Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.31, NO.6, Dec. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.6.1227

데이터 예측 클래스 기반 적대적 공격 탐지 및 분류 모델

고 은 나 래,[1†] 문 종 섭[2‡]
[1,2]고려대학교 (대학원생, 교수)

Adversarial Example Detection and Classification Model Based on the Class Predicted by Deep Learning Model

Eun-na-rae Ko,[1†] Jong-sub Moon[2‡]
[1,2]Korea University (Graduate student, Professor)

요 약

딥러닝 분류 모델에 대한 공격 중 하나인 적대적 공격은 입력 데이터에 인간이 구별할 수 없는 섭동을 추가하여 딥러닝 분류 모델이 잘못 분류하도록 만드는 공격이며, 다양한 적대적 공격 알고리즘이 존재한다. 이에 따라 적대적 데이터를 탐지하는 연구는 많이 진행되었으나 적대적 데이터가 어떤 적대적 공격 알고리즘에 의해 생성되었는지 분류하는 연구는 매우 적게 진행되었다. 적대적 공격을 분류할 수 있다면, 공격 간의 차이를 분석하여 더욱 견고한 딥러닝 분류 모델을 구축할 수 있을 것이다. 본 논문에서는 공격 대상 딥러닝 모델이 예측하는 클래스를 기반으로 은닉층의 출력값에서 특징을 추출하고 추출된 특징을 입력으로 하는 랜덤 포레스트 분류 모델을 구축하여 적대적 공격을 탐지 및 분류하는 모델을 제안한다. 실험 결과 제안한 모델은 최신의 적대적 공격 탐지 및 분류 모델보다 정상 데이터의 경우 3.02%, 적대적 데이터의 경우 0.80% 높은 정확도를 보였으며, 기존 연구에서 분류하지 않았던 새로운 공격을 분류한다.

ABSTRACT

Adversarial attack, one of the attacks on deep learning classification model, is attack that add indistinguishable perturbations to input data and cause deep learning classification model to misclassify the input data. There are various adversarial attack algorithms. Accordingly, many studies have been conducted to detect adversarial attack but few studies have been conducted to classify what adversarial attack algorithms to generate adversarial input. if adversarial attacks can be classified, more robust deep learning classification model can be established by analyzing differences between attacks. In this paper, we proposed a model that detects and classifies adversarial attacks by constructing a random forest classification model with input features extracted from a target deep learning model. In feature extraction, feature is extracted from a output value of hidden layer based on class predicted by the target deep learning model. Through Experiments the model proposed has shown 3.02% accuracy on clean data, 0.80% accuracy on adversarial data higher than the result of pre-existing studies and classify new adversarial attack that was not classified in pre-existing studies.

Keywords: Adversarial Attack, Evasion Attack, Deep Learning, Adversarial Example Detection

I. 서 론

딥러닝 시스템은 높은 정확도와 자동화로 인해 자율주행[1], 의료 진단[2] 등 Safety가 중요한 애플리케이션을 포함 매우 다양한 분야에서 활용되고 있으므로, 잘못된 딥러닝 분류 모델 결과는 실세계의

Received(10. 26. 2021), Modified(12. 03. 2021), Accepted(12. 03. 2021)

† 주저자, eun13@korea.ac.kr
‡ 교신저자, jsmoon@korea.ac.kr(Corresponding author)

심각한 결과를 초래할 수 있다. 따라서 외부 공격에 강력한 딥러닝 분류 모델을 구축하는 것이 중요하다.

딥러닝을 사용한 분류 모델에 대한 공격 중, 적대적 공격(Adversarial Attack)은 입력 데이터에 인간이 구별할 수 없는 섭동을 추가하여 분류 모델이 오 분류 하도록 하는 적대적 데이터를 생성하는 공격이며, 이에 대한 수 많은 연구가 진행되고 있다[3-8].

Fig 1.은 적대적 공격의 예시이다. 분류 모델은 원본 이미지를 'jeep, landrover, 609'로 분류하지만, 적대적 공격으로 생성한 데이터에 대해서는 'beach wagon, 436'으로 잘못 분류하고 있다. 적대적 공격의 특성상 원본 데이터와 적대적 데이터의 차이는 인간이 구별할 수 없는 정도로 미세하므로, 이를 방어 또는 탐지하는 연구 또한 활발하게 진행되고 있다.

방어 기법의 경우 적대적 데이터로 모델을 재훈련시키는 방법[9, 10], 전처리[11] 및 그래디언트(gradient) 마스킹[12] 등을 통해 딥러닝 모델 자체를 공격에 대해 견고하게(Robust) 만들어 적대적 데이터를 올바르게 분류하도록 한다.

탐지 기법의 경우 통계 기반 방법[14], 적대적 데이터와 정상 데이터에 대한 딥러닝 모델의 다른 동작을 식별하여 탐지하는 방법[15-17], 특징 압축 방법[19] 등을 통해 입력 데이터가 딥러닝 모델에 전달되기 전에 적대적 공격으로 생성된 데이터인지 탐지하여 막는 방식이다.

적대적 공격으로 생성된 적대적 데이터를 탐지하는 방식에 더 나아가서, 실제로 어떤 적대적 공격으로 생성된 데이터인지 식별할 수 있다면 효율적으로 해당 공격을 방어하는 분류 모델을 구축할 수 있다. 적대적 데이터를 학습에 활용하여 더 견고한 딥러닝 분류 모델을 구축하는 적대적 학습이라고 하는데, 만약 실제로 분류 모델을 공격하는 적대적 공격을 식별할 수 있다면 해당하는 적대적 공격 데이터에 대한 적대적 학습 단계를 진행하여 실제로 공격받은 적대적 공격에 대해 효율적으로 방어 체계를 구축할 수 있다. 추가로, 적대적 공격 분류 결과에 따라 공격 알고리즘 간의 차이점을 분석하고 공격 별 방어 전략을 설계하여 더욱 강력한 분류 모델을 구축할 수 있을 것이다[20].

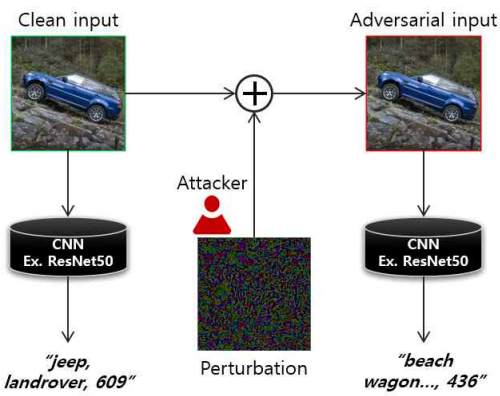

Fig. 1. example of adversarial attack

그러나 적대적 공격을 구별하는 연구는 거의 진행되지 않았다. 따라서 본 논문에서는 분류 모델이 분류하는 클래스에 따라 데이터를 처리하여 적대적 데이터를 탐지 및 분류하는 문제를 해결하는 것을 목표로 한다. 본 논문의 의의는 다음과 같다.

1. 다양한 적대적 공격으로 생성된 적대적 데이터를 탐지 및 분류하였다.
2. 실험을 통해 적대적 데이터에 대한 탐지 정확도와 정상 데이터에 대한 탐지 정확도가 기존 연구보다 더 높음을 보였다.
3. 기존 적대적 공격 분류 모델에서 분류하지 않았던 새로운 화이트 박스 공격에 대한 분류에 성공하였다.

본 논문의 구성은 다음과 같다. 2장에서는 적대적 공격 관련 연구와 적대적 공격 탐지 및 분류 관련 연구를 소개한다. 3장에서는 본 논문에서 제안하는 적대적 데이터 탐지 및 분류 모델에 관해 설명한다. 4장에서는 3장에서 제안한 모델의 성능을 보이고, 5장에서 결론과 함께 향후 연구 방향에 대해 제시한다.

II. 관련 연구

이 장에서는 적대적 공격 관련 연구를 소개하고, 이를 탐지하거나 분류하는 기존 연구에 관해 설명한다.

2.1 적대적 공격

적대적 데이터의 경우 공격자가 공격 대상 모델에 대해 가지는 정보량을 기준으로 화이트 박스 공격과

블랙박스 공격으로 나뉜다. 화이트 박스 공격의 경우 공격자가 공격 대상 모델에 대해 파라미터 정보를 포함한 모든 정보를 알고 있는 상태에서 적대적 데이터를 생성하는 공격인 반면 블랙박스의 경우 공격자가 대상 모델에 대한 정보 없이 적대적 데이터를 생성하는 공격을 말한다.

본 논문에서는 화이트 박스 공격에 대한 탐지 및 분류를 진행할 예정이므로 본 장에서는 다양한 화이트 박스 공격에 관해 설명한다.

2.1.1 Fast Gradient Sign Method (FGSM) Attack

딥러닝 모델의 손실 함수를 최대화하는 방향으로 입력 데이터를 업데이트하여 적대적 데이터를 생성하는 공격이다. Ian J. Good fellow 등이 제안했으며[3] 적대적 데이터를 생성하는 식은 (1)과 같다. 적대적 데이터 x'는 정상 입력 데이터 x에서 공격 대상 모델의 손실 함수 $\ell(\theta,x,y)$가 최대화되는 방향으로 ϵ만큼 갱신된다. $\epsilon(epsilon)$은 데이터의 섭동 정도를 표현하는 상수이다.

$$x' = x + \epsilon sign(\nabla_x \ell(\theta,x,y)) \quad (1)$$

2.1.2 Projected Gradient Descent (PGD) Attack

FGSM 공격을 단계별로 나누어 반복적으로 수행하여 적대적 데이터를 갱신시키는 공격 방법으로, A. Madry 등이 제안했다[4]. 단계마다 정상 데이터 x는 손실 함수 $\ell(\theta,x,y)$가 최대화되는 방향으로 움직이며, 갱신 후 특정 제약조건 범위 밖으로 벗어나는 경우 투영(projection)을 이용해 다시 범위 내로 이동시켜 수행하여 내부 최대화(inner maximization)를 수행한다. 식은 (2)와 같다.

$$x^{t+1} = proj\left\{x^t + \alpha sign(\nabla_x \ell(\theta,x,y)\right\} \quad (2)$$

2.1.3 Jacobian Saliency Map (JSMA) Attack

입력 데이터에서 딥러닝 모델이 분류 결과를 도출하는 데 중요하다고 판단한 픽셀을 식별하기 위해 그래디언트를 기반으로 saliency map을 생성, 입력 데이터의 픽셀별 중요도를 계산한다. 이후 적대적 데이터를 생성하기 위해 수정할 픽셀의 최소 수를 찾는 공격이다. N. Papernot 등이 제안하였다[5].

2.1.4 DeepFool Attack

DeepFool 공격 알고리즘은 적대적 데이터를 생성하는 데 필요한 최소한의 섭동을 기하학적으로 탐색하는 방법이다[6]. 딥러닝 모델의 결정 경계가 있을 때, 입력 데이터 x가 이를 넘어 적대적 데이터가 될 수 있도록 투영하여(projection) 적대적 데이터를 생성한다.

2.1.5 C&W Attack

C&W 공격의 경우 적대적 예제를 적대적 데이터 x'와 정상 데이터 x의 유사성을 측정하기 위해 L_0, L_2, L_∞ 놈(norm)을 활용한다[7]. C&W 공격의 목적함수는 식(3)과 같다. 정상 데이터 x와 섭동 δ을 추가하여 생성한 적대적 데이터 x'의 거리 매트릭(Distance Metric) $D(x, x+\delta)$을 최소화하는 동시에 제약조건 $f(x')$ 최소화하는 적대적 데이터를 생성한다. $f(x')$ 함수의 경우 생성된 적대적 데이터가 잘못 분류되도록 하는 함수로 여러 가지가 있으며, 식(4)는 $f(x')$ 의 예시이다.

$$\text{min}imize\ D(x, x+\delta) + c \cdot f(x+\delta) \quad (3)$$

$$such\ that\ x+\delta \in [0,1]^n$$

$$f(x') = -loss_{F,t}(x') + 1 \quad (4)$$

2.1.6 Auto-PGD Attack

F. Croce 등이 제안한 공격 방법으로[8], PGD와 달리 그래디언트 단계에 모멘텀 항을 추가하고, 전체 공격 예산(budget)에 따라 반복에 걸쳐 단계 크기(step size)를 조정한 후 최상의 지점에서 다시 데이터 갱신을 진행한다.

2.2 적대적 공격 탐지 및 분류 연구

적대적 공격 탐지 연구의 경우 크게 적대적 공격 알고리즘에 의해 생성된 적대적 데이터를 이용하는지도 탐지 기법과 적대적 공격에 대한 사전 지식이 없이 적대적 데이터를 탐지하는 비지도 탐지 기법으로 나눌 수 있다[13].

2.2.1 적대적 공격 지도 탐지

X. Li 등은 공격 대상 분류 모델의 은닉 계층별로 SVM 분류기를 두어 적대적 데이터를 탐지하는 방안을 제시하였다[14]. 이때, SVM 분류기의 입력은 분류 모델의 계층별 출력에 대한 Principle component Analysis(PCA)이다.

HF. Eniser 등은 정상 데이터와 적대적 데이터 사이의 뉴런 활성화 값 차이를 입력으로 사용한 이진 분류 모델 "Randomized Adversarial Input Detection (RAID)"을 제안하여 적대적 데이터를 탐지했다[15].

S. Pertigkiozogolu 등은 공격 대상 모델의 첫 번째 컨볼루션 계층 출력의 히스토그램을 입력으로 사용하는 이진 SVM 분류기를 이용해 적대적 데이터를 탐지하는 방안을 제시했다[16].

J. Lu 등은 적대적 데이터가 공격 대상 분류 모델의 후기(late stage) Relu 함수(Rectified Linear Unit)에서 정상 데이터와 다른 활성화 값 패턴에 의해 생성된다는 가정을 기반으로 마지막 Relu 함수의 활성화 값을 이용해 구축한 이진 RBF-SVM 분류기로 적대적 데이터를 탐지하는 SafetyNet 모델을 제안하였다[17].

2.2.2 적대적 공격 비지도 탐지

F. Carrara 등은 공격 대상 딥러닝 모델의 예측 클래스에 대해 점수를 매기고, 해당 점수가 지정한 임계값보다 클 때 적대적 데이터로 탐지하는 모델을 제시했다[18]. 해당 점수는 대상 딥러닝 모델의 계층 중 하나의 출력을 이용한 k-NN 분류기를 구성하여 산출하였다.

W. Xu 등은 특징 압축(Feature Squeezing)을 통해 입력 데이터를 압축한 데이터를 생성해 적대적 데이터를 탐지하는 방안을 제안한다[19]. 기존 입력 데이터에 대한 모델의 예측과 압축된 입력 데이터에 대한 모델의 예측 차이가 임계값보다 큰 경우 적대적 데이터로 식별한다.

2.2.3 적대적 공격 분류 연구

적대적 공격을 탐지하는 연구가 많이 진행된 반면에 적대적 공격을 분류하는 연구는 거의 진행되지 않았다. DeClaw 모델은 N. Manohar-Alers 등이 제안한 모델로[20], 적대적 데이터를 탐지하는 동시에 데이터를 생성하는 데 사용한 적대적 공격 알고리즘까지 분류하는 모델이다. DeClaw 모델은 적대적 공격 대상 모델의 은닉층(hidden Layer)의 출력값을 활용하며, 탐지 및 분류를 위해 추가적인 딥러닝 분류 모델을 이용한다. Declaw 모델의 주요 아이디어는 정상 데이터와 다양한 적대적 공격 알고리즘으로 생성된 적대적 데이터 간의 통계적인 차이를 적대적 공격 분류 모델의 입력 데이터로 사용한다는 점이다.

III. 제안 모델

이 장에서는 본 논문에서 제안하는 적대적 데이터를 탐지 및 분류하는 모델의 개요 및 구성 요소를 소개하고, 모델의 각 요소에 대하여 자세히 설명한다.

3.1 개요

본 논문에서 제안하는 적대적 공격 탐지 및 분류 모델은 특징 선택, 특징 추출, 적대적 공격 탐지 단계로 구성되며, Fig 2.와 같다.

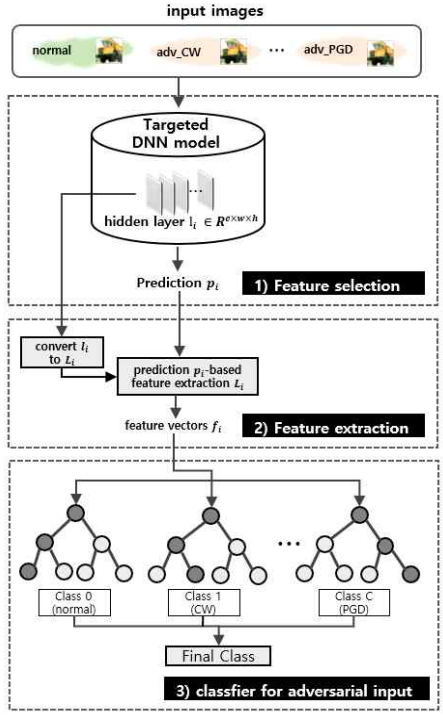

Fig. 2. Overview of the Proposed model

첫 번째, 특징(Feature) 선택 단계는 적대적 데이터 탐지 및 분류에 사용할 공격 대상 딥러닝 모델의 내부의 값을 가져오는 단계이다. 두 번째, 특징 추출 단계는 공격 대상 딥러닝 모델에서 가져온 특징으로부터 변환 및 차원 감소를 통해 새로운 특징 추출을 하는 단계이다. 세 번째, 분류기 학습 단계는 추출한 특징을 입력으로 하는 적대적 데이터 탐지 및 분류 모델을 구성하는 단계이다. 분류 모델의 경우 랜덤 포레스트(Random Forest) 모델을 활용한다.

특징 선택, 특징 추출, 분류기 학습 단계에 대하여 3.2, 3.3, 3.4 절에서 자세히 설명한다.

3.2 Feature selection

특징 선택 단계의 경우 특징 추출에 활용할 두 가지 데이터를 뽑아내는 단계이다. 첫 번째는 하나의 입력 데이터 D_i에 대해 딥러닝 모델의 은닉층 중 특정 계층의 출력 $l_i \in R^{c \times w \times h}$(이때 c는 채널(channel), w는 너비(width), h는 높이(height)를 의미한다)이며, 두 번째는 데이터 D_i에 대한 공격 대상 모델의 예측 클래스 P_i이다. 특징 추출에서는 이 두 가지 정보를 이용해 적대적 데이터 분류 모델에 들어갈 특징을 추출한다.

3.3 Feature extraction

특징 추출의 경우 특징 선택 단계에서 가져온 l_i와 P_i를 이용하여 적대적 데이터 분류 모델에 들어갈 입력 데이터를 추출하는 단계이다. 크게 세 가지로 나눌 수 있으며, 이는 아래와 같다.

1. 특정 은닉층의 출력 l_i의 차원을 줄이는 단계
2. 특징을 추출하는 데 기반이 되는 normative clean vector N을 생성하는 단계
3. 특징 추출 단계

3.3.1 dimension reduction

특정 은닉층의 출력 l_i를 이용하기 전 효율적인 계산을 위해 차원 감소를 진행한다. l_i는 채널별 평균값 L_i로 변경된다. 식은 (5)와 같다.

$$L_{i,c} = \frac{\sum l_{i,c}}{len(l_i)} \tag{5}$$

특정 은닉층의 출력 l_i의 크기가 $[1, c, w, h]$라면 l_i는 c개의 채널과 $w \times h$의 값(value)으로 구성된다. 이 경우 식 (1)에서 $len(l_i) = w \times h$이 되어 차원 감소를 진행한 L_i의 크기는 $[1, c]$가 된다.

차원 감소를 진행한 데이터 L_i와 적대적 공격 대상 모델이 예측하는 클래스 P_i를 이용해 특징 추출을 진행한다.

3.3.2 normative clean vector

특징 추출 단계에서 정상 데이터와 적대적 데이터 사이의 특징 차이를 극대화하기 위해, 기존 연구[20]와 같이 n개의 정상 데이터를 무작위로 선정하여 활용한다. 선정된 n개의 정상 데이터 셋에 대한 L_i값을 이용하여 실제 데이터 추출에 사용하는데 기준이 되는 normative clean vector N을 구한다. N의 경우 공격 대상 딥러닝 모델이 실제로 분류하는 클래스의 수만큼 생성되는 벡터로, 식 (6) 과같이 표현할 수 있다.

normative clean vector N을 구하는 식은 식 (7)과 같으며, 분류 클래스별 L_i값의 평균을 의미한다. 식 (7)에서 $n^{target = C_i}$는 n개의 데이터 셋 중 클래스가 C_i인 데이터의 개수이다.

$$N = [N^0, N^1 \cdots, N^c], \quad c = \#\ classes \tag{6}$$

$$N^{C_i} = \frac{\sum L^{target = C_i}}{n^{target = C_i}} \tag{7}$$

3.3.3 feature extraction

제안한 모델의 특징 추출은 Table 1.과 같다. 입력 데이터 D_i에 대한 L_i 값과 정상 데이터 간의 차이를 식별하기 위해 normative clean vector N과의 차이 T_i와 상대적인 차이 rel_T_i를 계산한다.

N의 경우 입력 데이터 D_i에 대해 대상 딥러닝 모델이 예측하는 클래스 값 P_i에 해당하는 대푯값 N^{P_i}를 이용한다. T_i와 rel_T_i를 적대적 데이터 탐지 및 분류 모델의 입력 데이터로 사용할 수 있도록 Linear Discriminant Analysis(LDA) 기법을

이용해 차원을 축소한다. LDA는 집단 간 평균 차이는 극대화하면서, 각 집단 내부의 분산은 최대한 작게하도록 고유값(eigen value)을 분해, 데이터의 차원을 감소시키는 기법이다. LDA의 경우 지도학습으로 실제 분류 결과를 필요로 하므로, T_i와 rel_T_i 에서 적대적 데이터 분류 모델의 학습 단계에 사용하는 데이터로 LDA를 학습시킨 후 특징을 추출하였다.

학습된 LDA를 이용해 추출하는 성분 개수의 경우, 추출된 성분에 의해 최소한 99%의 학습 데이터 분산이 표현될수 있도록 추출된 고유 벡터(eigen vector)에 따라 누적되는 고유값(eigen value)의 백분율이 99%될 때까지 추출하였다.

추가로, T_i와 rel_T_i 값의 통계적인 차이를 식별하기 위해 해당 값의 평균, 표준편차 및 합계를 특징으로 사용하였다.

Table 1. Pseudo code for extraction features

Pseudo Code
Inputs N: normative clean vector L: hidden layer output P: prediction for targeted model
outputs - F: extracted features
1: 2: //create T, rel_T 3: for $i \in 1 \cdots len(N)$ do 4: $T_i = L_i - N^{P_i}$ 5: $rel\_T_i = (L_i - N^{P_i}) / N^{P_i}$ 6: end for 7: 8: //devided into dataset train and test 9: $(T\_test,\ T\_train) \leftarrow T$ 10: $(rel\_T\_test,\ rel\_T\_train) \leftarrow rel\_T$ 11: 12: //extract feature for T 13: $LDA1 = LearningLDA(T\_train)$ 14: $F_1 = LDA1(T)$ 15: //extract feature for rel_T 16: $LDA2 = LearningLDA(rel\_T\_train)$ 17: $F_2 = LDA2(rel\_T)$ 18: 19: //statistical feature 20: $F_3 = mean(T), std(T), \sum(T)$ 21: $F_4 = mean(rel\_t), std(rel\_t), \sum(rel\_t)$ 22: return F_1, F_2, F_3, F_4

3.4 Classifier for adversarial input

추출된 특징을(F_1, F_2, F_3, F_4) 입력으로 하는 분류 모델의 경우 간단하고 빠른 다중 클래스 알고리즘인 랜덤 포레스트 모델을 활용한다. 랜덤 포레스트의 경우 여러 개의 의사 결정 트리를 통해 예측된 결과를 앙상블(Ensemble)하여 최종 예측값을 출력한다.

제안하는 모델에서 랜덤 포레스트의 입력은 제2절, 제3절에서 소개한 방법으로 추출한 특징(F_1, F_2, F_3, F_4)이며, 출력은 입력 공격 대상 모델의 입력 데이터가 정상 데이터인지, 적대적 공격인지, 적대적 공격이라면 어떠한 공격 알고리즘에 의해 생성되었는지 분류한다. 정상 데이터의 경우 하나의 분류로, 적대적 공격의 경우 적대적 공격 알고리즘에 따라 각기 다른 종류로 분류된다.

IV. 실험 결과

이 장에서는 다음과 같이 구성된다. 4.1장에서 본 논문에서 제안하는 적대적 데이터 탐지 및 분류 모델의 성능을 보이기 위해 사용한 데이터 셋에 대해 설명하고, 4.2장에서 실험에 사용한 공격 대상 분류 모델 등 실험 구성에 관해 설명한다. 4.3장에서는 실험 결과를 제시한다. 실험을 진행한 환경은 Table 2.와 같다.

Table 2. Experiments environments

Experiments	Version
OS	Ubuntu 18.04.5 LTS
CPU	Intel Core i7-10700K
GPU	GeForce GTX 1080 Ti
Pytorch	1.9.1+cu102

4.1 데이터 셋

실험에는 CIFAR-10 데이터 셋을 사용하였으며, 60,000개의 정상 데이터와 공격 별 10,000개의 적대적 데이터를 활용하여 실험을 진행하였다. 실험에 사용된 적대적 데이터 공격의 종류는 총 7개이며, Table 3.과 같다.

적대적 데이터 탐지 및 분류 성능 비교를 위해 기존 연구[20]에서 활용한 적대적 공격에 대해서 동일한 공격 생성 오픈소스(IBM Adversarial

Robustness Tool)와 파라미터를 이용해 적대적 데이터를 생성하였고, 새로운 적대적 공격 데이터의 경우 동일한 공격 생성 오픈소스와 파라미터를 기본 값으로 설정하여 생성하였다.
정상 데이터와 생성된 적대적 데이터 총 130,000개를 70%, 30%로 나누어 70%는 모델을 학습시키는 데이터로, 30%는 모델의 성능을 평가하는 데 사용하였다. Table 4.에서 실제로 학습에 사용하는 데이터와 성능 평가에 사용하는 데이터의 수를 적대적 데이터와 정상 입력으로 나누어 정리하였다.

Table 3. white-box attack dataset for adversarial attack detection and classification

attack name	attack description	Ref
FGSM	Fast Gradient Sign Method	[3]
PGD	Projected Grad	[4]
JSMA	Jacobian Saliency Map Attack	[5]
DF	DeepFool	[6]
CW	Carlini & Wagner(C&W) L2	[7]
APGD_CE	Auto-PGD with cross entropy	[8]
APGD_DLR	Auto-PGD with logits ratio	[8]

Table 4. dataset for adversarial attack detection and classification

dataset	# adversarial	# clean	# total
train	49,000	42,000	91,000
test	21,000	18,000	39,000

4.2 실험 구성

CIFAR-10 데이터에 대한 공격 대상 분류 모델의 경우, 실험 결과 비교를 위해 기존 연구[20]와 동일한 Wide-Residual Network[21] 모델을 활용하였다. 해당 분류 모델의 경우 CIFAR-10 정상 데이터에 대해 95.49%의 정확도를 보이며, 본 논문에서 생성한 적대적 데이터에 대해 평균 6%의 정확도를 보인다.

적대적 데이터를 탐지하기 위해 사용하는 특정 은닉층의 값 l_i은 적대적 공격 대상 모델의 가장 마지막 배치 정규화(Batch normalization)의 입력으로 들어가는 값을 활용하였다. 해당 값을 사용하는 이유는 두 가지이다. 첫 번째, 해당 값이 가장 마지막에 위치하여 고급 특징을 뽑아내고 있으며 배치 정규화에 들어가기 전의 값이기 때문에 배치 별로 정규화되지 않은 실제 값을 활용할 수 있다는 점이다. 두 번째, 동일하게 마지막 배치 정규화의 입력값을 이용하여 적대적 공격 탐지 및 분류 문제를 해결한 기존 연구[20]가 있기 때문이다. 실험에서 사용한 공격 대상 분류 모델 Wide-Residual Network의 경우 입력 데이터 D_i에 대한 마지막 배치 정규화 이전 계층의 출력값 l_i은 [640, 8, 8] 크기의 텐서이며, 특징 추출에 해당 텐서를 활용하였다.

LDA를 이용해 추출된 성분 개수의 경우, 추출된 성분에 의해 최소한 99%의 학습 데이터의 분산이 표현될 때까지 추출한 결과 normative clean vector N과의 차이 T_i에 대한 LDA 모델의 경우 5개, 상대적인 차이 rel_T_i에 대한 LDA 모델의 경우 6개의 성분을 추출하여 하나의 입력 데이터 D_i에 대해 총 17개의 특징을 추출하여 적대적 데이터를 탐지 및 분류하는 데 사용하였다.

적대적 데이터를 탐지 및 분류하기 위해 사용하는 랜덤 포레스트 분류 모델은 scikit-learn 라이브러리를 이용하여 구축하였고, 의사결정트리 개수는 100으로 설정하였다.

4.3 실험 결과

실험 결과는 적대적 데이터를 탐지하는 성능과 적대적 데이터를 분류하는 성능으로 나누어 결과를 제시한다.

4.3.1 적대적 데이터 탐지 성능

분류 모델의 적대적 데이터를 탐지하는 성능은 적대적 데이터에 대한 정확도(Accuracy)와 정상 데이터에 대한 정확도로 나누어 계산한다. 본 논문에서 제안한 모델의 적대적 데이터 탐지 성능은 Table 5.와 같다.

N. Manohar-Alers 등이 제안한 기존 연구[20]의 경우 적대적 데이터에 대한 정확도는 96%, 정상 데이터에 대한 정확도는 93%를 보였다. 제안하는 모델의 경우 적대적 데이터는 기존 연구와 비슷

Table. 5. Evaluation of proposed Model for detection

	Accuracy on adversarial data (%)	Accuracy on clean data (%)
Proposed Model	96.82	96.02

한 성능을, 정상 데이터에 대한 정확도는 96.02%로 기존 연구보다 3.02% 높은 정확도를 보인다. Fig 3.는 적대적 데이터 탐지 결과에 대한 컨퓨전 매트릭스(confusion matrix)의 값을 정규화하여 표현하였다.

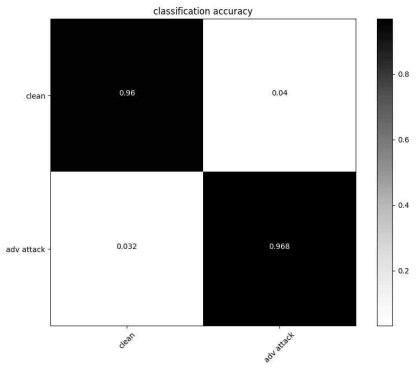

Fig. 3. Normalized confusion matrix for the detection result

4.3.2 적대적 데이터 분류 성능

제안하는 모델의 적대적 데이터 분류 성능의 경우 Fig 4.와 같다. Fig 4.는 적대적 데이터 분류 결과에 대한 컨퓨전 매트릭스의 값을 정규화하여 표현하였다.

기존 연구의 경우 적대적 데이터 분류 성능을 높이기 위해 공격 간의 False Positive Rate(FPR)와 False Negative Rate(FNR)을 이용한 클러스터링 모델을 통해 공격을 클러스터링한다[20]. 클러스터링 결과 CW 공격, DeepFool 공격, Auto-PGD(logits ratio)를 하나의 공격 유형으로 설정한다.

본 논문에서는 Auto-PGD(logits ratio)에 대해서는 0.965의 컨퓨전 매트릭스 값을 가져 기존 연구에서는 하나의 공격 유형으로 설정한 Auto-PGD(logits ratio)를 분류하는 데 준수한

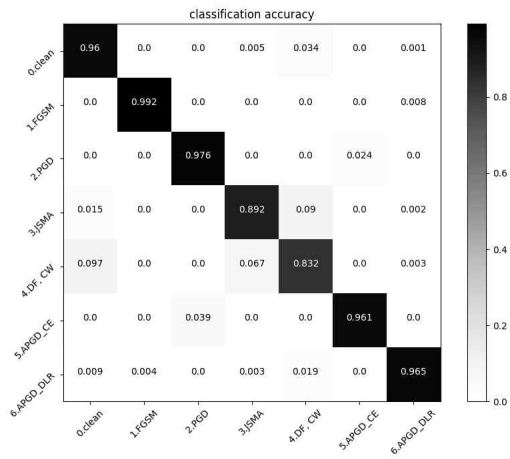

Fig 4. Normalized confusion matrix for the classification result

성능을 보인다.

추가로, 본 논문에서 제안한 모델은 기존 연구에서 다루지 않는 3.Jacobian Saliency Map (JSMA) Attack 공격에 대해 0.892의 컨퓨전 매트릭스 값을 보이고 있다. 즉 제안한 모델이 새로운 공격을 효과적으로 분류하고 있음을 확인하였다.

또한 기존 연구의 경우 하나의 입력 데이터 D_i에 대하여 176개의 특징을 추출하여 적대적 탐지 및 분류를 진행하였지만, 제안한 논문의 경우 훨씬 더 적은 새로운 17개의 특징을 이용하여 기존 연구와 비슷한 성능을 보인다. 본 실험에서 추출된 17개의 특징은 Table. 6 와 같다. 각 특징에 대한 정보는 Table. 1에서 정의한 내용과 동일하다.

Table. 6. Extracted Features for the Experiments

Feature name	description	# feature
F_1	$LDA1(T)$	5
F_2	$LDA2(rel\_T)$	6
F_3	$mean(T)$, $std(T)$, $\sum(T)$	3
F_4	$mean(rel\_t)$, $std(rel\_t)$, $\sum(rel\_t)$	3

V. 결 론

본 논문에서는 적대적 공격 대상 딥러닝 모델이 예측하는 클래스에 따라 분류에 사용하는 데이터를 생성하여 적대적 공격을 탐지 및 분류하는 방안을 제시하였다. 실험을 통해 기존 연구보다 훨씬 더 적은 특징을 이용해 비슷한 분류 성능을 보였고, 새로운 공격에 대한 분류에 성공하였다. 또한, 정상 데이터에 대한 정확도에 대해서는 3.02% 향상된 성능을 보였다.

향후 연구를 통해서 다른 적대적 공격으로 생성한 데이터를 활용해 추가적인 적대적 공격 종류에 대한 분류 모델을 구축하는 연구가 가능하고, 제안한 특징 이외에 추가적인 특징을 추출하여 적대적 공격을 더욱 정밀하게 분류할 수 있는 모델을 구축할 수 있을 것이다.

References

[1] H. Caesar, V. Bankiti and AH. Lang, "nuScenes: A multimodal dataset for autonomous driving," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11621-11631, Jun. 2020.

[2] G. Litjens, T. Kooi and B.E. Bejnordi, "A survey on deep learning in medical image analysis," Medical Image Analysis, vol. 42, pp. 60-88, Jul. 2017.

[3] Ian J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and Harnessing Adversarial Examples," arXiv preprint arXiv: 1412.6572v3, Mar. 2015.

[4] A. Madry, A. Makelov and L. Schmidt, "Towards Deep Learning Models Resistant to Adversarial Attacks," arXiv preprint arXiv: 1706.06083v4, Sep. 2019.

[5] N. Papernot, P. McDaniel and S. Jha, "The Limitations of Deep Learning in Adversarial Settings," IEEE European Symposium on Security and Privacy (EuroS&P), pp. 372-387, Mar. 2016.

[6] S.M. Moosavi-Dezfooli, A. Fawzi and P. Frossard, "DeepFool: a simple and accurate method to fool deep neural networks," Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2574-2582, Jun. 2016.

[7] N. Carlini and D. Wagner, "Towards Evaluating the Robustness of Neural Networks," IEEE Symposium on Security and Privacy. pp. 39-57, May. 2017.

[8] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," Proceedings of the 37th International Conference on Machine Learning (PMLR), vol.119, pp. 2206-2216, Jul. 2020.

[9] F. Tramèr, A. Kurakin and N. Papernot, "Ensemble Adversarial Training: Attacks and Defenses," arXiv preprint arXiv: 1705.07204v5, Apr. 2020.

[10] A. Shafahi, M. Najibi and A. Ghiasi, "Adversarial Training for Free!," Proceedings of the 33rd International Conference on Neural Information Processing Systems, pp. 3358-3369, Dec. 2019.

[11] A. Prakash, N. Moran and S. Garber, "Deflecting adversarial attacks with pixel deflection," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8571-8580, Jun. 2018.

[12] N. Papernot, P. McDaniel and X. Wu, "Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks," IEEE Symposium on Security and Privacy, pp. 582-597, May. 2016.

[13] A. Aldahdooh, W. Hamidouche and S

A. Fezza, "Adversarial Example Detection for DNN Models: A Review," arXiv preprint arXiv: 2105.00203v2, Sep. 2021.
[14] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 5764 - 5772, Oct. 2017.
[15] HF. Eniser, M. Christakis and V. Wüstholz, "RAID: Randomized adversarial-input detection for neural networks," arXiv preprint arXiv: 2002.02776v1, Feb. 2020.
[16] S. Pertigkiozoglou and P. Maragos, "Detecting Adversarial Examples in Convolutional Neural Networks," arXiv preprint arXiv: 1812.03303v1, Dec. 2018.
[17] J. Lu, T. Issaranon and D. Forsyth, "SafetyNet: Detecting and Rejecting Adversarial Examples Robustly," Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 446-454, Oct. 2017.
[18] F. Carrara, F. Falchi and R. Caldelli, "Detecting adversarial example attacks to deep neural networks," Proceedings of the 15th International Workshop on Content-Based Multimedia Indexing, pp. 1 - 7, Jun. 2017.
[19] W. Xu, D. Evans and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," arXiv preprint arXiv: 1704.01155v2, Dec. 2017.
[20] N. Manohar-Alers, R. Feng and S. Singh, "Using Anomaly Feature Vectors for Detecting, Classifying and Warning of Outlier Adversarial Examples," arXiv preprint arXiv: 2107.00561v1, Jul. 2021.
[21] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," arXiv preprint arXiv: 1605.07146v4, Jun. 2017.

〈저자소개〉

고 은 나 래 (Eun-na-rae Ko) 학생회원
2019년 2월: 광운대학교 컴퓨터공학과 학사 졸업
2020년 3월: 고려대학교 정보보호학과 석사 과정
〈관심분야〉 정보보호, 시스템 보안

문 종 섭 (Jongsub Moon) 종신회원
1981년 2월: 서울대학교 계산통계학과 학사
1983년 2월: 서울대학교 계산통계학과 석사
1991년 2월: Illinois Institute of Technology 전산학과 박사
1993년 3월~현재: 고려대학교 전자 및 정보공학부 교수
2001년 2월~현재: 고려대학교 정보보호대학원 겸임교수
〈관심분야〉 정보보호, 운영체제, 침입탐지