N-grams를 사용한 CNN 기반의 악성코드탐지 기법 연구

허정원, 문봉교
동국대학교 컴퓨터공학과
hacel@dongguk.edu, bkmoon@dgu.edu

Malware Detection Based on CNN with N-grams

Jeong-Won Her, Bong-Kyo Moon
Dept. of Computer Science Engineering, Dongguk University

요 약

본 논문에서는 악성코드탐지 기법으로 n-grams를 사용한 특징 추출을 통해 이미지 인식 분야에서 널리 쓰이는 Convolutional Neural Network로 학습하는 프레임워크를 제안한다. 윈도우즈 실행 파일의 PE 포맷에서 특징을 추출하여 6-grams 확률을 구하고 grayscale 을 통해 이미지로 변환한다. 이것을 기존에 연구된 탐지방법과 비교하여 우수함을 보인다. 학습에 사용된 데이터는 총 55,000개로 5-folds 교차검증을 하였으며 예측 정확도는 98.87%였다.

1. 서론

오늘날에는 매 순간 새로운 악성코드들이 등장하고 있다. 하지만 현재 사용되는 탐지 알고리즘의 대부분은 악성코드의 해시값과 같은 시그니처를 기반으로 한다. 새롭게 변형된 악성코드는 기존 파일의 시그니처를 비교해 검출할 수 없다. 따라서 기존 방법의 단점을 해결하기 위해 딥러닝과 통계 기반의 탐지 기법이 주목받고 있다.

2. 관련 연구

악성코드의 딥러닝을 위한 특징 추출 기법으로 문자열 추출, import tables, byte n-gram, opcode, byte entropy 등을 살펴보는 방법들이 있다. Saxe et al..[1]은 byte entropy, import tables, 문자열 추출, PE metadata 네 가지 특징의 조합에 대한 Deep Neural Network (DNN) 탐지율을 비교한다. 다양한 특징 추출방법을 제시하지만, n-grams 방법에 대한 제시가 없으며 DNN만 사용하여 특징 인식에 더 뛰어난 성능을 보이는 CNN에 대한 논의가 부족하다.

특징 추출을 할 범위를 정하는 것도 중요하다. 실행 파일의 크기는 다양하다. 실행 파일의 모든 내용에 대한 특징 검사는 매우 느리고 불필요한 정보를 검토하게 될 가능성이 크다. Stolfo et al.[2]은 파일의 시작과 끝만을 n-gramming 하여 정보를 추출한다. 본 연구도 유용한 부분만을 살펴보고, 특정 길이를 얻기 위해 PE 헤더만을 특징 추출에 사용한다.

N-grams가 악성코드 분류에 어떤 성능을 지니는지에 대해 Raff et al.[3]은 Elastic-Net과 LR를 사용해 조사한다. N-grams를 파일에 효과적으로 적용하는 방법을 제시하지만, Elastic-Net과 LR에 대한 성능만 제시되는 한계가 있다.

Raff et al..[4]에선 다양한 학습 기법의 악성코드 탐지 성능을 비교한다. Raff도 PE 포맷을 특징으로 추출해 악성코드를 탐지한다. 이것에 대해 Extra Random Trees(ET), Random Forests(RF), Logistic Regression(LR), Fully Connected Neural Network(FC), Long Short-Term Memory(LSTM) 다섯 가지 학습 기법을 적용하고 탐지 성능을 비교한다. 다양한 탐지 기법에 소개가 있지만 역시 CNN에 대한 논의가 부족하다.

본 연구에서는 n-grams 특징 추출과 CNN 학습 기법을 동시에 사용해 기존 연구 결과와 탐지 정확도를 비교한다. 결과로 얻은 탐지 정확도는 98.87%로 기존 연구 결과보다 우수한 악성코드탐지에 기법임을 보인다.

3. 제안 모델

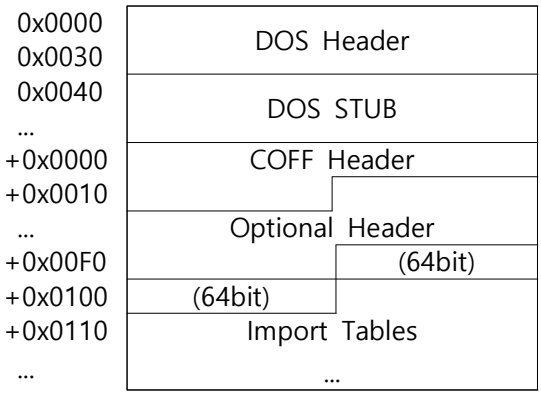
제안하는 탐지 프레임워크는 (그림 1)과 같이 학습한다. 첫째로 파일 특징 추출로 PE 포맷 추출과 n-grams 제작 단계로 나뉜다. 둘째로 데이터의 이미지화 단계이다. 이 단계에서 PE 포맷의 byte는 문맥에서 등장할 확률에 따라 grayscale 하여 이미지로 변환된다. 셋째로 CNN의 학습이다. CNN은 filter를 사용하여 데이터의 특징을 추출한다. 따라서 사소한 내용의 차이(회전, 왜곡, 변형)가 있어도 특징을 검출하기 때문에 악성코드 변종에 대한 감지가 수월할 것으로 생각된다. 다음 단락에서 각각의 단계를 설명한다.

1. 특징 추출
PE 포맷 추출
n-grams 제작
2. 이미지화
Gray Scale
3. Convolutional Neural Network
Input layer 21x16x1
Convolution Layer, ReLU units (21x16x32)
2x2 Max Pooling (11x8x32)
Convolution Layer, ReLU units (11x8x64)
2x2 Max Pooling (6x4x64)
Flatten (1536)
Output layer, softmax units

(그림 1) 악성코드탐지 프레임워크

가. 특징 추출

PE 포맷은 윈도우즈 OS에 실행 파일 포맷으로 실행에 필요한 정보, 메모리 로드 위치, 사용하는 라이브러리, 실행시간에 결정되는 정보를 담은 구조체이다. PE 포맷은 (그림 2)와 같이 크게 DOS header, Common Object File Format(COFF), optional header, import tables로 나뉜다. 본 연구는 PE 포맷에서 DOS header, COFF, Optional header 부분을 추출해 총 328 bytes 배열을 얻게 된다. 이것은 실행 파일의 핵심적인 특징이며, 적은 분량이고, 일정한 크기를 가진다. 이 특징들은 이후 신경망에 적용하기 쉽게 해준다.



(그림 2) PE 포맷의 구조

n-grams는 통계에 기반해 나타날 단어를 추론한다. n-grams는 앞서 등장한 n-1개 단어 h에 대해 n번째 단어 w가 나타날 확률을 (식 1)과 같이 계산한다.

$$P(w|h) = \frac{C(h+w)}{C(h)} \qquad \text{(식 1)}$$

n-grams로 byte의 문맥을 학습시킬 수 있다. 이 방법은 대상에 대한 사전 지식이 필요 없이 특징의 자동적인 추출이 가능하다는 장점이 있다. 본 연구에서는 6-grams로 13500개의 정상코드의 문맥을 학습시켜 byte의 등장 확률을 구한다.

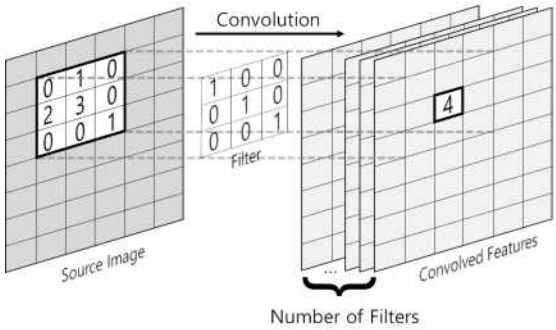
나. 이미지화

앞선 방법으로 얻은 byte 등장 확률을 각각의 이미지 픽셀로 변환하게 된다. 정상코드의 문맥에 가까울수록 높은 확률값을 가지고 처음 등장하는 문맥, 즉 악성코드이면 낮은 확률값을 가진다. 0을 검은색 1을 흰색으로 하여 0~1 범위의 확률값을 grayscale 하여 이미지로 바꿔주게 된다. 따라서 문맥에 어긋나는 부분일수록 검은색으로 나타나게 된다. 328 bytes 배열은 위치를 알기 쉽게 16개의 행을 가지도록 한다. 328/16 = 20.5이므로 8개의 0 값 padding을 더해 (그림 3)과 같은 16*21 (=336) 크기의 이미지를 얻는다.

(그림 3) 16x21 Grayscale 예시

다. Convolutional Neural Network

Convolution은 (그림 4)와 같이 특정한 필터값으로 입력 데이터에 대한 합성곱 연산을 수행함을 말한다. 따라서 필터에 따라 이미지 일부분이 강조되고 이는 특징을 추출하는 기능을 한다. 본 연구에서는 2개의 convolution layer를 사용하며 각각 32, 64개의 3x3 필터 합성곱 연산을 한다.



(그림 4) Convolution

4. 실험 및 분석

사용된 데이터는 KISA에서 수집한 윈도우즈 실행 파일 정상코드 13,500개 악성코드 41,500개 총 55,000개의 데이터를 사용하였다. 이 데이터를 사용할 때 5-fold cross validation 기법을 적용하여 검증하였다. 실험 환경은 윈도우즈 10(64bit) 운영체제에서 Tensorflow backend Keras로 실험하였다. 상세 실험 환경은 <표 1>에 기술하였다.

제안하는 모델과 비교하는 모델은 Raff et al.[4]의 방법들이다. Raff는 악성코드탐지를 위해 PE 포맷에서 328byte를 추출해 FC, LSTM, ET, RT, LR 3-grams를 사용한다. <표 2>는 Raff가 industry partner에게 받은 데이터(Group B)를 사용해 얻은 모델 정확도와 제안하는 모델의 정확도를 비교한다.

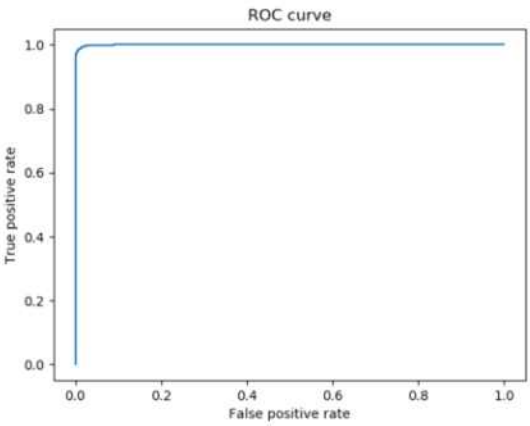
앞서 얻은 모델의 정확도(Accuracy)는 98.87%이다. 하지만 모델성능을 정확도로만 판별하는 것은 위험하다. 이 맹점을 해결하기 위해 Receiver Operating Characteristic(ROC)을 사용한다. (그림 5)는 6-grams로 전처리를 한 ROC 그래프이고 (그림 6)은 Raff의 모델에 대한 ROC 그래프이다. 이것을 수치로 정확히 판별하기 위해 Area Under the Curve(AUC) 값을 구한다. 이 값은 100%의 정확도를 가지는 완벽한 모델에서 1 값을 가진다. <표 2>에서 제안 모델과 Raff의 모델에 대한 AUC 값을 비교하였다. 6-grams로 전처리를 한 제안 모델에 대해 AUC는 0.999의 값을 가져 단순히 신경망을 적용한 것보다 더 좋은 모델을 생성했음을 알 수 있다.

이름	내용
OS	Windows 10.0.18362
CPU	i5-8250U 3.40GHz
RAM	8GB
Tensorflow	2.0.0
Keras	2.3.1

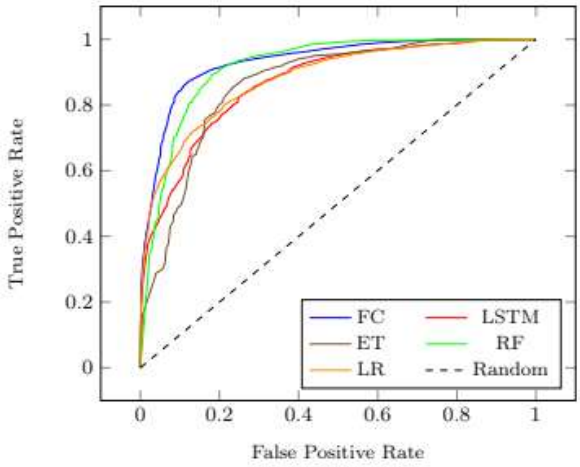
<표 1> 실험 환경

	Accuracy(%)	AUC(%)
FC	83.7	91.4
LSTM	77.5	86.7
ET	80.7	86.1
RT	82.3	91.2
LR 3-grams	77.8	87.3
CNN 6-grams	98.9	99.9

<표 2> 결과 비교



(그림 5) 6-grams 전처리 CNN의 ROC 그래프



(그림 6) ROC ploat for all models on Group B test data[4].

4. 결론

본 연구는 배경 지식 없이 적절한 특징 추출 기법과 빅데이터만으로도 효과적인 학습을 수행할 수 있음을 보여준다. 그러나 PE 포맷의 특징 추출 기법은 windows 운영체제에서만 적용할 수 있다. 이점은 다른 운영체제의 실행 파일에 적용할 수 없다는 명확한 한계점을 지닌다. 이를 개선하기 위해 악성코드 분류와 통합 플랫폼을 만들 수 있을 것이다. Islam et al.[5]은 문자열을 추출해 효과적으로 악성코드 분류법을 제안한다. 이 논문에서 보이는 분류 정확도는 98.8%로 상당한 정확도로 분류 가능함을 보인다. 이러한 방법을 응용해 효과적으로 파일의 실행 가능한 운영체제나 포맷, 유형에 따라 분류하고, 각각에 적절한 특징 추출 기법을 연구한다면 변형 악성코드에도 빠르게 대처할 수 있을 것이다.

본 연구는 한국인터넷진흥원(KISA)에서 운영하는 정보보호 R&D 데이터셋 [대용량 정상/악성파일 I, 대용량 정상/악성파일 II, 대용량 정상/악성파일 III]을 활용하여 작성되었음

참고문헌

[1] Saxe, Joshua, and Konstantin Berlin. "Deep neural network based malware detection using two dimensional binary program features." 2015 10th International Conference on Malicious and Unwanted Software (MALWARE). IEEE, 2015.

[2] Stolfo, Salvatore J., Ke Wang, and Wei-Jen Li. "Towards stealthy malware detection." Malware Detection. Springer, Boston, MA, 2007. 231-249.

[3] Raff, Edward, et al. "An investigation of byte n-gram features for malware classification." Journal of Computer Virology and Hacking Techniques 14.1 (2018): 1-20.

[4] Raff, Edward, Jared Sylvester, and Charles Nicholas. "Learning the pe header, malware detection with minimal domain knowledge." Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security. ACM, 2017.

[5] Islam, Rafiqul, et al. "Classification of malware based on string and function feature selection." 2010 Second Cybercrime and Trustworthy Computing Workshop. IEEE, 2010.