature 값을 0과 1 사이로 변환하여 사용하였다. 학습에 사용한 DNN Model의 구조는 표 6과 같으며, epochs = 10을 학습 파라미터로 지정하였다. DNN 모델의 학습 결과는 표 7과 같다.

(그림 7) 악성코드 이미지화
(Figure 7) Malware image conversion

(표 4) CNN Model 구성
(Table 4) CNN Model Configuration

Layer	Output shape	Param #
input	(None, 42, 42, 1)	0
conv2d_1	(None, 38, 38, 6)	156
batch_normalization	(None, 38, 38, 6)	24
max_pooling2d_1	(None, 19, 19, 6)	0
conv2d_2	(None, 15, 15, 16)	2416
max_pooling2d_2	(None, 7, 7, 16)	0
flatten	(None, 784)	0
dense_1	(None, 120)	94200
dense_2	(None, 84)	10164

(표 5) CNN Model 학습 결과
(Table 5) Results of CNN Model learning

	Accuracy	Recall	Precision	F1 score
CNN	0.9767	0.9770	0.9767	0.9767

(표 6) DNN Model 구성
(Table 6) DNN Model Configuration

Layer	Param #
Linear_1	2,335,095
Linear_2	1,167,768
Linear_3	389,403
Linear_4	56,576
Linear_5	8,256
Linear_6	2,080
Linear_7	264
Linear_8	18

(표 7) DNN Model 학습 결과
(Table 7) Results of DNN Model learning

	Accuracy	Recall	Precision	F1 score
DNN	0.9542	0.9542	0.9542	0.9542

4.3 실험 결과

4.3.1 XAI 기반 공격

XAI 기법 중 SHAP을 통해 주요 Feature를 산출하여 상위 Feature 값 변경을 통해 Adversarial example을 생성하였다. 42*42 이미지에서 하나의 열별로 3개의 주요 Feature의 값을 변경하였으며, 따라서 데이터 하나당 총 126개의 Feature 값을 변경하였다. 총 1,653개의 데이터 중 1,044개의 Adversarial example을 생성하여, 성공률은 63.1%로 확인하였으며, 1개의 Adversarial example 생성 시 평균 0.35초가 소요됨을 확인하였다.

4.3.2 Reference 기반 공격

악성코드 데이터 셋을 이용한 분류 모델에서 Reference 탐색 기반의 기법을 통해 Adversarial example을 생성하였다. Reference based attack에 사용된 파라미터 값은 표 8과 같다. 총 1,653개의 데이터 중 1,615개의 Adversarial example을 생성하여, 성공률은 97.7%로 확인하였으며, 1개의 Adversarial example 생성 시 평균 0.47초가 소요됨을 확인하였다.

(표 8) Reference 기반 공격 파라미터
(Table 8) Reference based attack Parameter

Parameter	Value
steps	500
learning_rate	0.02
lbd	0.05

4.3.3 PGD 공격

Adversarial attack 기법 중 대중적으로 사용되는 PGD attack을 통해 Adversarial example을 생성하였다. PGD attack에 사용된 파라미터 값은 표 9와 같다. 총 1,653개의 데이터 중 1,249개의 Adversarial example을 생성하여, 성공률은 75.5%로 확인하였으며, 1개의 Adversarial exmaple 생성 시 평균 20분이 소요됨을 확인하였다.

(표 9) PGD 공격 파라미터
(Table 9) PGD attack Parameter

Parameter	Value
norm	2
eps	0.01
eps_step	0.00001
max_iter	10000
num_random_init	5

4.3.4 성능 산출 결과 비교

XAI based attack, Reference based attack, PGD attak 기법을 통해 생성된 Adversarial example에 대하여 비교를 진행한다. 비교에 사용된 항목은 Adversarial example 생성에 성공한 데이터들의 평균적인 성공 시간, 성공 비율을 계산한 성공률이다.

악성코드 데이터 셋에 대하여 표 10과 같이 기존의 Adversarial example 생성 기법인 PGD attack을 비롯하여 제안한 2가지 기법에 대해 2가지 기준으로 비교하였다. 성공률은 Reference based attack이 97.7%로 가장 높았으며, PGD attack 75.5%, XAI based attack 63.1% 순으로 높았음을 알 수 있었다. 평균적인 Adversarial example 생성 시간은 XAI based attack이 0.35초로 가장 빠른 것으로 확인하였고, Reference based attack 0.47초, PGD attack 20분 순으로 빠름을 확인할 수 있었다. 동일한 환경의 CNN 모델을 사용하여 실험을 진행한 XAI based attack과 PGD attack을 비교하였을 때 XAI based attack은 성공률이 63.1%로 PGD attack의 75.5%로 다소 낮은 성공률을 보이나, 생성 시간의 경우 XAI based attack이 0.35초로 PGD attack의 20분에 비할 때 훨씬 빠른 시간 안에 Adversarial example이 생성 가능하므로 충분히 효율적인 Adversarial example 생성 기법으로 보인다. 또한 기존 악성코드 분류

(표 10) 성능 산출 결과 비교
(Table 10) Comparison of Performance Output Results

	XAI based attack (Proposed model - 1)	Reference based attack (Proposed model - 2)	PGD attack
Success rate	63.1%	97.7%	75.5%
Average generation time	0.35sec	0.47sec	20min

모델과 같은 모델인 DNN 모델을 사용하여 실험을 진행한 Reference based attack의 경우 성공률은 97.7%로 매우 높은 성공률을 보이며, 생성 시간 또한 0.47초로 매우 빠른 시간에 높은 성공률을 보이므로 효율적인 Adversarial example 생성 기법임을 확인할 수 있었다.

5. 결 론

보안 분야를 비롯하여 다양한 분야에 AI의 도입이 확산되고 있다. 하지만 AI 기술의 발전에 따라 AI의 알고리즘을 우회하는 다양한 공격 기법 또한 발전되고 있으며, AI 모델의 오 분류와 신뢰도 하락을 유도하는 Adversarial attack이 등장하였다. 머신러닝 알고리즘 내에 내재하고 있는 취약점에 의해 발생되는 AI 모델에 대한 Adversarial attack이 발생할 경우, 이러한 공격에 대응 가능하다는 보장이 없다. 따라서 Adversarial attack이 발생하지 않도록 AI의 견고성을 향상시키는 것이 앞으로 AI 보안의 핵심이다. 하지만 기존의 Adversarial attack은 일반적인 이미지 기반의 분류 모델을 대상으로 한다. 악성코드 분류 모델 또한 Adversarial attack에 취약한 DNN 기반의 분류 모델이지만 악성코드 분류 모델에 대한 Adversarial attack에 관한 연구는 거의 없다. 또한 기존의 Adversarial attack 기법으로 악성코드 데이터 셋을 통해 Adversarial attack을 진행하였을 때 Adversarial example에 대한 생성률이 높지 않으며, 매우 오랜 시간이 소요된다는 단점이 존재한다.

따라서 본 연구에서는 AI 모델의 Robustness 향상을 위한 XAI based attack 기법과 Reference based attack 기법으로 효율적인 Adversarial attack 기법 2가지를 제안하였으며, 악성코드 데이터 셋을 이용하여 Adversarial attack을 진행하고, 성능을 검증하였다. XAI based attack 기법은 모델 판단에 대한 기여도가 높은 Feature를 판단 후 적은 수의 Feature를 변경하여 모델의 판단을 변경하는 Adversarial example 생성을 통해 주요한 Feature의 변경만으로 모델의 판단 흐리는 Adversarial attack을 수행한다. 또한 Reference based attack 기법은 분류 모델의 경계를 효과적으로 탐색하여 원본과의 거리는 작지만 분류 모델의 판단 경계를 넘어서 라벨을 오 분류 하는 지점까지 조금씩 값을 변화시키며 Adversarial attack을 수행한다. 본 연구를 통해 제안한 2가지 방법으로 실험을 진행한 결과, 기존의 Adversarial attack인 PGD attack과 비교하였을 때, 두 가지 기법 모두 20분이 소요되었던 PGD attack에 비하여 XAI based attack이 0.35초, Reference based attack이 0.47초로 훨씬 빠른 시간 안에 Adversarial example 생성이 가능함을 확인하였다. XAI based attack의 경우 Adversarial example 생성 성공률이 63.1%로 기존 PGD attack의 75.5%에 비하여 낮은 성공률을 보이나 생성 시간이 50배가 넘게 차이가 나므로 기존 방식에 비하여 더욱 효율적인 방식으로 볼 수 있다. 또한 Reference based attack 기법의 경우 97.7%의 높은 성공률을 보이므로 기존 PGD attack에 비하여 더욱 빠르고 높은 성공률을 보임을 확인하였다. 따라서 본 논문을 통해 제안한 두 가지 기법을 통해 기존의 Adversarial attack보다 더욱 효율적인 Adversarial attack이 가능함을 확인할 수 있었다. 따라서 이러한 Adversarial attack에 대응 가능한 더욱 견고한 모델을 만들어야 하는 요구가 증가되고 있으며, 향후 존재할 Adversarial attack에도 대응이 가능한 견고한 AI 모델을 생성하기 위한 대비책이 필요하다. 본 논문에서 제시한 2가지의 효율적인 Adversarial attack 생성 기법으로 견고한 AI 모델을 구축하기 위한 연구에 기여할 수 있을 것으로 기대한다.

참고문헌(Reference)

[1] IBM. IBM Global AI Adoption Index, 2022. https://www.ibm.com/downloads/cas/

[2] Diro, Abebe Abeshu, and Naveen Chilamkurti. “Distributed attack detection scheme using deep learning approach for Internet of Things”, Future Generation Computer Systems, Vol.82, pp.761-768. 2018. https://doi.org/10.1016/j.future.2017.08.043

[3] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. “Explaining and harnessing adversarial examples”, arXiv preprint arXiv:1412.6572, 2014. https://doi.org/10.48550/arXiv.1412.6572

[4] Shafique, M., Naseer, M., Theocharides, T., Kyrkou, C., Mutlu, O., Orosa, L., & Choi, J. “Robust machine learning systems: Challenges, current trends, perspectives, and the road ahead”, IEEE Design & Test, Vol.37, No.2, pp.30-57, 2020. https://ieeexplore.ieee.org/document/8979377

[5] GU, Jindong. Explainability and Robustness of Deep Visual Classification Models. arXiv preprint arXiv:2301.01343, 2023. https://doi.org/10.48550/arXiv.2301.01343

[6] Im Choi, Jung, and Qing Tian. “Adversarial attack and defense of YOLO detectors in autonomous driving

scenarios", 2022 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2022. https://doi.org/10.1109/IV51971.2022.9827222

[7] SINGH, Jagsir; SINGH, Jaswinder. "A survey on machine learning-based malware detection in executable files", Journal of Systems Architecture, 2021. https://doi.org/10.1016/j.sysarc.2020.101861

[8] Moosavi-Dezfooli, Seyed-Mohsen, Alhussein Fawzi, and Pascal Frossard. "Deepfool: a simple and accurate method to fool deep neural networks", Proceedings of the IEEE conference on computer vision and pattern recognition, pp.2574-2582, 2016. https://openaccess.thecvf.com/content_cvpr_2016/html/Moosavi-Dezfooli_DeepFool_A_Simple_CVPR_2016_paper.html

[9] Papernot, N., McDaniel, P., Jha, S., Fredrikson, M., Celik, Z. B., & Swami, A. "The limitations of deep learning in adversarial settings", 2016 IEEE European symposium on security and privacy (EuroS&P), pp. 372-387, 2016. https://ieeexplore.ieee.org/document/7467366

[10] Chen, P. Y., Zhang, H., Sharma, Y., Yi, J., & Hsieh, C. J. "Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models", Proceedings of the 10th ACM workshop on artificial intelligence and security, pp. 15-26, 2017. https://doi.org/10.1145/3128572.3140448

[11] arlini, N., & Wagner, D. "Towards evaluating the robustness of neural networks," 2017 ieee symposium on security and privacy (sp), pp. 39-57, 2017. https://ieeexplore.ieee.org/document/7958570

[12] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., & Vladu, A. "Towards deep learning models resistant to adversarial attacks", arXiv preprint arXiv:1706.06083, 2017. https://doi.org/10.48550/arXiv.1706.06083

[13] Gunning, D., & Aha, D. "DARPA's explainable artificial intelligence (XAI) program", AI magazine, Vol.40, No.2, pp.44-58, 2019. https://doi.org/10.1609/aimag.v40i2.2850

[14] Ribeiro, Marco Tulio, Sameer Singh, and Carlos 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, 2016. https://doi.org/10.1145/2939672.2939778

[15] Lundberg, Scott M., and Su-In Lee. "A unified approach to interpreting model predictions", Advances in neural information processing systems, Vol.30, 2017. https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html

[16] Han, D., Wang, Z., Chen, W., Zhong, Y., Wang, S., Zhang, H., ... & Yin, X. "DeepAID: interpreting and improving deep learning-based anomaly detection in security applications", Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, pp. 3197-3217, 2021. https://doi.org/10.1145/3460120.3484589

◐ 저 자 소 개 ◑

정 시 온(Si-on Jeong)
2023년 호서대학교 컴퓨터공학부(공학사)
2023년~현재 호서대학교 대학원 정보보호학과(공학석사)
관심분야 : 악성코드 분석, 정보보호, AI
E-mail : 20181263@vision.hoseo.edu

한 태 현(Tae-hyun Han)
2018년~현재 호서대학교 컴퓨터공학부(공학사)
관심분야 : 악성코드 분석, 정보보호, AI
E-mail : 20181291@vision.hoseo.edu

임 승 범(Seung-bum Lim)
2023년 호서대학교 컴퓨터공학부(공학사)
2023년~현재 호서대학교 대학원 정보보호학과(공학석사)
관심분야 : 침입 탐지, 이상징후 탐지, 정보보호, AI
E-mail : 20171258@vision.hoseo.edu

이 태 진(Tae-jin Lee)
2003년 포항공과대학교 컴퓨터공학과(공학사)
2008년 연세대학교 컴퓨터공학과(공학석사)
2017년 아주대학교 컴퓨터공학과(공학박사)
2017년 한국인터넷진흥원 R&D 팀장
2017년~현재 호서대학교 정보보호학과 교수
관심분야 : 시스템 보안, 침해사고 대응, Trustworthty AI
E-mail : kinjecs0@gmail.com