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

딥러닝 기술을 활용한 멀웨어 분류를 위한 이미지화 기법☆

Visualization of Malwares for Classification Through Deep Learning

김 형 겸[1] 한 석 민[1] 이 수 철[1*] 이 준 락[2*]
Hyeonggyeom Kim Seokmin Han Suchul Lee Jun-Rak Lee

요 약

Symantec의 인터넷 보안위협 보고서(2018)에 따르면 크립토재킹, 랜섬웨어, 모바일 등 인터넷 보안위협이 급증하고 있으며 다각화되고 있다고 한다. 이는 멀웨어(Malware) 탐지기술이 암호화, 난독화 등의 문제에 따른 질적 성능향상 뿐만 아니라 다양한 멀웨어의 탐지 등 범용성을 요구함을 의미한다. 멀웨어 탐지에 있어 범용성을 달성하기 위해서는 탐지알고리즘에 소모되는 컴퓨팅 파워, 탐지 알고리즘의 성능 등의 측면에서의 개선 및 최적화가 이루어져야 한다. 본고에서는 최근 지능화, 다각화 되는 멀웨어를 효과적으로 탐지하기 위하여 CNN(Convolutional Neural Network)을 활용한 멀웨어 탐지 기법인, stream order(SO)-CNN과 incremental coordinate(IC)-CNN을 제안한다. 제안기법은 멀웨어 바이너리 파일들을 이미지화 한다. 이미지화 된 멀웨어 바이너리는 GoogLeNet을 통해 학습되어 딥러닝 모델을 형성하고 악성코드를 탐지 및 분류한다. 제안기법은 기존 방법에 비해 우수한 성능을 보인다.

☞ 주제어 : 멀웨어 이미지 생성, 멀웨어 탐지 및 분류, 딥러닝, CNN

ABSTRACT

According to Symantec's Internet Security Threat Report(2018), Internet security threats such as Cryptojackings, Ransomwares, and Mobile malwares are rapidly increasing and diversifying. It means that detection of malwares requires not only the detection accuracy but also versatility. In the past, malware detection technology focused on qualitative performance due to the problems such as encryption and obfuscation. However, nowadays, considering the diversity of malware, versatility is required in detecting various malwares. Additionally the optimization is required in terms of computing power for detecting malware. In this paper, we present Stream Order(SO)-CNN and Incremental Coordinate(IC)-CNN, which are malware detection schemes using CNN(Convolutional Neural Network) that effectively detect intelligent and diversified malwares. The proposed methods visualize each malware binary file onto a fixed sized image. The visualized malware binaries are learned through GoogLeNet to form a deep learning model. Our model detects and classifies malwares. The proposed method reveals better performance than the conventional method.

☞ keyword : Malware visualization, mawlare detection and classification, deep learning, CNN

1. 서 론

최근 ICT(Information and Communication Technology)기술이 인터넷 시대를 맞아 급속하게 발전하였다. 무선 통신 분야에서 5G기술의 상용화를 목전에 두고 있으며, 인공지능 기술이 의료, 자연어 처리 등 다양한 분야에 접목된 형태로 연구수준에서 머물던 다양한 기술들이 실현가능한 형태로 구현되고 있다.

정보보호 관점에서는 ICT기술의 발전의 긍정적인 영향만을 고려할 수 없다. 다양한 어플리케이션이 등장함에 따라 매일 새로운 멀웨어가 새롭게 개발되어 전파되고 있다. 우리는 언론 등 디지털미디어를 통해 그중의 극히 일부만을 목격하고 있는 것이다. Symantec의 인터넷 보안위협 보고서(2018) [1]에 따르면 블록체인 기술이 급격한 관심을 불러일으킴에 따라 크립토재킹이 85배 증가했다고 한다. 랜섬웨어는 여전히 기승을 부리고 있으며 모바일 악성 코드가 확산을 거듭하고 변종도 54%까지 증가했다. 이처럼 점차 지능화 되고 있는 보안위협에 대응하여 인공지능기술에 기반을 둔 방어방법들이 제안되었다.

멀웨어의 탐지(detection) 혹은 분류(classification)은 방

1. Dept. of Computer Science and Information Engineering, Korea National University of Transportation, Uiwang, Kyunggi, 16106, Korea.
2. Dept. of Humanities and Social Sciences, Kangwon National University, Samcheok, Kangwon, 25913, Korea.
* Co-corresponding authors(jrlee@kangwon.ac.kr, sclee@ut.ac.kr)
[Received 17 July 2018, Reviewed 20 July 2018(R2 6 August 2018), Accepted 16 August 2018]
☆ 이 성과는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2017R1C1B5017028). 이 연구는 2018년 한국교통대학교 지원을 받아 수행하였음. 이 연구는 2017년도 강원대학교 대학회계 학술연구조성비로 연구하였음(관리번호-620170073).

법론(methodology)측면에서 다양성을 가지나 기저의 근본적인 원리는 같다. 멀웨어 분류 문제는 PE파일, 스크립트를 포함한 S/W, 네트워크 플로우(패킷) 등을 멀웨어의 종류에 따라서 분류(n-ary classification)한다. 반면, 멀웨어 탐지문제는 이를 악성(malicious) 또는 비악성(benign)으로 이진 분류(binary classification)한다.

멀웨어의 탐지문제는 정보보호 분야에서 극도로 보수적인 접근이 요구된다. 예컨대 인공지능 기술을 활용한 멀웨어의 분류기법 A의 Precision 성능이 0.95가 나온다고 가정하자. 상당히 높은 수치임에도 불구하고 이를 멀웨어 탐지의 최종단계로서는 활용할 수 없다. 왜냐하면 이 방법을 멀웨어 탐지의 최종단계로서 사용한다면 이상적인 상황*을 가정해도 5%의 멀웨어는 A기술을 우회할 것이기 때문이다. 그러나 본고에서는 아이러니 하게도 A기술을 사용할 수 없다고 주장하지 않는다. 정보보호문제에서는 A기술을 최종적으로 활용할 수 없지만 A기술을 우회하는 5%를 탐지하기 위해 정보보호전문가 혹은 CERT (Computer Emergency Response Team)의 노력을 대폭 감소시켜 줄 수 있기 때문이다. 매일 양산되는 수많은 S/W를 모두 분석해야 하는 것은 노동 집약적(labor intensive)이며 심지어는 그렇게 할 필요도 없다.

딥러닝 기술은 현재 가장 각광받고 있는 인공지능 기술의 갈래중의 하나이다. 넓게는 Machine Learning(ML)으로 정의되는 기술의 일부이며 이미지 또는 series 형태의 데이터 분류에서 우수한 성능을 보임이 널리 알려져 있다. 이미지 분류에 딥러닝을 적용하려면 통상적으로 Convolutional Neural Network(CNN)를 사용하고, 자연어 처리, 번역 등 series데이터에 딥러닝을 적용하려면 Recurrent Neural Network(RNN)를 사용한다. 본고에서 제안하는 방법은 멀웨어를 이미지화하는 기법을 사용하므로 CNN을 사용한다. 최근 수년간 인공지능 기술을 활용한 멀웨어 분류방법에 관한 수많은 연구가 진행되었다 [2~13]. 관련 연구 동향은 2장에서 기술한다.

본고에서는 CNN기술을 활용하여 멀웨어를 분류하는 기술을 제안한다. CNN기술을 활용하기 위해서한 다양한 형태로 존재하는 멀웨어를 정형화된 이미지로 변환하여야 한다. 이미지로 변환하는 단계를 Image Making(IM) 단계로 정의하며, 이 과정은 멀웨어 N개를 정형화된 N개의 256x256 픽셀이미지로 변환하는 과정이다. IM에서는 stream order 이미지 생성방법과 [2] incremental coordinate 이미지 생성방법을 [3] 활용하여 CNN에 적용 가능하도록 멀웨어 각각에 대하여 이미지를 생성한다. 생성된 이미지는 Label과정을 거친 후 Google Inception V3** [16] 모델을 TensorFlow [17]를 활용하여 학습한다. 본고에서는 MS에서 Kaggle을 통해 발표한 Microsoft Malware Classification Challenge dataset [14]을 이용하여 제안기법 및 비교기법의 성능을 평가한다. stream order(SO)및 incremental coordinate(IC)이미지 생성기법과 CNN을 결합한 방법을 본고에서는 각기 SO-CNN, IC-CNN이라 명명하였다. SO-CNN기법과 IC-CNN기법의 분류 정확도(Test Accuracy)는 94.3%, 98.0%이다. 이 성능은 각 방법은 GoogLeNet으로 400,000번, 200,000번 정도 학습하였을 경우 얻을 수 있으며, GTX1080 GPU를 1개 탑재한 통상의 PC에서 5시간이상 학습을 진행해야 한다.

본고의 구성은 다음과 같다. 2장에서는 관련연구를 요약하고 3장에서는 제안 기법을 상세히 기술한다. 4장에서 제안 기법의 성능을 평가하고 관련한 이슈들을 논한다. 마지막으로 5장에서 결론을 맺는다.

2. 관련 연구 동향

인공지능 등 기계학습 기반 멀웨어 분석 및 분류기술은 전처리 단계를 거치게 된다. 통상적으로 기계학습 연구 분야에서는 이 과정을 특징 공학(feature engineering)이라고 한다. 어떠한 특징(feature)을 어떠한 방법으로 가공 및 적용할 것인지에 관련된 연구분야라고 볼 수 있다. 멀웨어 분류에서는 정적(static), 동적(dynamic), 이미지화(visualized)분석 등 대개 세 가지 방법론으로 귀결된다.

2.1 정적 분석

[4]에서 정적 분석에서 멀웨어 탐지를 위한 데이터 마이닝 개념을 도입하였다. PE(Portable Executable), 문자열 및 바이트 시퀀스의 세 가지 정적 feature에 Ripper알고리즘을 [5] 적용하였다. Kong et al의 연구에서는 [6] 멀웨어의 함수 호출 그래프를 기반으로 한 분류기법을 제안하였다. 모델의 ensemble값을 통해 멀웨어 간 거리를 정량화 하고 이를 통해 멀웨어를 분류하는 기법을 제안했다. Li and Li의 연구는 [7] 모바일 환경을 대상으로 진행되었는데, 안드로이드 APK에서의 API 호출 및 코드구조의 통계적인 유사성에 근거하여 모바일 환경에서 동작하는 멀

* 특정 데이터로 생성한 인공지능 모델(멀웨어 탐지/분류방법)이 타 데이터에도 동일한 성능을 낼 수 있는 상황.

** GoogLeNet이란 이름으로 널리 알려져 있다.

웨어를 분류할 수 있다.

정적 분석분야에서 최근 많이 시도되고 있는 방법은 역어셈블을 활용하는 방법이다. 통상의 PE파일형태로 존재하는 멀웨어의 소스코드를 얻기는 매우 어려우므로 이를 역어셈블하여 어셈블리어 수준에서 특징공학을 적용하는 방법론이다. 예컨대, [8]에서는 실행 파일의 벡터 표현을 구성하기 위해 opcode 시퀀스를 활용하는 방법을 제안했다. 이러한 정적 분석 방법의 장점은 동적 분석방법에 비하여 많은 노력을 들이지 않고 상당한 수준의 멀웨어 분류 성능을 얻어낼 수 있다는 장점이 있는 반면, 공격자는 난독화, 암호화 등 악의적 obfuscation방법을 멀웨어에 적용하여 다양한 경로로 정적 탐지 방법들을 우회할 수 있다.

2.2 동적 분석

정적분석의 약점에 대응하기 위한 방법으로 동적방법이 제안되었다. 멀웨어의 실행과정에서 생성되는 네트워크 패킷, 혹은 난독화, 암호화가 해제된 멀웨어, 실질적인 악성행위를 유발하는 멀웨어의 핵심 소스코드 등을 특징 공학에 활용할 수 있는 방법이다. 이러한 특징들을 수집하기 위해서는 멀웨어의 실행이 수반되어야 하는데, 이는 외부망(인터넷)과 철저히 격리된 가상머신(virtual machine)환경에서 이루어져야 하며 이를 샌드박스(sandbox)라 [15] 한다.

Bayer et al의 연구는 [9] PE파일을 행위기반으로 클러스터링하고 이를 기반으로 멀웨어를 탐지/분류하는 기법을 제안했다. [10]에서는 opcode trace를 기반으로 그래프를 생성하고 그래프 분석을 통한 멀웨어 탐지 기법을 제시하였다. Fujino et al의 연구에서는 [11] 멀웨어의 API 함수 호출을 근거로 멀웨어의 유사성을 정량화할 수 있는 척도를 제안하였으며, 해당 척도를 non-negative matrix factorization 기법을 적용하여 멀웨어를 탐지할 수 있는 방안을 제안하였다.

동적 분석은 정적 분석의 문제점을 해결할 수 있는 방안으로 정보보호 분야에서 다양한 실용적인 응용들이 제안되었다. 그러나 동적 분석에 기반을 둔 멀웨어 탐지기법을 우회하기 위해 교활한 공격자들은 다양한 기법들을 적용할 수 있다. 예컨대, 멀웨어의 실제 악성행위까지의 잠복기간을 가지도록 할 수 있다. 많은 코드를 테스트해야 하는 샌드박스는 잠복기에 있는 멀웨어의 활동을 기다려줄 많은 시간이 없다. 일부 멀웨어는 네트워크의 연결을 통해서 trigger되도록 할 수 있다. 인터넷과 격리된 상태에서 동작하는 샌드박스에서는 영원히 해당 멀웨어의 동작은 시작되지 않을 것이다. 이렇듯 동적 분석이 항상 악의적인 동작을 발견하지는 못한다. 또한 근본적으로 동적 분석을 사용한다 한들 악성 행위를 항상 특징으로 적절히 활용할 수 있음을 보장할 수 없다.

2.3 이미지화 분석

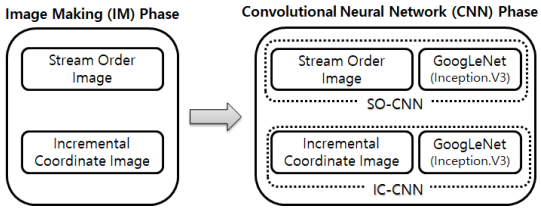
최근 멀웨어를 이미지로 변환하여 이미지 수준에서 멀웨어를 분류하고 탐지하는 다양한 기법들이 제안되었다 [2][12][13]. [12]가 가장 최신의 연구로서 opcode시퀀스를 활용하여 simple hashing기법을 도입, 이를 통해 멀웨어를 분류하는 기법을 제안했다. Nataraj et al의 연구는 [2] 멀웨어 바이너리를 그레이 스케일 이미지로 시각화하는 이미지 처리 기술을 사용하였으며, K-nearest neighbor(KNN) 기계학습 기법을 통해 멀웨어를 분류하였다. [13]에서는 시각화 된 이미지 및 엔트로피 그래프를 사용하여 멀웨어 분류 및 변종의 탐지 등을 수행하는 방법을 제안하였다.

아마도 본고에서 제안하는 기법과 가장 유사도가 높은 기술은 [2][3]일 것이다. [2]에서는 stream order 이미지화 방법을 제안하였으며, 이를 Gabor filter와 KNN방법을 적용하여 성능을 평가하였다. 본고에서는 stream order 방법을 개선하여 CNN을 적용할 수 있도록 하였다. [3]에서는 incremental coordinate방법을 적용하였으며 본고에서도 동일한 이미지화 방법을 적용하였다. 그러나 어떠한 CNN모델을 적용하느냐에 따라 성능을 개선할 수 있는 여지가 있음에 착안, 다양한 CNN 네트워크 모델에 대하여 성능을 실험적으로 평가하였다. 가장 우수한 성능을 내는 CNN모델은 GoogLeNet으로 널리 알려진 Inception.V3 [16] 이었으며 제안기법에서도 해당 네트워크를 채택하였다.

3. 딥러닝 기반 멀웨어 분류 기법

제안하는 기법은 앞서 기술한 바와 같이 SO-CNN방법과 IC-CNN기법이다. CNN을 활용한 알고리즘은 대개 특징 추출(feature extraction), 멀웨어 이미지 생성(malware visualization), CNN 학습으로 세분화된다. 특징 추출 단계에서 적절한 특징을 추출하느냐에 따라서 성능이 개선될 여지가 있으나 본고에서는 논외로 한다. 본고에서는 특징 추출을 따로 수행하지 않고 멀웨어의 수정, 변경 없이 활용하여 멀웨어별로 이미지를 생성하는데, 해당 이미 생성 과정을 Image Making(IM)단계라 명명한다. 마지막으로 생성된 이미지들은 GoogLeNet을 활용하는 CNN단계를

통해 최종 멀웨어 분류기법이 완성된다. 그림 1에서 제안 기법의 전체적인 동작과정을 도식하였다.



(그림 1) 제안기법의 전체적인 동작과정
(Figure 1) Overall procedure of the proposed scheme

이어지는 3.1절에서는 제안기법의 IM단계를 상세하게 기술하며, 3.2절에서 CNN단계를 상세하게 기술한다.

3.1 멀웨어 이미지 생성

3.1.1 Stream Order(SO) 이미지화 기법

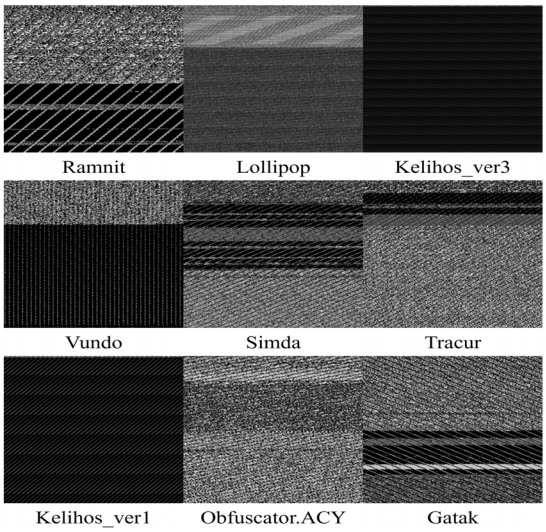
첫 번째 제안기법인 SO-CNN에서 stream order 이미지화 기법은 N개의 멀웨어 파일에 대해 N개의 이미지를 생성한다. 각 멀웨어에 대한 이미지 생성기법은 다음과 같다.

① 멀웨어 파일의 크기(S)를 측정한다. 정사각형 형태의 이미지 초안을 생성하는데, 이미지의 크기는 파일의 크기에 의해 $\lfloor\sqrt{S}\rfloor$ x $\lfloor\sqrt{S}\rfloor$로 결정된다. 예컨대, 초안 이미지의 모서리의 좌표는 총 네 개이며 (0, 0), (0, $\lfloor\sqrt{S}\rfloor$ - 1), ($\lfloor\sqrt{S}\rfloor$ - 1, 0), ($\lfloor\sqrt{S}\rfloor$ - 1, $\lfloor\sqrt{S}\rfloor$ - 1)이다.

② 멀웨어 파일을 처음부터 끝까지 2bytes(16bits)씩 읽는다. 읽어오는 값은 2bytes이므로, 0~255범위의 값이며 순차적으로 해당 값을 이미지의 좌표에 비트맵이미지로 표현한다. 예컨대 멀웨어 파일에서 읽어오는 값이 (0x00, 0x00, 0x00, ...)이라고 하면 초안 이미지에의 좌표 (0, 0), (0, 1), (0, 2)는 흰색(0), 반전일 경우 검정색(255)일 것이다. ①에서 floor연산으로 인해 멀웨어 파일에서 읽는 횟수보다 초안 이미지의 좌표의 총 개수가 적을 수 있는데 이 경우는 멀웨어 파일에서 읽어 들였으나 이미지화 할 수 없으며 그 개수가 매우 적으므로 이미지 생성에 있어 활용하지 않는다.

③ 정사각형의 초안 이미지를 256x256크기의 이미지로 resizing한다.

stream order 이미지화 기법을 통해 생성된 이미지의 샘플은 멀웨어의 종류별로 (그림 2)에 도식하였다.



(그림 2) Stream order이미지화 기법을 통해 생성된 이미지
(Figure 2) Image samples generated from the stream order method

3.1.2 Incremental Coordinate (IC) 이미지화 기법

두 번째 제안기법인 IC-CNN에서도 SO방법과 마찬가지로 incremental coordinate 이미지화 기법은 N개의 멀웨어 파일에 대해 N개의 이미지를 생성한다. 각 멀웨어에 대한 이미지 생성기법은 다음과 같다.

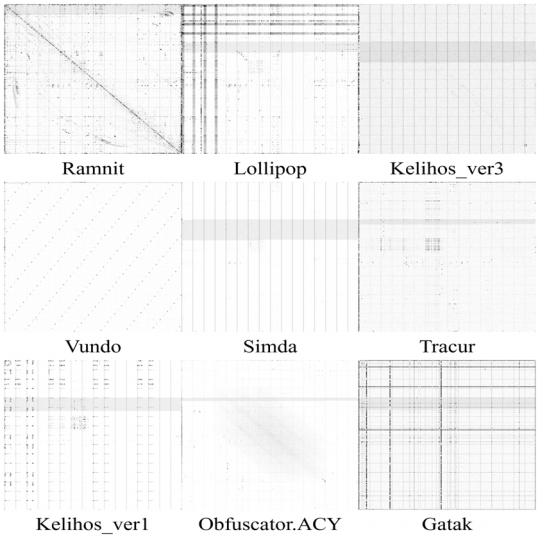
① 멀웨어 파일을 처음부터 끝까지 4bytes(32bits)씩 읽는다. 읽어오는 값은 4bytes이므로 2bytes씩 분할하여 2개의 값을 만들 수 있으며 이를 0~255값을 범위로 가지는 값 한 쌍(pair)을 좌표로 변환할 수 있다. 예컨대 멀웨어 파일에서 읽어온 값이 0xFFFF이면 좌표는 (255, 255)가 된다.

② IC기법에서는 256x256의 2차원 배열을 활용한다. 변환된 모든 좌표는 256x256크기의 2차원 배열과 1:1로 맵핑된다 이를 A라 가정하자. 멀웨어에서 읽어오는 값들에 대하여 맵핑된 배열의 값을 1증가시킨다. 예컨대 읽어온 값이 0xFFFF이면 2차원 배열에서 A[255][255]값을 1증가 시킨다.

③ 모든 값을 다 읽어오면 2차원 배열 A는 다양한 값한 값을 원소로 갖게 된다. A를 이미지로 변환하기 위

해 A의 원소의 값 중 최댓값으로 A를 normalize한다. normalized A의 모든 원소의 값에 256을 곱하고 floor 연산을 수행하면 해당 배열은 그레이 스케일 256x256 비트맵 이미지로 변환할 수 있다. 반전을 위해서는 A배열의 각 원소의 값을 255에서 각 원소의 값을 뺀 값으로 대체하면 된다.

incremental coordinate 이미지화 기법을 통해 생성된 이미지의 샘플은 멀웨어의 종류별로 그림 3에 도식하였다.

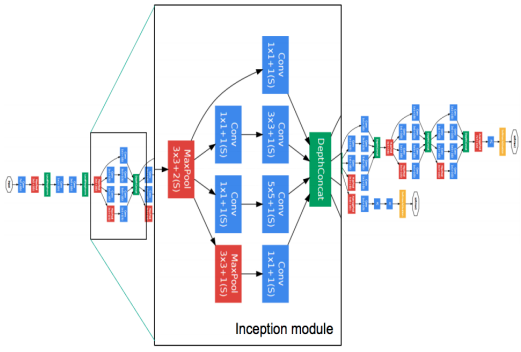


(그림 3) Incremental coordinate 이미지화 기법을 통해 생성된 이미지
(Figure 3) Image samples generated from the incremental coordinate method

3.2 Convolutional Neural Network(CNN) 적용

입력 데이터의 지역적 특징 정보를 추출하여 학습에 사용하는 CNN은 숫자를 인식하기 위해 LeNet [18]부터 본격적으로 사용되기 시작했다. Convolution, pooling, complete link 등으로 구현된 계층(layer)을 반복적으로 사용하는 neural networks의 일종인 CNN은 이미지 인식을 위한 심화학습에서 가장 많이 사용되는 구조로 자리매김했다. 최근 비선형 활성화 함수인 ReLU(Rectified Linear Unit)계층과 Drop-out 계층이 등장하면서 오버 피팅을 줄이고 성능이 더욱 향상되는 모습을 보이고 있다.

LeNet 이후 성능을 더욱 개선하기 위해 많은 CNN 구조가 제시되었다. 제시된 많은 CNN 구조 중 매년 개최되는 세계적인 이미지인식 대회인 ImageNet Large Scale Visual Recognition Competition(ILSVRC) [19]에서 우수한 성능을 보이는 몇몇 CNN 네트워크 구조가 제안되었다. 본고에서는 비교적 최신의 다양한 네트워크를 우리의 실험에 적용하였으며 최종적으로 GoogLeNet[16]이 우수한 성능을 냄을 실험적으로 발견하였다. CNN 구조를 가진 GoogLeNet은 기존의 CNN 구조들보다 훨씬 더 깊은 구조를 가진다. GoogLeNet이 기존 CNN의 구조와 가장 차별화 되는 특성은 인셉션 모듈을 가진다는 점이다. 인셉션 모듈은 Arora의 [20] 방법을 응용하여 설계되어 네트워크 안의 네트워크 구조로 설계되어 기존의 1차원 직렬 구조만을 가지는 CNN과는 차이가 있다. 그래서 GoogLeNet의 공식이름은 Inception.V3이다. 그림 4에 GoogLeNet의 네트워크 구조를 도식하였다.



(그림 4) GoogLeNet의 네트워크 구조
(Figure 4) Network structure of GoogLeNet

3.1절에서 기술한 두 가지 이미지 생성방법들로 생성된 256x256크기의 멀웨어 이미지들이 GoogLeNet의 인풋데이터로 사용된다. 모든 멀웨어 이미지는 레이블되어 있어야 하며, GoogLeNet의 마지막 계층인 Softmax classifier에 의해 최종적으로 멀웨어 9종 중 하나로 분류된다.

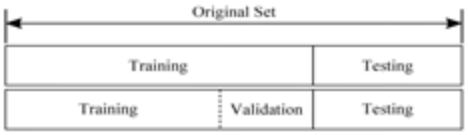
4. 성능평가

4.1 실험 환경

4.1.1 데이터셋 및 실행 환경

본고에서 사용한 실험 데이터셋은 Microsoft에서 연구 및 멀웨어 탐지기법 경쟁을 위해 Kaggle을 통해 발표한 Microsoft Malware Classification Challenge dataset [13]

이다. 해당 악성코드는 Rammit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, Gatak의 9종, 총 10,868개의 멀웨어에 대한 샘플로 구성된다.



(그림 5) 데이터셋의 개념 정의
(Figure 5) Conceptual definition of datasets

그림 5에 도식한 바와 같이 랜덤추출을 통해 전체 데이터(original data)의 75%를 학습(trainingt)을 위해 사용하고 전체데이터에서 랜덤 추출한 15%를 학습 iteration마다 validation으로 나머지(testing) 10%를 성능평가를 위해 사용한다. 학습과정에서 딥러닝 모델 fitness를 측정하기 위하여 학습 iteration마다 validation accuracy를 측정한다. 본고에서는 CNN에 적용 가능한 다양한 네트워크에 대한 실험을 진행하였으며 그 결과 GoogLeNet (Inception V3) [6]의 성능이 우수함을 실험적으로 확인, GoogLeNet을 제안기법에서 사용할 네트워크로 채택하였다. 실험에 사용된 모든 네트워크는 python으로 작성된 TensorFlow를 활용하여 구현하였다. 실험에 사용된 PC는 Intel Core i7-8700, 16GB RAM, GTX1080 GPU가 탑재되었다.

4.1.2 성능 평가 척도

본고에서는 성능 평가 척도(performance metric)로서 통상적으로 활용되는 분류 정확도(overall accuracy), 정밀도(precision), 재현율(recall), F점수(F-measure)를 활용한다. 성능 평가 척도들은 통계적으로 다음과 같이 정의된다.*.

$$\text{분류정확도} = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

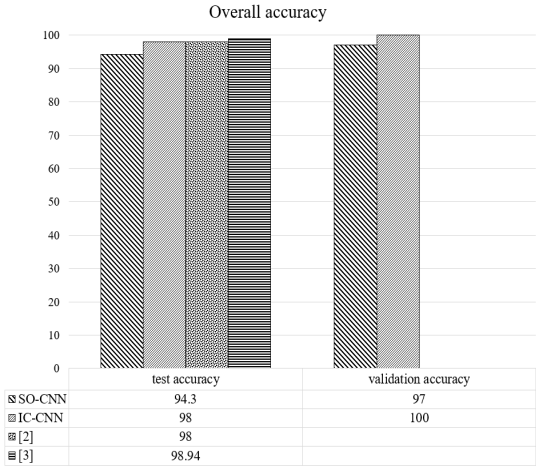
$$\text{정밀도} = \frac{TP}{TP+FP},\ \text{재현율} = \frac{TP}{TP+FN} \quad (2)$$

$$F\text{점수} = \frac{2\times \text{정밀도}\times \text{재현율}}{\text{정밀도}+\text{재현율}} \quad (3)$$

* TP는 true positive의 개수, TN은 true negative의 개수 FP는 false positive의 개수, FN은 false negative의 개수를 의미한다.

4.2 실험 결과

4.2.1 분류 정확도



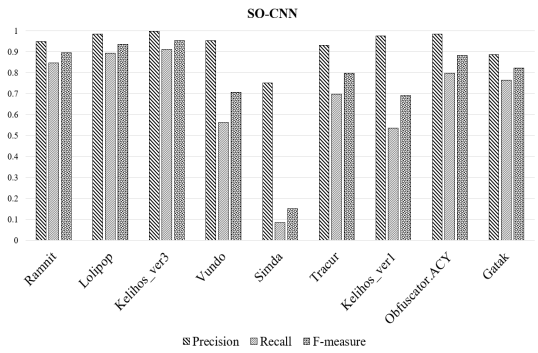
(그림 6) 각 기법의 분류 정확도
(Figure 6) Overall accuracy

IC와 SO 이미지화 기법을 통해 생성한 이미지를 CNN을 통해 학습한 결과는 (그림 6)과 같다. 제안기법 SO-CNN과 IC-CNN의 test 분류 정확도는 94.3%, 98%이며, validation 분류 정확도는 97%, 100%이다. 전반적으로 IC-CNN의 방법의 성능이 우수한 것으로 판명되었다. 이는 비교방법들과 비교해서 우수한 성능을 보이며, 특히 IC-CNN방법을 통해서는 100%의 validation 분류 정확도를 달성하므로, [19]의 멀웨어 분류 대회에서 제안된 모든 기법들과의 비교 가능한 우수한 성능을 낸다고 판단할 수 있다. [2][3]의 성능은 test accuracy라고 가정하였다.

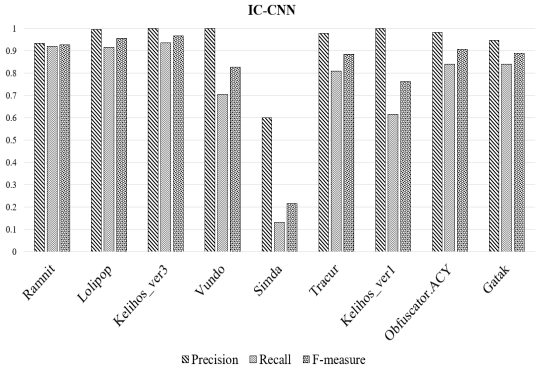
4.2.2 멀웨어 종류별 정밀도, 재현율, F점수

(그림 7)과 (그림 8)은 SO-CNN과 IC-CNN의 정밀도, 재현율, F점수를 나타낸다. 9종의 멀웨어 클래스에 대하여 90%이상의 정밀도를 달성함을 확인할 수 있다. 주목할 만한 결과는 9종의 멀웨어 클래스 중 Simda 멀웨어 군의 정밀도, 재현율, F점수가 다른 클래스에 비해 현저히 낮다는 점이다. 이러한 실험결과가 도출된 이유에 대해 심도 있는 분석결과, 실험 데이터셋의 10,868개의 멀웨어 중 Simda는 42개이다. 통상적으로 9종의 멀웨어가 균등하게 분포한다면 멀웨어 종류별로 1000개 이상의 멀웨어 샘플 데이터가 있어야 한다. 그렇지만 Simda는 겨우 42개의 데

이터만을 이용하여 멀웨어 이미지 생성 및 딥러닝 학습을 수행하므로 TP와 FN의 값에 차이가 크게 발생하여 성능 평가에 부정적인 영향이 가해졌음으로 판단된다. 우리는 이를 해결할 수 있는 방안에 대하여 현재 연구를 진행 중이다. 이미지화 된 데이터셋을 늘리는 방법, 혹은 simda의 샘플을 추가적으로 수집하는 방법 등을 포함한다. 다만, 본고에서는 관련 연구를 future work으로 남긴다.



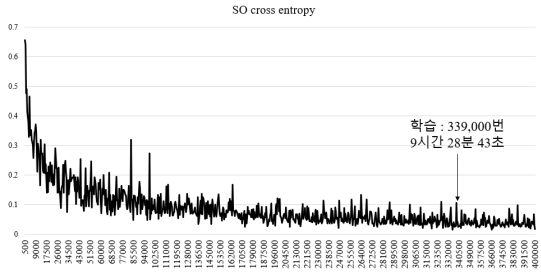
(그림 7) SO-CNN의 클래스별 분류 정밀도, 재현율, F점수
(Figure 7) Precision, recall, F-measure of SO-CNN



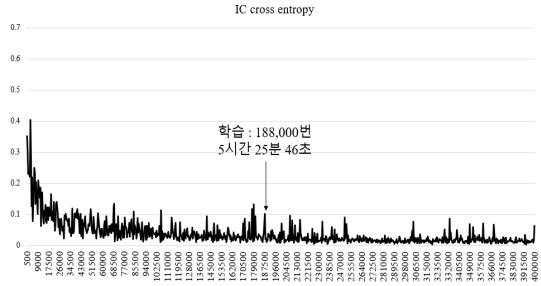
(그림 8) IC-CNN의 클래스별 분류 정밀도, 재현율, F점수
(Figure 8) Precision, recall, F-measure of IC-CNN

4.2.3 학습에 따른 cross 엔트로피 변화 및 소요 시간

(그림 9)와 (그림 10)은 SO-CNN과 IC-CNN의 cross 엔트로피 변화를 나타낸 그래프이다. IC-CNN은 188,000번 training 부터 cross 엔트로피 값이 0.1을 넘지 않는다. 이는 우리의 실험용 PC에서 대략 5시간 26분 정도 학습이 필요하다. 반면, SO-CNN은 339,000번 이상 training을 진행해야 cross 엔트로피 값이 0.1을 넘지 않는다. 이는 우리의 실험용 PC에서 대략 9시간 29분 정도 학습이 필요하다. 즉 학습 진행을 위해 1.7배 이상의 시간이 소요됨을 알 수 있다.



(그림 9) SO-CNN 학습과정동안의 cross 엔트로피 변화
(Figure 9) Cross entropy change of SO-CNN training



(그림 10) IC-CNN 학습과정동안의 cross 엔트로피 변화
(Figure 10) Cross entropy change of IC-CNN training

5. 결 론

본고에서는 CNN기술을 활용하여 멀웨어를 분류하는 기술을 제안하였다. SO와 IC 이미지화 기법을 통해 멀웨어의 시각화된 특징을 CNN에 적용 가능하도록 하였다. 제안기법은 Microsoft Malware Classification Challenge dataset을 활용하여 성능을 평가하였다. 성능평가 결과, 기존 방법대비 우수한 성능을 보였다.

그러나 본고에서 사용한 이미지화 기법은 멀웨어의 특징공학을 통한 성능향상을 기대할 수 없는 한계가 있다. 예컨대 제안기법은 특징공학 기술적용 없이 멀웨어의 변형을 최대한 배제한 상태로 적용되었다. 이는 다양한 방법들을 통해 정확도 측면에서의 성능을 극대화하거나, 딥러닝 모델의 학습에 소요되는 시간을 획기적으로 줄일

수 있는 여지가 존재함을 의미한다. 본 연구팀에서는 향후 이미지의 결합 방안이나 특징 공학의 장점을 추가적으로 활용하는 방안 등 연구를 진행할 예정이다.

참고문헌(Reference)

[1] “Innovation, organisation, and sophistication－these are the tools of cyber attackers as they work harder and more efficiently to uncover new vulnerabilities”, Symantec Internet Security Threat Report, 2018. https://resource.elq.symantec.com/LP=5840?cid=70138000000mleAAA

[2] Nataraj L, Karthikeyan S, Jacob G, Manjunath B. S., “Malware images: visualization and automatic classification”, In proc. of the 8th ACM international symposium on visualization for cyber security 2011. http://doi.org/10.1145/2016904.2016908

[3] Ji H., and Im E., “Malware Classification Using Machine Learning and Binary Visualization”, the Korea Computer Congress. KCC, pp.1084－1086, 2017. http://dx.doi.org/10.5626/KTCP.2018.24.4.198

[4] Schultz MG, Eskin E, Zadok F, Stolfo SJ, “Data mining methods for detection of new malicious executables”, In IEEE symposium on security and privacy(S&P ’01), 2001. https://doi.org/10.1109/SECPRI.2001.924286

[5] Cohen W. W., “Fast effective rule induction”, In Proceedings of the Twelfth International Conference on Machine Learning, 1995. https://doi.org/10.1016/B978-1-55860-377-6.50023-2

[6] Kong D. and Yan G., “Discriminant malware distance learning on structural information for automated malware classification”, In ACM SIGKDD 2013. http://dx.doi.org/10.1145/2487575.2488219

[7] Li Q. and Li X., “Android malware detection based on static analysis of characteristic tree”, In international conference on cyber-enabled distributed computing and knowledge discovery (cyberc), 2015. https://doi.org/10.1109/CyberC.2015.88

[8] Santos I., Brezo F., Ugarte-Pedrero X., Bringas P. G., “Opcode sequences as representation of executables for data-mining-based unknown malware detection”, Elsevier Information Sciences, Vol. 231, pp. 64-82, 2013. https://doi.org/10.1016/j.ins.2011.08.020

[9] Bayer U., Comparetti P. M., Hlauschek C., Kruegel C., and Kirda E., “Scalable, behavior-based malware clustering”, In NDSS 2009. https://www.ndss-symposium.org/ndss2009/scalable-behavior-based-malware-clustering/

[10] Anderson B., Quist D., Neil J., Storlie C., and Lane T., “Graph-based malware detection using dynamic analysis”, Journal in computer Virology, Vol. 7, 247-258, 2011. https://doi.org/10.1007/s11416-011-0152-x

[11] Fujino A., Murakami J., and Mori T., “Discovering similar malware samples using api call topics”, In IEEE CCNC, 2015. https://doi.org/10.1109/CCNC.2015.7157960

[12] Ni S., Qian Q., and Zhang R., “Malware identification using visualization images and deep learning”, Elsevier Computers & Security, 2018. https://doi.org/10.1016/j.cose.2018.04.005

[13] Han KS, Lim JH, Kang B, Im EG, “Malware analysis using visualized images and entropy graphs”, Int Journal of Information Security, Vol.14, pp. 1-14, 2015. https://doi.org/10.1007/s10207-014-0242-0

[14] Ronen R., Radu M., Feuerstein C., Yom-Tov E., and Ahmadi M., “Microsoft Malware Classification Challenge”, arXiv preprint arXiv:1802.10135, 2018. https://arxiv.org/abs/1802.10135

[15] Gong L., Mueller M, Prafullchandra H, and Schemers R., “Going beyond the sandbox: An overview of the new security architecture in the Java development kit 1.2”, In USENIX Symposium on Internet Technologies and Systems, 1997. https://www.usenix.org/conference/usits-97/going-beyond-sandbox-overview-new-security-architecture-java-development-kit-12

[16] Szegedy C., Liu W., Jia Y., Sermanet P., Reed S., Anguelov D., and Rabinovich A., “Going deeper with convolutions”, In Proceedings of the IEEE conference on computer vision and pattern recognition(CVPR), 2015. https://doi.org/10.1109/CVPR.2015.7298594

[17] Abadi M., Barham P., Chen J., Chen Z., Davis A., Dean J., and Kudlur M., “TensorFlow: A system for large-scale machine learning”, in the Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation(OSDI), pp. 265-283, 2016. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[18] LeCun Y., Bottou L., Bengio Y., and Haffner P., “Gradient-Based Learning Applied to Document Recognition”, in Proceeding of the IEEE 86.11, pp. 2278-2324, 1998.
https://doi.org/10.1109/5.726791
[19] “ImageNet Large Scale Visual Recognition Competition”, http://www.image-net.org/challenges/LSVRC/
[20] Arora S., Bhaskara A., Ge R., and Ma T., “Provable bounds for learning some deep representations”, In International Conference on Machine Learning, pp. I-584-I-592, 2014.
http://proceedings.mlr.press/v32/arora14.pdf

◐ 저 자 소 개 ◑

김 형 겸(Hyeonggyeom Kim)
2015년~현재 한국교통대학교 철도대학 철도경영·물류·컴퓨터학부(컴퓨터정보공학전공) 학사과정
관심분야 : 정보보호 및 인공지능
E-mail : hyeonggyeom.kim@gmail.com

한 석 민(Seokmin Han)
2000년 서울대학교 전기공학부(공학사)
2003년 서울대학교 대학원 전기·컴퓨터공학부(공학석사)
2008년 서울대학교 대학원 전기·컴퓨터공학부(공학박사)
2008년~2015년 삼성전자 종합기술원
2015년~2017년 삼성전자 의료기기 사업부
2017년~현재 한국교통대학교 철도대학 철도경영·물류·컴퓨터학부(컴퓨터정보공학전공) 부교수
관심분야 : 정보통신 및 보안
E-mail : seokmin.han@ut.ac.kr

이 수 철(Suchul Lee)
2008년 서울대학교 전기·컴퓨터공학부(공학사)
2014년 서울대학교 대학원 컴퓨터공학부(공학박사)
2014년~2016년 한국전자통신연구원 부설연구소 연구원
2016년~현재 한국교통대학교 철도대학 철도경영·물류·컴퓨터학부(컴퓨터정보공학전공) 조교수
관심분야 : 정보통신 및 보안, 인공지능
E-mail : sclee@ut.ac.kr

이 준 락(Jun-Rak Lee)
1984년 인하대학교 수학과(이학사)
1986년 인하대학교 대학원 수학과(이학석사)
1991년 인하대학교 대학원 수학과(이학박사)
1995년~현재 강원대학교 인문사회과학대학 교양학부 교수
관심분야 : 해석학, 데이터베이스, 정보통신 및 보안
E-mail : jrlee@kangwon.ac.kr