악성코드 이미지 분류를 위한 CNN 모델 성능 비교

강채희*, 오은비*, 이승언*, 이현경*, 김성욱*
*서울여자대학교 정보보호학과
ellin0817@swu.ac.kr, fsrfsr4033@swu.ac.kr, oktmd6160@swu.ac.kr, gbrb9916@swu.ac.kr,
kim.sungwook@swu.ac.kr

Comparison Study of the Performance of CNN Models for malicious code image classification

Chae-Hee Kang*, Eun-Bi Oh*, Seung-Eon Lee*, Hyun-Kyung Lee*, Sung-Wook Kim*
*Dept. of Information Security, Seoul Women's University

요 약

최근 IT 산업의 지속적인 발전으로 사용자들을 위협하는 악성코드, 피싱, 랜섬웨어와 같은 사이버 공격 또한 계속해서 발전하고 더 지능화되고 있으며 변종 악성코드도 기하급수적으로 늘어나고 있다. 지금까지의 시그니처 패턴 기반의 탐지법으로는 이러한 방대한 양의 알려지지 않은 악성코드를 탐지할 수 없다. 따라서 CNN(Convolutional Neural Network)을 활용하여 악성코드를 탐지하는 기법들이 제안되고 있다. 이에 본 논문에서는 CNN 모델 중 낮은 인식 오류율을 지닌 모델을 선정하여 정확도(Accuracy)와 F1-score 평가 지표를 통해 비교하고자 한다. 두 가지의 악성코드 이미지화 방법을 사용하였으며, 2015 년 이후 ILSVRC 에서 우승을 차지한 모델들과, 추가로 2019 년에 발표된 EfficientNet 을 사용하여 악성코드 이미지를 분류하였다. 그 결과 2 바이트를 한 쌍의 좌표로 변환하여 생성한 256 * 256 크기의 악성코드 이미지를 ResNet-152 모델을 이용해 분류하는 것이 우수한 성능을 보임을 실험적으로 확인하였다..

1. 서론

최근 it 산업의 지속적인 발전으로 인터넷과 여러 분야들이 접목된 기술들이 늘어나면서 컴퓨팅 기술의 수준 또한 높아지고 있다. 이로 인해 인터넷 사용에 있어 편의성이 높아지고 삶에 질도 함께 올라가는 긍정적인 상황에 놓였지만 그만큼 사용자들을 위협하는 악성코드, 피싱, 랜섬웨어와 같은 사이버 공격 또한 계속해서 발전하고 더 지능화되고 있다. 최근에는 악성코드를 자동으로 생성하는 자동화 툴과 봇넷까지 개발되었으며, 기존의 코드에서 일부를 수정한 변종 악성코드도 기하급수적으로 늘어나고 있다[1].

지금까지는 전통적인 악성코드 탐지 기법인 시그니처 패턴 기반의 탐지법을 이용해 알려진 악성코드(Known Malware)를 탐지하는 방식을 사용하였다. 하지만 기존의 탐지 기법으로는 방대한 양의 알려지지 않은 악성코드(Unknown Malware), 즉 변종 악성코드를 탐지할 수 없다. 이러한 한계를 극복하기 위해 최근에는 인공지능(AI)을 통한 악성코드 탐지 기법에 대한 연구가 활발히 진행되고 있다. 인공지능은 악성코드를 빠르게 분석해 정보를 업데이트하여 더 많은 양의 악성코드를 비교적 짧은 시간 안에 분류할 수 있게 해준다. 인공지능을 이용한 악성코드 분류 방법 중 가장 대표적인 것은 악성코드를 이미지화하여 CNN(합성곱 신경망, Convolutional Neural Network)으로 분류하는 것이다. CNN 을 활용한 다양한 악성코드 분류 연구가 있다. 악성코드를 이미지로 인코딩하는 방법도 다양하다. 그런데 이들에 대해 실증적으로 탐지율을 비교 분석한 연구는 알려져 있지 않다. 따라서 이 논문에서는 악성코드 연구에서 많이 쓰이는 Microsoft Malware Classification Challenge dataset[2] 대해 주요한 CNN 및 이미지 인코딩 기법의 정확도를 실증적으로 비교/분석한 결과를 제시한다.

2 장에서는 악성코드 이미지화 방법에 대해 기술하고, 3 장에서는 본 연구에서 비교한 CNN 모델에 대해 기술하였다. 4 장에서는 실험 환경과 계획을, 5 장에서는 평가 지표에 따른 실험 결과를 분석하였다. 6 장에서는 결론에 대해 기술하였다.

2. 시각화 분류 연구

2.1 데이터 셋

CNN 을 이용한 악성코드 탐지 및 분류를 위해 사용한 악성코드 데이터 셋은 Microsoft 에서 Kaggle 을 통해 발표한 Microsoft Malware Classification Challenge dataset 이다. 데이터 셋은 총 10,868 개의 악성코드 샘플로 구성되어 있으며, 9 개의 패밀리로 분류된다. 악성코드 패밀리 종류에는 Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, Gatak 가

있다.

<표 1> 악성코드 패밀리별 악성코드 수

악성코드 종류	악성코드 개수
Ramnit	1,541
Lollipop	2,478
Kelihos_ver3	2,942
Vundo	475
Simda	42
Tracur	751
Kelihos_ver1	398
Obfuscator.ACY	1,228
Gatak	1,013

2.2 시각화 분류 방법

2.2.1 1 바이트를 grayscale 이미지의 한 픽셀로 변환하는 방법

악성코드의 바이너리 파일을 부호가 없는 8 비트 정수 벡터로 읽어 들인다. 각 바이트는 이미지의 한 픽셀로 0 ~ 255 (0:검정, 255:흰색) 범위의 값을 갖게 되고, 이를 이용하여 grayscale 이미지를 만든다[3].

합성곱 신경망을 이용한 악성코드 탐지 및 분류 실험은 256 * 256 의 이미지를 입력으로 사용하기 때문에 악성코드 이미지의 크기를 256 * 256 크기로 조정하였다[4].

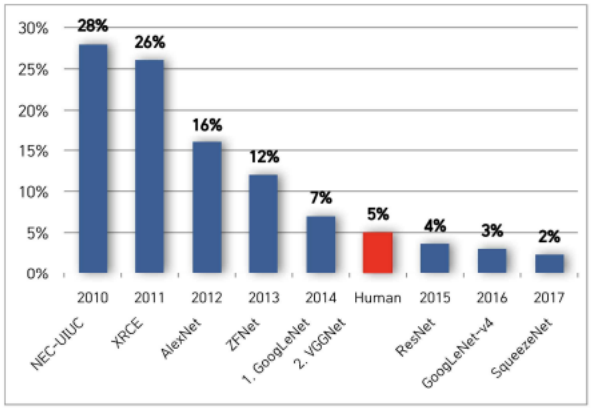
2.2.2 2 바이트를 한 쌍의 좌표로 변환하는 방법

악성코드의 바이너리 파일을 2 바이트씩 읽어 들여 각 1 바이트를 x 와 y 에 대응시켜 한 쌍의 (x, y) 좌표로 변환시킨다. 256 * 256 크기의 배열에서 좌표 값과 일치하는 배열의 값을 1 씩 증가시킨다. 악성코드의 바이너리 파일을 다 읽을 때까지 1 바이트씩 이동하며 위 방법을 반복한다. 이를 통해 얻은 256 * 256 크기의 배열은 비트맵 이미지로 변환하여 악성코드 이미지를 생성할 수 있다[5].

3. CNN 모델

3.1 선정 배경

CNN 은 주로 이미지 인식에 사용되는 뉴럴 네트워크의 한 종류이다. 딥러닝 기반인 CNN 모델이 이미지 인식에 사용되면서 인식 오류율은 확실하게 낮아졌다. 2012 년에 들어서면서 AlexNet 이라고 불리는 CNN 모델을 이용한 이미지 인식 방법이 인식 오류율을 26%에서 16%로 낮추며 이미지 인식 공모전 ILSVRC (ImageNet Large Scale Visual Recognition Competition)에서 우승을 하였고, 이후 본격적으로 CNN 모델을 이용한 이미지 인식 방법이 주류가 되었다. GoogLeNet 의 Inception module, ResNet 의 Skip-connection 등 일반적인 CNN 구조보다 더 효율적으로 특징들을 다룰 수 있는 다양한 구조에 대한 연구가 있었고, 2015 년 이후부터 CNN 모델의 인식 오류율은 3.6%로 낮아지며 사람의 정확도라고 알려진 인식 오류율 5%를 추월해 인간을 뛰어넘는 성능을 낼 수 있게 되었다. 따라서 본 논문에서는 2015 년 이후 ILSVRC 에서 우승을 차지한 인식 오류율이 5% 미만인 CNN 모델들을 실험에 적용하였다. 추가로 ResNet, Inception, SqueezeNet 과 대비하여 확연히 개선된 성능을 보여주는 CNN 모델인 EfficientNet 을 실험에 적용하였다.



(그림 1) ILSVRC 우승 알고리즘 오류율

3.2 CNN 모델

3.2.1 ResNet

ResNet 은 2015 ILSVRC 에서 1 위를 차지한 모델이다. 모델의 layer 가 너무 깊어지면 오히려 성능이 떨어지는 현상인 vanishing/exploding gradient (기울기 소실/폭발) 문제가 발생한다. ResNet 은 skip connection 을 이용한 residual learning(잔차 학습 방법)을 통해 layer 가 깊어짐에 따른 vanishing/exploding gradient 문제를 해결한다. ResNet 은 잔차 학습 방법을 사용하기 위한 빌딩블록을 여러 개 쌓는 구조로 이루어져 있다. 또 shortcut 이 추가되어 입력이 출력에 그대로 연결되어서 파라미터 수에 영향이 없다. 이러한 구조로 인해 더 쉬운 최적화와 깊은 네트워크에서의 정확도 향상이 가능하다. 본 연구에서는 ResNet-152 구조를 사용하였다[6].

3.2.2 Inception-v4

Inception-v4 는 2016 ILSVRC 에서 1 위를 차지한 모델이다. 모델이 깊어질수록 성능이 높아지지만 연산량과 파라미터 수가 많아져 이에 따른 총 학습시간, 연산 속도 등의 문제들이 발생한다. Inception 은 이러한 한계를 극복하기 위해 제안된 모델이다. Inception-v4 는 다양한 형태의 convolution 을 통해 추출된 결과를 하나로 결합하는 Inception module 을 사용하며, 필터의 개수를 줄이도록 정의된 1*1 크기의 convolution 구조를 최대한으로 활용하여 연산량과 파라미터 수를 줄여 위의 문제들을 해결한다. 이러한 구조로 인해 모델을 가볍게 만들면서도, 모델을 더 깊게 구현해 성능을 높이는 것이 가능하다. 본 연구에서는 GoogLeNet 의 Inception-v4 구조를 사용하였다[7].

3.2.3 SqueezeNet

SqueezeNet 은 2017 ILSVRC 에서 1 위를 차지한 모델이다. 연산량이 적기 때문에 학습이 빠르며, 실시간으로 정보를 전송해야 하는 업무에 적용 가능하다. 해당 모델에는 3 가지 전략이 적용되는데 우선, 3x3 filter 를 1x1 filter 로 대체하여 연산량을 9 배 낮춘다. 또한,

3x3 filter 로 입력되는 입력 채널의 수를 감소시켜 연산량을 감소시킨다. 마지막으로 pooling layer 를 최대한 늦게 배치하는 방법을 통해 정확도를 높이도록 한다[8][9].

3.2.4 EfficientNet

기존의 CNN 모델은 모델의 깊이, 너비, 입력 이미지의 크기 조절을 통해 모델의 정확도를 높이고자 했으나 이를 수동으로 조절했기에 최적의 성능과 효율을 얻지 못하였다. 이에 EfficientNet 은 Compound Scaling 방법을 제안하여 모델의 깊이, 너비, 입력 이미지의 크기를 효율적으로 조절하도록 한다. 본 연구에서는 EfficientNet-B0 구조를 사용하였다[10].

4. 실험 계획

4.1 실험 개요

전체 데이터(original data)의 65%를 학습(training)을 위해 사용하고, 전체 데이터에서 랜덤으로 추출한 15%를 학습 iteration 마다 validation 으로 사용, 나머지(testing) 20%를 성능 평가를 위해 사용한다.

4.2 실험 환경

각 모델은 Google Colab Pro+에서 훈련을 진행하였다.

<표 2> 가상머신 환경

Name	Version
Operation System	Windows 10 (64-bit)
RAM	4GB

<표 3> Google Colab Pro+ 환경

Name	Version
CPU	Intel(R) Xeon(R) CPU @ 2.00GHz (Dual-Core)
GPU	Nvidia Tesla V100-SXM2
GPU Memory	8GB

5. 실험 결과 및 분석

5.1 분류 모델의 성능 평가 지표

각 CNN 모델별 분류 성능 평가를 위해 True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN)을 활용하여 2 가지 성능지표인 정확도(Accuracy)와 F1_score 를 적용하였다.

TP 는 실제 클래스가 i 일 때(Positive) 예측 클래스 j 가 i 와 같은(j = i) 참(True)인 경우이다. FP 는 실제 클래스가 i 일 때 예측 클래스 j 가 i 가 아닌(j≠i) 거짓(False)인 경우이다. FN 은 실제 클래스가 i 가 아닐 때(Negative) 예측 클래스 j 가 i 인(j = i) 거짓(False)인 경우이다. TN 은 실제 클래스가 j 가 아닐 때 예측 클래스 j 가 i 가 아닌(j≠i) 참(True)인 경우이다[11].

5.1.1 정확도(Accuracy)

정확도는 실제 데이터에서 예측 데이터가 얼마나 같은지를 판단하는 지표로서 가장 직관적으로 모델의 성능을 나타낼 수 있는 평가 지표이다.

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

5.1.2 F1_score

F1_score 는 정밀도(Precision)와 재현율(Recall)의 조화평균으로서 데이터 label 이 불균형 구조일 때 모델의 성능을 정확하게 평가할 수 있다.

정밀도란 모델이 True 라고 분류한 것들 중에서 실제 True 인 것의 비율이다.

$$Precision = \frac{TP}{TP + FP}$$

재현율이란 실제 True 인 것들 중에서 모델이 True 라고 예측한 것의 비율이다.

$$Recall = \frac{TP}{TP + FN}$$

$$F1\_score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

5.2 실험 결과 분석

모델별 산출 결과의 평균을 <표 4>에 나타내었다. <표 4>의 img1 은 1 바이트를 grayscale 이미지의 한 픽셀로 변환하여 생성한 이미지이고 img2 는 2 바이트를 한 쌍의 좌표로 변환하여 생성한 이미지이다.

모델별 산출 결과의 평균을 비교하였을 때, img1 의 경우 정확도는 EfficientNet-B0 가 96.78%로 가장 높았으며, SqueezeNet 이 94.32%로 가장 낮았다. F1_score 는 ResNet-152 이 89.76%로 가장 높았고, Inception-v4 가 85.88%로 가장 낮았다. img2 의 경우 정확도는 ResNet-152 가 99.18%로 가장 높았고, SqueezeNet 이 98.49%로 가장 낮았다. F1_score 는 ResNet-152 이 98.18%로 가장 높았고, Inception-v4 가 96.37%로 가장 낮았다.

본 실험의 경우 데이터 셋의 label 값이 클래스 별로 상이하므로 정확도만으로 평가를 하기에는 모델 성능에 왜곡이 있을 것으로 보여 데이터 label 이 불균형 구조일때 모델의 성능을 정확하게 평가할 수 있는 F1_score 를 통해 성능평가를 진행하였다. 이때 img1 과 img2 모두 ResNet-

<표 4> 실험 결과

	ResNet-152		Inception-v4		SqueezeNet		EfficientNet-B0	
	img1	img2	img1	img2	img1	img2	img1	img2
Accuracy (%)	95.51	99.18	95.54	98.97	94.32	98.49	96.78	98.98
F1_score (%)	89.76	98.18	85.88	96.37	86.18	97.06	88.57	97.35

152 가 가장 높은 F1_score 값을 가지므로 악성코드 분류에 가장 적합한 모델은 ResNet-152 임을 알 수 있다.

추가로 img1 의 이미지화 방법과 img2 의 이미지화 방법을 비교해 보았을 때, F1_score 측면에서 평균적으로 10% 차이로 img2 가 더 높았다. 이를 통해 img2 의 이미지화 방법이 악성코드 분류에 더 적합함을 알 수 있었다.

본 연구의 실험 결과를 모든 경우에 일반화할 수는 없으나 Microsoft Malware Classification Challenge dataset 을 활용했을 때 2 바이트를 한 쌍의 좌표로 변환하여 생성한 이미지의 경우 4 개의 CNN 모델에서 모두 높은 성능을 보였다. 그중 ResNet-152 의 F1_score 가 98.18%로 가장 높았다.

6. 결론

본 연구는 악성코드 이미지화 분류에 적합한 CNN 모델을 도출하는 것이 목적이다. 사용된 CNN 모델은 2015 년 이후 ILSVRC 에서 뛰어난 성과를 거둔, ResNet, Inception-v4, SqueezeNet 과 추가로 2019 년에 발표된 EfficientNet 을 사용하였다.

악성코드를 이미지화하기 위해 1 바이트를 grayscale 이미지의 한 픽셀로 변환하는 방법과 2 바이트를 한 쌍의 좌표로 변환하는 방법을 사용하였으며, 두 방법 모두 이미지의 크기는 256 * 256 으로 고정하였다.

이미지화한 악성코드 전체 데이터의 65%를 이용해 학습을 진행하였으며, 15%를 랜덤으로 추출해 검증에 이용하고, 나머지 20%는 성능 평가를 위해 사용하였다.

평가 지표로는 정확도와 F1_score 를 사용하였고, 그 결과 CNN 모델에서는 ResNet-152 가 뛰어난 성능을 보였으며, 이미지화 방법에서는 2 바이트를 한 쌍의 좌표로 변환하는 방법이 적합하다.

2 바이트를 한 쌍의 좌표로 변환하는 방법을 통해 생성한 이미지를 CNN 으로 분류해 본 결과 4 개의 모델 모두 높은 성능을 보였다. 그중 ResNet-152 로 분류하는 것이 가장 높은 성능을 보임을 알 수 있다.

참고문헌

[1] HongGeun Kim, A Malware Classification Method By CNN-based Malware Visualization, Master's thesis in Korea, Graduate School of Engineering, Yeungnam University,2019

[2] Microsoft, “Microsoft Malware Classification Challenge” Available:https://www.kaggle.com/c/malware-classification (accessed 2021.November.13)

[3] Nataraj L, Karthikeyan S, Jacob G, and Manjunath B. S. , Malware Images: Visualization and Automatic Classification, Proceedings of the International Symposium on Visualization for Cyber Security, 2011

[4] Seonhee Seok, Howon Kim, Visualized Malware Classification Based-on Convolutional Neural Network, Journal of the Korea Institute of Information Security & Cryptology, 2016

[5] Jihyeon Park, Taeok Kim, Yulim Shin, Jiyeon Kim, Eunjung Choi, Design and Implementation of a Pre-processing Method for Image-based Deep Learning of Malware.Journal of Korea Multimedia Society, 2020

[6] He, Kaiming, et al. Deep residual learning for image recognition, Proceedings of the IEEE conference on computer vision and pattern recognition, 2016

[7] Szegedy, Christian, et al. Going deeper with convolutions, Proceedings of the IEEE conference on computer vision and pattern recognition, 2015

[8] Iandola, F., Han, S., Moskewicz, M., Ashraf, K., Dally, W., & Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, 2016

[9] Hu, Jie, Li Shen, and Gang Sun., Squeeze-and-excitation networks, Proceedings of the IEEE conference on computer vision and pattern recognition, 2018

[10] Mingxing Tan, Quoc V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2019

[11] Dong-Hyeon Seol, Ji-Hoon Oh, Hong-Jin Kim, Comparison of Deep Learning-based CNN Models for Crack Detection.JOURNAL OF THE ARCHITECTURAL INSTITUTE OF KOREA Structure & Construction, 2020