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.2022.32.5.817

블랙 박스 모델의 출력값을 이용한 AI 모델 종류 추론 공격*

안 윤 수,[1†] 최 대 선[2‡]
[1,2]숭실대학교 (대학원생, 교수)

Model Type Inference Attack Using Output of Black-Box AI Model*

Yoonsoo An,[1†] Daeseon Choi[2‡]
[1,2]Soongsil University (Graduate student, Professor)

요 약

AI 기술이 여러 분야에 성공적으로 도입되는 추세이며, 서비스로 환경에 배포된 모델들은 지적 재산권과 데이터를 보호하기 위해 모델의 정보를 노출시키지 않는 블랙 박스 상태로 배포된다. 블랙 박스 환경에서 공격자들은 모델 출력을 이용해 학습에 쓰인 데이터나 파라미터를 훔치려고 한다. 본 논문은 딥러닝 모델을 대상으로 모델 종류에 대한 정보를 추론하는 공격이 없다는 점에서 착안하여, 모델의 구성 레이어 정보를 직접 알아내기 위해 모델의 종류를 추론하는 공격 방법을 제안한다. MNIST 데이터셋으로 학습된 ResNet, VGGNet, AlexNet과 간단한 컨볼루션 신경망 모델까지 네 가지 모델의 그레이 박스 및 블랙 박스 환경에서의 출력값을 이용해 모델의 종류가 추론될 수 있다는 것을 보였다. 또한 본 논문이 제안하는 방식인 대소 관계 피쳐를 딥러닝 모델에 함께 학습시킨 경우 블랙 박스 환경에서 약 83%의 정확도로 모델의 종류를 추론했으며, 그 결과를 통해 공격자에게 확률 벡터가 아닌 제한된 정보만 제공되는 상황에서도 모델 종류가 추론될 수 있음을 보였다.

ABSTRACT

AI technology is being successfully introduced in many fields, and models deployed as a service are deployed with black box environment that does not expose the model's information to protect intellectual property rights and data. In a black box environment, attackers try to steal data or parameters used during training by using model output. This paper proposes a method of inferring the type of model to directly find out the composition of layer of the target model, based on the fact that there is no attack to infer the information about the type of model from the deep learning model. With ResNet, VGGNet, AlexNet, and simple convolutional neural network models trained with MNIST datasets, we show that the types of models can be inferred using the output values in the gray box and black box environments of the each model. In addition, we inferred the type of model with approximately 83% accuracy in the black box environment if we train the big and small relationship feature that proposed in this paper together, the results show that the model type can be infrerred even in situations where only partial information is given to attackers, not raw probability vectors.

Keywords: AI security, Privacy, Exploratory Attack, Inference Attack

Received(07. 01. 2022), Modified(08. 29. 2022), Accepted(09. 19. 2022)

* 이 논문은 2022년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2020R1A2C1014813)

† 주저자, yoonsooh@soongsil.ac.kr
‡ 교신저자, sunchoi@ssu.ac.kr(Corresponding author)

I. 서 론

인공지능은 컴퓨터 비전, 자연어 처리 분야에서 서비스로 도입이 될 뿐만 아니라 국가적으로 중요한 의사결정을 해야 하는 군사 분야, 항공기의 시스템까지 영역을 넓혀나가고 있다. 상용화되는 서비스 모델(Machine Learning as a Service, MLaaS)[1]들은 지식 재산권과 정보 보호를 위해 모델 정보를 노출하지 않는 블랙 박스[2] 상태로 사용자들에게 배포된다.

그러나 블랙 박스 상태일지라도 배포된 모델은 공격자에 의해 학습 데이터 정보가 유출되거나, 모델의 성능을 저하시키거나 모델이 잘못 동작하게끔 유도하는 공격을 당할 위험성에 직면하게 된다[3,4,5]. 블랙 박스 모델 환경에서 타겟 모델 정보를 추출하는 공격에는 대표적으로 공격자가 모델에 질의해 타겟 모델과 유사한 대체 모델을 만드는 모델 추출 공격[6,7]이 있다. 모델 추출 공격이 대체 모델과 타겟 모델 간 기능의 유사성에 집중하며, 머신러닝 모델을 타겟으로 삼는 공격이라는 사실에서 착안하여 본 논문은 딥러닝 모델을 타겟으로 모델의 종류를 추론해 모델의 레이어 구성 정보를 직접 알아내는 방법을 제안한다.

딥러닝 모델의 종류를 추론할 수 있게 되면, 모델 추출 공격으로 타겟 모델의 기능을 모방하는 대체 모델을 생성하는 것보다 더욱 타겟 모델과 유사한 구조의 대체 모델을 쉽게 만들어 낼 수 있으므로 모델의 파라미터를 알아내기에도 용이해져 궁극적으로 블랙 박스 모델을 쉽게 화이트 박스 환경으로 만들 수 있다. 타겟 모델에 화이트 박스 환경으로 접근할 수 있다면, 대체 모델을 필요로 하는 모든 공격에 취약해진다. 공격자는 현실적으로 블랙 박스 모델에 접근할 확률이 높으므로, 대체 모델이 있다는 가정하에 적대적 공격을 수행하는 경우가 있다. 대체 모델을 사용하는 공격에는 모델 도치 공격(model inversion attack)[8], 블랙 박스 환경에서의 적대적 예제(adversarial example)[9], 멤버십 추론 공격(membership inf erence attack)[10,11], 중독 공격(poisoning at tack)[12,13] 등이 있다. 따라서 공격자가 모델의 종류를 추론해 레이어 구성 정보를 알 수 있게 되면 타겟 모델을 공격하기 위한 대체 모델 생성 시에 중요한 힌트가 되며, 타겟 모델을 상기한 바와 같이 여러 가지 공격에 노출시킬 수 있다.

본 논문은 정확도가 95% 이상인 이미지 분류기들을 타겟 모델로 두고 모델의 출력 확률 벡터를 얻을 수 있는 그레이 박스, 모델의 출력 확률 벡터를 얻을 수 없는 블랙 박스 환경에서 타겟 모델에 질의 해 얻을 수 있는 출력값을 이용하여 모델 종류를 추론하였고, 본 논문이 제안하는 피쳐 가공을 사용한 모델 종류 추론의 결과를 피쳐를 가공하지 않은 조건에서 모델 종류 추론 결과와 비교, 분석하였다.

본 논문이 기여한 바는 다음과 같다.

1. 블랙 박스 환경 모델의 출력값에서 모델의 종류 정보가 추론될 위험성을 최초로 보였다.

2. 본 논문이 제안하는 피쳐 가공 방법을 통해 모델 종류 정보 추론의 성능을 높였다.

3. 모델의 확률 벡터를 얻을 수 없는 상황에서도 80% 이상의 정확도로 모델 종류 추론이 가능함을 보였다.

본 논문의 구성은 다음과 같다. 2장에서 블랙 박스 환경에서의 탐색적 공격 기법 및 주요 CNN 모델들의 특성과 CNN 모델의 출력값의 특징에 대해 기술한다. 3장에서는 본 논문의 제안 방법인 모델 종류 추론 공격의 방법 및 모델의 출력값의 특징을 직관적으로 드러내는 피쳐 가공 방법을 설명하고, 4장에서는 블랙 박스 환경 및 그레이 박스 환경에서 출력값만을 사용한 모델 종류 추론 결과와 본 논문이 제안하는 피쳐 가공 방법을 사용한 모델 종류 추론 결과를 비교하는 실험을 수행하며, 5장의 결론으로 마무리를 짓는다.

II. 관련 연구 및 배경지식

2.1 블랙 박스 모델 환경에서의 탐색적 공격 기법

AI 모델 공격 유형으로는 크게 기만 공격(evasion attack)[14], 중독 공격, 탐색적 공격(exploratory attack)[15]이 있다.

이 중에서 탐색적 공격은 공개된 API가 존재하는 타겟 모델에서 입력값에 대해 출력된 분류 결과와 신뢰도(confidence)를 분석하여 공격자에게 숨겨진 정보를 알아내는 방식의 공격이다. 대표적으로 모델 추출 공격과 모델 도치 공격, 멤버십 추론 공격이 있다.

모델 추출 공격은 MLaaS에서 타겟 모델은 공개되지 않지만, 출력값을 반환한다는 점을 이용해 타겟 모델과 유사하게 작동하는 대체 모델을 만들어 내고, 타겟 모델을 학습시키는 데 쓰인 파라미터까지 알아내는 공격이다. 최소한의 질의를 통해 큰 비용을 들여 학습시킨 타겟 모델과 최대한으로 동일한 기능을 하는 모델을 생성해내는 것이 주목적이며, 공격자는

모델 추출 공격을 통해 얻은 대체 모델을 기반으로 추가적인 연계 공격을 수행할 수 있다.

모델 도치 공격은 모델에 질의를 해 얻을 수 있는 출력값과 신뢰도 값을 통해 모델의 학습 데이터를 복구하는 공격이다.

멤버십 추론 공격은 블랙 박스 환경에서 질의를 이용해 타겟 모델을 학습시키는 단계에서 특정 데이터가 사용되었는지 여부를 알아내는 공격이다. 타겟 모델과 유사한 여러 개의 그림자 모델을 생성하고, 해당 그림자 모델들의 학습에 사용된 데이터와 학습에 사용되지 않은 데이터를 각각 그림자 모델에 입력했을 때의 출력값을 훈련 데이터로, 해당 값들에 학습 데이터에 포함되었는지 여부를 정답 레이블로 학습시킨 멤버십 추론 공격 모델을 생성하는 방식이다.

본 논문은 타겟 모델의 훈련된 조건에 따라 쿼리 아웃풋이 다르다는 점을 이용한 멤버십 추론 공격의 방식에서 아이디어를 얻어, 데이터가 각기 다른 종류의 모델에서 학습되어 출력된 값들을 훈련 데이터로, 모델의 종류를 정답 레이블로 학습시킨 모델 종류 분류기를 공격 모델로 학습시켜 모델 종류를 분류한다.

2.2 대표적인 CNN 모델의 종류와 특징

AlexNet[16]은 ILSVRC(ImageNet Large-Scale Visual Recognition Challenge)[17] 2012년 대회에서 1위를 차지한 네트워크이며, AlexNet의 영향으로 CNN에서 드랍아웃(dropout)의 적용을 통한 과적합 방지 및 GPU 여러 개를 사용하는 병렬처리[18] 학습을 통한 학습 효율 개선이 보편화 되었다. AlexNet은 5개의 컨볼루션 레이어(convolutional layer)와 3개의 완전 연결 계층 (fully-connected layer)의 구조를 가진 모델이다. 활성화 함수로는 ReLU[19] 함수를 적용한다. 또한, 오버래핑 풀링 (overlapping pooling)과 LRN(Local Response Normalization)을 사용해 정확도를 개선한 것이 특징이다.

VGGNet[20]은 ILSVRC 2014 대회에서 준우승을 거둔 네트워크이다. VGGNet은 모델의 깊이에 따라 A, A-LRN, B, C, D, E가 있는데 이 중 VGG16(D), VGG19(E)가 이미지 인식에 널리 쓰인다. 모델의 깊이가 객체 인식의 에러율에 미치는 영향을 중심으로 연구한 네트워크이며, 모델의 깊이가 깊을수록 성능이 좋기 때문에, VGGNet은 가장 작은 3×3 convolution 필터를 모든 레이어에 사용해 최대한 많은 ReLU 활성화 함수를 적용했다. 그로 인해 비선형성을 더 강하게 가져 뛰어난 성능을 낼 수 있으며, VGGNet 이후의 CNN모델들은 이전보다 깊은 구조를 가지게 되었다.

많은 레이어를 가진 모델의 성능이 항상 뛰어난 것은 아니다. 기울기 소실(gradient vanishing)[21] 및 기울기 폭발(gradient explosion)[22] 의 문제가 있기 때문이다. 기울기 소실과 기울기 폭발은 각각 깊은 구조의 인공 신경망의 학습 중 역전파(back propagation) 단계에서 입력 레이어에 가까워질수록 기울기가 점차 작아지는 현상과 기울기가 점차 커져 발산하는 현상을 말한다. 기울기 소실 혹은 기울기 폭발로 인해 가중치가 정상적으로 업데이트되지 않으면, 최적의 모델로 학습시킬 수 없게 된다. ResNet[23]은 skip connection(shortcut)과 잔차 학습(residual learning)을 통해 신경망 모델이 깊어짐에 따라 작은 미분값이 0으로 수렴하거나, 큰 미분값이 지나치게 큰 값으로 발산하는 기울기 소실, 기울기 폭발 문제를 해결하고 충분히 깊은 구조에서 높은 성능을 가질 수 있게 한 네트워크이다.

2.3 모델에서 출력되는 확률 벡터의 특징

많은 CNN 모델들이 이미지를 입력으로 받고, 각 레이블별 확률 벡터를 출력으로 내놓는데, 마지막 레이어의 활성화 함수는 주로 소프트 맥스이다. 출력 벡터는 소프트 맥스 활성화 함수를 거친 뒤의 0~1 사이의 정규화된 확률값을 원소로 가진다.

학습된 모델에 데이터를 입력했을 때 출력되는 확률 벡터는 모델의 학습 정도에 따라 차이가 있다. 예를 들어 분류 정확도가 95%가량인 타겟 모델에 데이터를 입력하면 해당 데이터의 클래스에 분류될 확률이 약 0.95가 되고, 나머지 클래스들로 분류될 확률은 합쳐서 0.05가 안되는 작은 값이다. 반면, 분류 정확도가 10% 정도로 낮은 학습이 덜된 모델에 같은 데이터를 입력했을 때 정답 클래스에 분류될 확률이 평균적으로 약 0.1이고 그 외 나머지 클래스에 분류될 확률의 합이 평균적으로 약 0.9가 된다. 타겟 모델이 높은 분류 정확도를 가지도록 학습될수록 모델의 출력 확률 벡터는 정답인 클래스 하나에만 큰 값이 집중되며 타겟 모델이 덜 학습 될수록 모델의 출력 확률 벡터는 정답 클래스를 포함한 여러 클래스에 제각각의 값을 가진 벡터가 된다. 이 벡터들을 이용한 모델 종류 추론 공격과 그 성능을 개선시키기

위한 피쳐 가공 방법을 3장에서 제안하고, 피쳐를 추가해 개선된 모델 종류 분류 결과와 원본 출력값을 사용한 모델 종류 분류의 성능을 4장에서 비교한다.

III. 제안 방법

3.1 개요

본 논문에서 제안한 모델 종류 추론의 목적은 블랙 박스 모델 환경 및 그레이 박스 모델 환경에서 공격자가 얻을 수 있는 출력값으로 타겟 모델의 종류를 알아내는 것이다. 타겟 모델이 배포된 이후, 타겟 모델에 공격자가 쿼리 권한을 가진다고 가정한다. 공격자는 가지고 있는 데이터를 타겟 모델에 질의해 출력값을 얻는다. 이때 모델의 출력값으로 확률 벡터를 얻을 수 있는 환경을 그레이 박스 환경, 클래스별 확률의 랭킹 정보를 얻을 수 있는 환경을 블랙 박스 환경이라고 가정한다. 모델별 출력값이 어떤 종류의 모델에서 출력된 것인지 분류하기 위해 타겟 모델과 같은 데이터로 학습된 후보 모델군에 질의해 얻은 출력값을 학습시킨 MLP 모델을 만들고, 모델 종류 분류기(Model Type Classifier, 이하 MTC)라 지칭한다. MTC의 성능은 MTC를 훈련시키는데 사용되지 않은 데이터셋의 쿼리 데이터의 분류 정확도 및 ROC 곡선[24], 정밀도-재현율 그래프를 통해 평가한다.

본 논문은 제안한 모델 종류 분류의 성능을 모델의 출력값만 사용했을 때보다 향상시키기 위해 데이터를 직관적이고 노이즈가 적어지게 하는 피쳐의 가공 방법도 함께 제안한다. 피쳐들은 모두 그레이 박스, 블랙 박스 환경에서 각각 얻을 수 있는 출력값을 사용해 가공할 수 있다. 최종적으로 모델의 출력값과 그 출력값을 사용해 가공된 피쳐를 합친 데이터로 모델의 종류를 추론한다. 모델 종류 분류의 방법은 Fig. 1.과 같다. 그레이 박스, 블랙 박스 환경에서 각각 얻을 수 있는 모델의 출력값과 가공 가능한 피쳐는 Table 1.과 같으며, 피쳐 가공의 방법은 Fig. 2와 같다.

Table 1. Processable features based on the amount of information an attacker can access

	Adversarial knowledge	Processable features
Gray box	Probability vector, Ranking Lables by Probability	**feature 1**
Black box	Ranking Labels by probability	**feature 2**

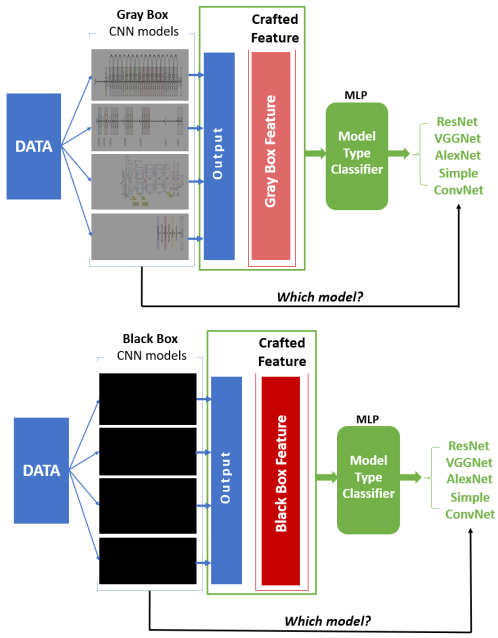

Fig. 1. Method of model type inference

Algorithm. Feature processing

Input : set of candidate target model F_{zoo} (trained on dataset D), Q_{in} : query input dataset where $Q_{\text{in}} \not\subset D$, Env : set of environment.
Function: $Conf\,Val(F, Q_{\text{in}})$: returns the confidence value of Q_{in} for each classes from classifier F
Function: $MC(C)$: returns the median index of C, the number of classes in D.
Function: $LBrank(Q_{\text{in}})$: returns the ascending sort of the probabilities output from F
Function: $Concat_{R/C}(mat1, mat2, mat3)$: concatenate all inputs in row or column
Output : D_{adv}:Data used to train MTC
1: $F_{zoo} = \{F_1, F_2, F_3, \ldots, F_n\}$
2: $Env = \{\text{Black box } E_B, \text{ Gray box } E_G\}$
3: **if** $Env = E_G$ **do**
4: **for** $n = \{1, 2, 3, \ldots, N\}$ **do**

$Q_{out1} = Conf\,Val(F[N],\ Q_{\text{in}})$
$Q_{out2} = LBRank(F[N],\ Q_{\text{in}})$
$(IDX,\ C)$ = the size of Q_{out}
for $Idx = \{1, 2, 3, ..., IDX\}$
  Delete $\arg max(Q_{out1}[Idx])$
  $QM = mean(Q_{out1}[Idx])$
  for $c = \{1, 2, 3, ..., C\}$
    if $Q_{out1}[Idx][c] < Q_{\text{out}}M$
      $Q_{out1}[Idx][c] \leftarrow 0$ replace
    else
      $Q_{out1}[Idx][c] \leftarrow 1$ replace
      $feature\,1 = Concat_R(Q_{out1}[Idx])$
    $D_{\text{adv}} = Concat_C(Q_{out1}, Q_{out2}, feature\,1)$
else if $Env = E_B$ **do**
  for $n = \{1, 2, 3, ..., N\}$ **do**
    $Q_{out} = LBRank(F[N],\ Q_{\text{in}})$
    for $Idx = \{1, 2, 3, ..., IDX\}$ do
      for $c = \{1, 2, 3, ..., C\}$ do
        if $c < MC(C)$ do
          $Q_{\text{out}}[Idx][c] \leftarrow 0$ replace
        else
          $Q_{\text{out}}[Idx][c] \leftarrow 1$ replace
      $feature\,2 = Concat_R(Q_{\text{out}}[Idx])$
    $D_{\text{adv}} = Concat_C(Q_{out}, feature\,2)$
return D_{adv}

Fig. 2. Feature processing method

3.2 Gray box feature: feature 1

본 논문은 모델의 클래스별 확률 벡터와 클래스별 확률의 내림차순 랭킹 정보를 모두 얻을 수 있는 환경을 그레이 박스 환경으로 정의했다. 그레이 박스 환경에서 얻을 수 있는 정보인 타겟 모델에 의해 예측된 클래스별 확률 벡터는 높은 정확도를 가지는 모델에서 출력된 벡터일 경우 소프트 맥스 함수의 영향으로 인해 정답 클래스에만 큰 확률값이 존재하고 나머지 클래스에 남아있는 확률값들의 절댓값이 매우 작아지므로 그 특성이 직관적으로 드러나는 것이 억제된다. 또한 정답 클래스의 정보 외에 전달할 수 있는 정보의 양도 희소해진다. 따라서 모델에 의해 예측된 클래스별 확률 벡터를 기계학습에 사용할 수 있는 정보로 가공하려면, 소프트맥스 함수의 영향력을 보정하는 과정이 필요하다. 비슷한 예로, 모델 압축 기법인 지식 증류(knowledge distillation)는 큰 모델의 클래스별 확률 분포를 작은 모델에게 전달되는 정보로써 효율적으로 사용하기 위해 소프트맥스 함수에 온도(temperature) 파라미터를 도입해, 온도 파라미터를 크게 설정할수록 소프트맥스 함수의 특성을 완화시켜 큰 확률값은 작아지게, 작은 확률값은 커지게 하는 과정을 거친다. 이 과정을 통해 소프트 라벨 사용의 이점이 커진다. 이 방법은 소프트맥스 이전 레이어의 로짓 값을 알 수 있을 때 적용할 수 있다.본 논문은 위의 예에서 아이디어를 얻어 로짓 벡터를 얻을 수 없는 상황에서 소프트맥스 함수의 영향력을 보정해 원소들간의 상대적 대소관계를 드러내는 feature 1을 가공한다. feature 1은 Fig. 2.의 7~14행과 같이 모델의 예측 결과로 출력된 확률 벡터를 사용해 가공한다. 정답 클래스의 확률값을 제외한 나머지 클래스의 확률값들의 평균값을 구하고 (Fig. 2. 9~10열), 평균보다 작은 값을 가지는 클래스에는 0, 평균보다 큰 값을 가지는 클래스에는 1로 대입한 벡터가 feature 1이 된다. Fig. 2.의 15열과 같이 모델별 출력 확률 벡터와 모델의 클래스별 확률의 내림차순 랭킹, 가공한 feature 1을 합친 데이터를 사용해 그레이 박스 환경에서 모델의 종류를 추론하는 MTC를 훈련시킨다.

3.3 Black box feature : feature 2

모델 상용화 단계에서 모델을 학습시킬때에는 주로 소프트맥스 레이어를 사용하고, 상용화된 후에는 비용을 절감하기 위해 소프트맥스 레이어를 제외하는 경우가 많다. 소프트맥스 레이어 이전의 로짓 값과 소프트맥스 레이어 이후의 출력 확률 벡터 값은 동일한 대소 관계를 가지므로 가장 큰 확률을 가지는 레이블을 정답으로 내놓으면 되기 때문이다. 따라서 현실적으로 모든 클래스별 확률 벡터보다는 확률값의 랭킹 정보에만 공격자가 접근 가능할 확률이 더 높기 때문에, 확률값의 랭킹 정보를 이용해 모델의 종류 정보를 추론할 수 있다면, 공격자가 비교적 적은 정보로도 수행할 수 있는 매우 효율적인 공격 방법이 된다. 본 논문은 클래스별 확률의 랭킹 정보에만 공격자가 접근할 수 있는 환경을 블랙 박스 환경이라고 정의하고, 블랙

박스 환경에서도 공격을 수행했다.

블랙 박스 환경에서 가공할 수 있는 피쳐 2의 가공 방법은 피쳐 1과 매우 유사하다. 확률 벡터를 얻을 수 없는 블랙 박스 환경에서는 확률값들의 평균을 구할 수 없으므로 피쳐 1과 동일한 방식으로 피쳐를 가공할 수는 없다. 그 대신 모델에서 출력된 확률값의 랭킹 정보를 얻을 수 있으므로, 최상위부터 중위 클래스에는 1, 그다음 클래스부터 최하위 클래스에는 0으로 해당 클래스의 확률값을 대입한 벡터를 생성한다. 피쳐 3의 가공 방법은 Fig. 2.의 20~26열과 같다. Fig. 2.의 27열과 같이 모델의 출력 랭킹정보에 feature 2을 합친 학습 데이터를 사용해 블랙 박스 환경에서 모델의 종류를 추론하는 MTC를 훈련시킨다.

IV. 모델 종류 추론 공격 실험

3장의 제안 방법을 사용해 모델 종류 추론 실험을 수행하고 결과를 통해 제안 방법이 모델의 종류 추론 공격 방법으로 유효한지 확인하였다.

MNIST[25] 데이터가 학습된 네 가지 CNN 모델 네 가지를 타겟 모델의 후보 모델 집합으로 삼고, 그레이 박스 환경에서는 네 가지 CNN 모델의 출력 확률 벡터와 랭킹 정보를 이용해 모델의 종류를 분류하는 MTC를, 블랙 박스 환경에서는 네 가지 CNN 모델의 랭킹 정보를 이용해 모델의 종류를 분류하는 MTC를 생성한다. 그 후 본 논문이 제안한 방법에 따라 그레이 박스 환경에서는 피쳐 1을, 블랙 박스 환경에서는 피쳐 2를 훈련 데이터에 추가해 MTC의 성능을 향상시킨다.

4.1 실험 방법

4.1.1 타겟 후보 모델 학습

타겟 모델의 후보가 될 CNN 모델 네 가지 중 세 가지를 AlexNet, ResNet, VGGNet구조를 기반으로 모델링하였고 나머지 한가지 모델은 두 개의 컨볼루션 레이어와 두 개의 맥스풀링(max pooling), 드랍아웃 레이어(0.5)를 가진 단순한 구조의 컨볼루션 신경망 (Simple MNIST Convolution Network simple -ConvNet[26], 이하 Simple ConvNet)으로 모델링해 네 가지 후보 모델군을 MNIST 데이터셋 30,000장의 손글씨 사진으로 학습시킨다. 타겟 후보 모델들의 분류 정확도는 Table 2와 같이 95% 이상의 분류 정확도를 가지는 AlexNet, ResNet, VGGNet, simple ConvNet 네 가지 모델들을 실험에 사용한다. 테스트 정확도 또한 훈련 정확도와 유사한 성능을 가지도록 학습되었다.

Table 2. MNIST classification accuracy of candidate target models

Model Type	accuracy
AlexNet	**0.9969**
ResNet	**0.9627**
VGGNet	**0.9699**
Simple ConvNet	**0.9502**

4.1.2 모델 종류 분류기 (MTC)

MTC는 실험에 사용된 CNN 모델로부터 얻은 벡터를 사용해 모델의 종류를 분류하는 모델이다. 모델 종류 분류기는 입력층, 출력층과 완전 연결 계층 두 개로 이루어져 있고, 은닉층의 활성화 함수는 ReLU를, 출력층의 활성화 함수는 소프트맥스를 사용한 MLP 모델이며, 학습데이터로 네 가지 모델의 확률 벡터를 사용한다. 이때, MNIST 학습데이터 중 타겟 모델 학습에 쓰이지 않은 30,000개의 데이터를 타겟 모델에 입력해서 얻은 출력 확률 벡터를 MTC의 훈련 데이터로 사용한다. 한 모델마다 30,000개씩, 총 120,000개의 출력 확률 벡터로 MTC를 학습시킨다. 정답으로는 네 가지의 모델 정답 레이블을 사용한다. MTC를 학습시킬 때 학습률은 0.001, 옵티마이저(optimizer)는 RMSprop, 배치 사이즈(batch size)는 256, 에포크(epoch)는 100으로 설정하고 테스트 정확도와 훈련 정확도를 통해 MTC의 성능을 실험 조건별로 비교한다. 본 논문의 모든 실험은 동일한 모델에 동일한 파라미터를 고정해 사용하였다.

4.2 그레이 박스 환경에서의 모델 종류 추론

공격자가 모델별 출력 확률 벡터와 클래스별 확률의 랭킹 정보를 훈련시킨 MTC를 통해 95% 이상의 분류 정확도를 가지는 AlexNet, ResNet, VGGNet, Simple ConvNet 모델에서 얻을 수 있는 확률 벡터들을 모델 종류에 따라 분류한다. 모델 종류 분류 결과는 Table 3. 의 Gray box attack 열과 같다. 모델의 출력 확률 벡터만을 훈련시킨 MTC의 정확도가 약 0.8로, 공격자가 모델별 출력값을 이용해 모델의 종류를 추론할 위험성이 있음

Table 3. MTC performance in graybox, blackbox model type inference attack

	MTC performance			
	Gray box attack		Black box attack	
Data / Metric	Graybox output	output+ feature 1	Blackbox output	output+feature2
accuracy	0.8072	0.8839	0.6693	0.8279
ROC_AUC	0.9488	0.9799	0.8720	0.9584

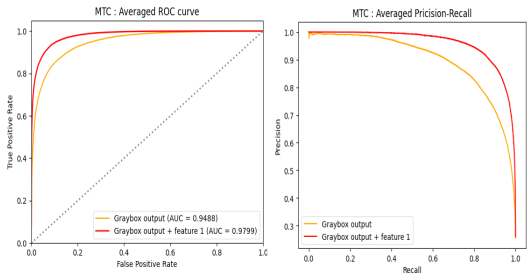

Fig. 3. ROC-Curve and precision-recall graph of MTC in gray box

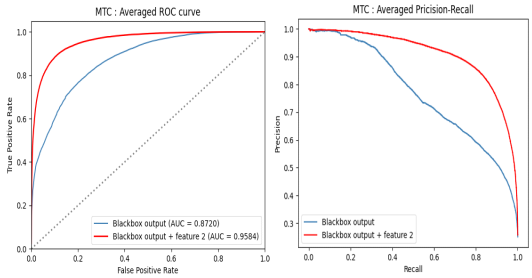

Fig. 4. ROC-Curve and precision-recall graph of MTC in black box

을 알 수 있다. 피쳐를 사용해 학습시킨 MTC의 정확도는 약 0.88로, 피쳐를 이용하면 더 효율적으로 모델의 종류가 추론됨을 확인할 수 있다. 또한, Fig. 3.의 AUC 및 정밀도-재현율 그래프를 기준으로 feature 1을 추가한 데이터를 MTC에 학습시켰을 시 성능이 향상되었다.

4.3 블랙 박스 환경에서의 모델 종류 추론

공격자가 모델에서 출력된 클래스별 확률의 랭킹 정보만을 알 수 있는 상황인 블랙 박스 모델 환경에서 MTC가 모델 종류를 정확하게 추론할 수 있다면 매우 큰 취약점이 될 수 있다. 실제로 공격자는 공격하고자 하는 모델을 블랙 박스 환경으로 제공 받을 가능성이 높기 때문이다. 블랙 박스 환경에서의 모델 종류 분류 결과는 Table 3.의 Black box attack 열과 같다. 모델의 클래스별 랭킹정보만을 훈련시킨 MTC의 정확도가 약 0.66으로, 그레이 박스 환경에 비해 비교적 제한적인 정보를 제공 받으므로 모델 종류 추론의 성능이 낮다. 그러나 본 논문이 제안하는 피쳐 가공 방법을 사용한 결과, MTC의 정확도가 약 0.83으로 향상되는 것을 확인할 수 있으며, Fig. 4.의 ROC 곡선과 정밀도-재현율 그래프상으로도 피쳐 가공의 효용성을 확인할 수 있다. 공격자는 모델의 출력 확률 벡터 없이도 모델의 출력값과 간단한 피쳐 가공을 통해 블랙 박스 환경에서도 모델의 종류를 추론할 가능성이 있다.

V. 결 론

본 논문은 CNN 모델별 출력 확률 벡터를 학습시킨 MTC를 이용해, 그레이 박스 및 블랙 박스 환경에서 모델의 출력값으로 모델의 종류 정보를 추론할 수 있음을 보였다. 또한 모델별 출력값의 특징을 표현한 피쳐를 가공해 함께 학습시키면 높은 정확도로 빠르게 수렴하는 개선된 MTC를 생성할 수 있음을 보였다. 추가로 모델의 출력 확률 벡터 없이, 모델별 출력 클래스의 랭킹 정보만으로도 모델의 종류 정보가 추론될 수 있다는 사실을 밝혔다.

모델의 정보 추론은 MTC의 파라미터나 구조, 데이터의 특성, 제공되는 정보의 양 등의 요소에 따라 성능이 달라질 수 있지만 성능이 40% 정도로 매우 저조한 MTC를 생성했을 때에도 본 논문에서 제안한 피쳐를 가공해 학습시켰을 때 모델 추론 성공률이 크게 높아짐과 동시에 빠르게 수렴하는 경향을 보였으며, 이는 본 논문의 제안 방법을 사용하면 더 높은 성능을 가진 MTC를 생성할 수 있음을 뜻한다.

모델 종류 정보가 추론되면 연계 공격을 통해 모

델의 내부 정보가 화이트 박스 상태로 공개될 수 있다. 모델의 구조를 포함한 내부 정보를 모두 알 수 있다면, 대체 모델을 필요로 하는 모델 도치 공격, 중독 공격, 블랙 박스에서의 적대적 예제, 멤버십 추론 등 여러 공격에 취약해진다. 이는 모델을 상용화할 때, 정확도가 높은 모델을 매우 제한적인 정보만 허용하는 상태로 배포해야 함을 뜻한다. 앙상블 학습을 통해 여러 모델을 합쳐 사용하는 것 또한 배포된 모델의 종류 정보가 추론되는 위험성에서 벗어나는 방법이 될 수 있을 것이다.

모델 종류 추론은 모델별로 오답의 패턴을 MTC에 학습시키는 방식이다. MNIST 데이터셋의 경우 열 개의 클래스로 이루어져 있기 때문에, 오답의 패턴을 아홉 개의 레이블 안에서 파악해야 하지만, CIFAR-100 등의 많은 수의 클래스를 가진 데이터셋에서는 오답의 패턴이 담긴 레이블의 수가 훨씬 많기 때문에 더 많은 변수를 가질 것이다. 또한, CNN의 발전 과정에서 많은 종류의 모델들이 이미지 분류에 쓰이고 있으므로, 타겟 모델일 것이라고 예상되는 모델의 종류가 더 많은 상황을 설계해 실험을 수행하면 모델 추론 결과에 변수가 생길 수 있다는 한계가 있다.

향후 모델의 종류를 예측하는 데 가장 최적화된 노이즈를 포함하는 이미지 데이터를 생성해, 여러 번의 질의로 모델 종류를 추론하는 것보다 간단하게 모델의 종류 정보를 추론하는 방법을 연구할 예정이다.

References

[1] M. Ribeiro, K. Grolinger and M. A. M. Capretz, "MLaaS: Machine Learning as a Service," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 896-902, Dec 2015.

[2] A. Ilyas, L. Engstrom, A. Athalye and J. Lin "Black-box Adversarial Attacks with Limited Queries and Information," Proceedings of the 35th International Conference on Machine Learning, PMLR vol. 80, pp. 2137-2146, Jul 2018.

[3] K. Ren, T. Zheng, Z. Qin and X. Liu, "Adversarial Attacks and Defenses in Deep Learning," Engineering vol. 6, no. 3, pp.346-360, March. 2020.

[4] M.Barreno, B. Nelson, A.D..Joseph and J.D. Tygar, "The security of machine learning," Machine Learning 81, pp. 121 - 148, May. 2010.

[5] A. Oseni, N. Moustafa, H. Janicke, P. Liu, Z. Tari and A. Vasilakos, "Security and Privacy for Artificial Intelligence: Opportunities and Challenges," arXiv, Feb. 2021.

[6] O. Bastani, C. Kim, and H. Bastani. "Interpreting Blackbox Models via Model Extraction," arXiv, May. 2017.

[7] M. kesarwani, B. Mukhoty, V. Arya and S. Mehta. "Model Extraction Warning in MLaaS Paradigm," arXiv, Nov. 2017.

[8] M. Fredrikson, S. Jha and T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15), pp. 1322- 1333, Oct 2015.

[9] W. Brendel J. Rauber and M. Bethge "Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models," International Conference on Learning Representations. Feb 2018.

[10] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership Inference Attacks Against Machine Learning Models," 2017 IEEE Symposium on Security and Privacy (SP), pp. 3-18, June. 2017.

[11] J. Hayes, L. Melis, G. Danezis and E. D. Cristofaro, "LOGAN: Membership inference attacks against generative models," arXiv, Aug, 2018.

[12] S. Alfeld, X. Zhu and P. Barford, "Data Poisoning Attacks against Autoregressive Models," Proceedings of the AAAI Conference on Artificial Intelligence. vol.30, no.1, Feb 2016.

[13] M. Jagielski et al, "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning," 2018 IEEE Symposium on Security and Privacy (SP), pp. 19-35, May. 2018.
[14] B. Biggio. et al. "Evasion Attacks against Machine Learning at Test Time," In Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2013. Lecture Notes in Computer Science, vol. 8190, pp. 387-402, Sep. 2013.
[15] T. S. Sethi and M. Kantardzic, "Data driven exploratory attacks on black box classifiers in adversarial domains," Neurocomputing vol. 289, pp. 129-143, Mar. 2018.
[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Association for Computing Machinery, vol.60, 6 pp.84 - 90, June. 2017.
[17] O. Russakovsky, J. Deng, H. Su et al., "ImageNet Large Scale Visual Recognition Challenge," International Journal of Computer Vision 115, pp. 211 - 252, Apr. 2015.
[18] N. Zhang, Y. Chen and J. Wang, "Image parallel processing based on GPU," 2010 2nd International Conference on Advanced Computer Control, pp. 367-370, June 2010.
[19] A. F. Agarap, "Deep Learning using Rectified Linear Units (ReLU)," arXiv, Feb. 2019.
[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv, Apr. 2015.
[21] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," International Journal of Uncertainty, Fuziness Knowledge-Based Systems. vol. 6, no. 2, pp.107 - 116, April. 1998.
[22] R. Pascanu, T. Mikolov and Y. Bengio ,"On the difficulty of training recurrent neural networks," Proceedings of the 30th International Conference on Machine Learning, PMLR, vol. 28 no. 3, pp. 1310-1318, Jun 2013.
[23] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778, June 2016.
[24] T. Fawcett, "An introduction to ROC analysis," in Pattern Recognition Letters, vol.27 no.8 pp. 861-874, Dec. 2005.
[25] L. Deng, "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]," in IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 141-142, Nov. 2012.
[26] Keras code example(Computer Vision), Simple MNIIST convnet, Available :https://keras.io/examples/vision/mnist_convnet/

〈저자소개〉

안 윤 수 (Yoonsoo An) 학생회원
2019년 2월: 공주대학교 응용수학과 학사
2021년 9월~현재: 숭실대학교 융합소프트웨어학과 석사과정
〈관심분야〉 금융보안, 딥러닝, 강화학습, 메타버스 보안, 연합학습

최 대 선 (Daeseon Choi) 종신회원
1995년 2월: 동국대학교 컴퓨터공학과 학사
1997년 2월: 포항공과대학교 컴퓨터공학과 석사
2009년 1월: 한국과학기술원 전산학과 박사
1997년 1월~1999년 6월: 현대정보기술 선임
1999년 7월~2015년 8월: 한국전자통신연구원 인증기술연구실 실장/책임연구원
2015년 9월~2020년 8월: 공주대학교 의료정보학과 부교수
2020년 9월~현재: 숭실대학교 소프트웨어학부 교수
2016년~현재: 정보보호학회 이사
〈관심분야〉 인증, 개인정보보호, 이상거래탐지, 의료정보보안, 머신러닝