DBpia

JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 19, No. 6, pp. 1177-1183, Jun. 2018

딥러닝 기반의 R-CNN을 이용한 악성코드 탐지 기법

조 영 복
대전대학교 정보보안학과 교수

The Malware Detection Using Deep Learning based R-CNN

Young-Bok Cho
Department of Computer & Information Security, Daejeon University, 62 Daehak-ro, Dong-gu, Daejeon 34520, Korea

[요 약]

최근 기계학습의 발달로 인공지능을 구현하는 머신러닝과 딥러닝 같은 기술이 많은 관심을 받고 있다. 본 논문에서는 딥러닝 기반의 R-CNN을 이용한 바이너리 악성코드를 이미지화 하고 이미지에서 특징을 추출해 패밀리를 분류한다. 본 논문에서는 딥러닝에서 두 단계를 이용해 악성코드를 CNN을 이용해 이미지화하고,악성코드의 패밀리가 갖는 특징을 R-CNN을 이용해 분류함으로 악성코드를 이미지화하여 특징을 분류하고 패밀리를 분류한 후 악성코드의 진화를 자동 분류한다. 제안 기법은 검출율이 93.4%로 우수한 탐지 성능을 보였고 정확도는 98.6%로 매우 높은 성능을 보였다. 또한 악성코드를 이미지화 하는 CNN 처리속도가 23.3ms, 하나의 샘플을 분류하기 위해서 R-CNN처리 속도는 4ms로 비교적 빠르게 악성코드를 판별하고 분류가 가능함을 실험을 통해 증명하였다.

[Abstract]

Recent developments in machine learning have attracted a lot of attention for techniques such as machine learning and deep learning that implement artificial intelligence. In this paper, binary malicious code using deep learning based R-CNN is imaged and the feature is extracted from the image to classify the family. In this paper, two steps are used in deep learning to image malicious code using CNN. And classify the characteristics of the family of malicious codes using R-CNN. Generate malicious code as an image, extract features, classify the family, and automatically classify the evolution of malicious code. The detection rate of the proposed method is 93.4% and the accuracy is 98.6%. In addition, the CNN processing speed for image processing of malicious code is 23.3 ms, and the R-CNN processing speed is 4ms to classify one sample.

색인어 : 악성코드 분석, 딥러닝, R-CNN, 특징 추출, 이미지 프로세싱

Key word : Malware, Deep learning, Regions with CNN, Image Processing

http://dx.doi.org/10.9728/dcs.2018.19.6.1177

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 20 May 2018; **Revised** 24 June 2018
Accepted 25 June 2018

***Corresponding Author; Young-Bok Cho**
Tel: ※ 개인정보 표시제한
E-mail: lotuscho73@gmail.com

www.dbpia.co.kr

Ⅰ. 서 론

최근 기계학습의 발달로 인공지능을 구현하는 머신러닝과 딥러닝 같은 기술이 많은 관심을 받고 있다[1,2]. 특히 강화학습과 딥러닝을 이용한 구글 딥 마인드의 알파고가 국내에서는 유명하고 이미지, 자연어 처리에 이어 음성처리 분야에서도 인공지능 기술이 놀라운 성능을 보여주고 있다. 또한 악성코드 탐지나 분류 분야에서도 머신러닝과 딥러닝 같은 인공지능 기술이 적용되어 악성코드의 고유한 패턴이나 시그니처를 학습하고 시그니처 기반의 악성코드 탐지에 적극 활용되고 있다. 반면 사용자의 시스템을 파괴하거나 정보를 탈취할 목적으로 만들어진 악성코드(malware)는 악의적인 활동을 수행하는 웜, 바이러스, 트로이목마 등의 공격이 있다. 그러나 하드웨어 및 소프트웨어의 발달로 자동화 도구가 증가하면서 악성코드의 급증에도 한몫하고 있는 추세이다[2,3,4,5].

최근에는 기존 보안 솔루션이 시그니처를 기반으로 알려진 위협을 식별한다는 사실을 공격자들도 인지하고 악성코드의 모양을 변형해 변종을 생성한다. 특히 이런 종류의 악성 코드는 대부분 악성코드 생성시 이미 존재하는 기존 악성 코드를 재사용하거나 악성코드 제작 도구의 발전 등으로 변종 악성코드가 급격히 증가하고 있다. 또한 자동화 도구로 제작된 악성코드로 암호화 코드부분 EPO(entry point obscuring)만 변경함으로 겉모습만 변경해 백신탐지를 회피하고 있다. 따라서 최근 급격히 발전하고 있는 딥러닝을 정보보안 분야에 적용시켜 악성코드를 탐지하거나 악성코드를 분류함으로 사이버 공격에 대응하고 , 딥러닝을 멀웨어 악성코드를 판별함으로 랜섬웨어 등 공격에 대비한다. 딥러닝 기반의 악성코드 탐지는 두 부분으로 정의한다. 첫째는 악성코드로부터 특징 패턴 정보를 추출하는 것으로 악성코드가 실행되는 경우 자동으로 API를 추출하거나 정적분석을 통해 opcode와 같은 어셈블리코드를 확보할 수 있다. 또는 악성코드를 이미지 파일로 간주하여 특징데이터를 추출하기도 한다. 두 번째는 이전에 추출된 특징 데이터를 기반으로 딥 러닝 모델을 활용해 트레이닝하게 되는데 이것은 여러 악성 코드의 특징데이터를 사용해 딥러닝 모델을 트레이닝 한 후에 탐지 대상인 악성코드를 입력으로 악성유무를 판단하고 악성코드일 경우 악성코드의 패밀리를 분류하게 된다. 그러나 최근 악성코드 작성자는 패킹 기법을 사용하여 원래의 바이트 순서를 무작위로 보이는 데이터로 변환함으로써 시그니처 기반 악성코드 탐지 기법을 무력화시키고 있다. 최근 지속적인 상업용 백신이 개발되고 있음에도 불구하고 악성코드에 의한 피해는 지속적으로 증가하고 있다. 그러나 악성코드 가운데 감염시킨 PC를 암호화하여 가용성을 마비시키는 랜섬웨어 같은 경우 불특정 다수를 대상으로 공격하며 지능화되어 가고 있다. 따라서 악성코드를 정확하게 탐지하고 대응할 방법이 요구된다.

본 논문에서는 안정화된 딥러닝 기술을 이용해 이미 알려진 악성코드 패턴을 딥러닝 모델을 이용해 학습하고 악성코드를 판별한다. 이때 악성코드가 가지고 있는 시그니처를 기반으로 악성코드의 일부분이 변경되어 생성된 악성코드의 경우 시그니처를 이미지가 생성되었을 때 원본과 큰 차이점을 보이지 않는다는 특성을 이용해 기존 악성코드 탐지기법에서 문제점을 해결한다. 이 논문에서는 CNN을 이용한 딥러닝 기반의 악성코드 탐지를 위해 기본 단계와 딥러닝 단계를 설계하고 바이너리 악성코드를 분석한 후 이미지화 한다. 이미지화 된 바이너리 코를 M-CNN을 이용해 악성코드를 판별하고 강화학습을 진행한다. 악성코드 학습은 MS data set(10.868) 을 학습용 악성코드로 활용해 악성코드 탐지 성능이 평균 92%을 유지함을 실험을 통해 확인하였다.

Ⅱ. 관련연구

2-1 R-CNN

CNN은 이미지의 특징을 추출하고 이미지의 이동이나 왜곡에 대한 항상성을 유지하기 위한 과정을 거쳐 해당 이미지를 구분하기 위한 분류작업 한다[4,6,7,8]. 반면 R-CNN(Regions with CNN)은 이미지 분류시 CNN 과 이미지에서 물체가 존재할 영역을 제안해주는 region proposal 알고리즘을 연결하고 높은 성능의 object detection을 수행할 수 있음을 제시해 주는 방법으로 동작과정은 다음과 같다.

(1) 이미지를 입력받는다.
(2) 이미지로부터 selective search를 이용해 region proposal을 추출한다.
(3) 각 region proposal 영역을 이미지로부터 잘라내고(cropping) 동일한 크기로 만들어(warping) CNN을 이용해 특징(feature)을 추출한다.
(4) 각 region proposal feature에 대한 classification을 수행한다.

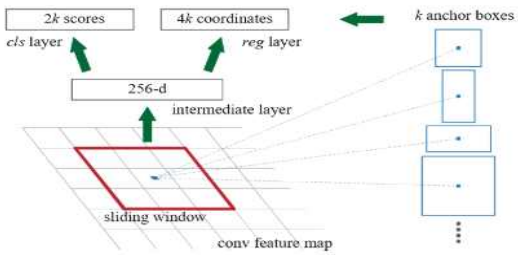

그림 1. R-CNN 구조
Fig. 1. The R-CNN Structure

그림 1은 R-CNN구조로 egion proposals 을 하기위해, 각 sliding-window 위치에서 k-reference boxes(k anchors)에 relative한 방법으로 최대 k개를 proposal 할 수 있다. 각 sliding-window 는 lower-dimensional feature(intermediate layer)로 매핑되고, 이 feature 들은 box-regression layer(reg) 와 box-classification layer(cls)에 fed 된다. 그러나 우리가 잘 알고

www.dbpia.co.kr

있듯이 R-CNN은 속도면에서 다소 느리다는 문제를 가지고 있다. 이유는 R-CNN의 Training을 느리게 만드는 가장 주된 이유로 3단계 파이프라인의 사용이라고 할 수 있다. 첫 번째로 R-CNN은 Train을 위해 CNN을 사용할 때 이미 알려진 성능 좋은 네트워크 모델의 가중치를 가져와 fine-tuning을 실행한다. 두 번째로 fine-tune된 CNN에 맞게 SVM을 fitting하고 마지막으로 bounding box regressor를 학습시킨다. 이 3단계의 순차적으로 발생하는 학습 시간이 비교적 오래 걸리기 때문이다.

2-2 악성코드 탐지와 머신러닝 기반의 악성코드 탐지

악성코드는 운영체제의 발전과 더불어 꾸준히 발전하고 있다. 악성코드의 분석은 일반적으로 이진 서명 추출을 이용한 시그니처 분석 방법으로 크게 두 가지 정적과 동적 분석으로 구분된다.

정적분석은 시그니처 기반 분석인 웹 사이트에 포함된 비정상 콘텐츠를 분석하는 방법과 웹 사이트의 메타정보를 기반으로 분석하여 잘 알려진 악성 메타정보와 비교하는 방법이 있다. 또한 코드를 분해하고 실행 파일에서 제어 흐름을 탐색해 악성 패턴을 찾아낸다. 또한 정적 분석의 경우 코드 난독화라는 문제점을 가지고 있지만 가장 완벽한 적용 범위를 제공하고, 실행 파일은 분석하기 전에 압축을 풀고 난후에 해독한다. 또한 분석이 다루기 힘든 복잡성 문제로 인해 방해 받을 수 있다[11,12].

동적 분석은 주로 가상 환경에서 바이너리 코드를 실행함으로 동작된다. 또한 실행 추적을 기반으로 실행 파일을 특성화하는 동작 보고서가 생성된다. 그러나 동적 분석은 실행파일을 압축해제하거나 해독할 필요가 없어 보다 효율적이지만, 시간이 많이 걸리고 자원이 많이 소모되므로 당연히 확장성 측면에서는 문제가 발생한다. 또한 환경이 악성코드의 트리거 조건을 충족시키지 못하기 때문에 일부 악의적인 동작은 탐지되지 못할 수도 있다는 문제점을 갖는다.[11,12]

머신러닝 기반의 악성코드 탐지란 정상파일 및 악성코드로 학습 모델을 학습시킨 후, 학습된 모델로 의심스러운 파일의 악성 여부를 탐지하는 것으로 학습단계에서 학습 모델을 파일들의 특징정보(문자열, 명령어, 바이트 정보, API 호출 기록 등)와 레이블(정상/악성코드)로 학습을 시킴으로써 모델이 악성코드 탐지에 최적화 되도록 한다.

탐지 단계에서는 학습된 모델을 이용하여 입력된 파일이 정상 파일인지 악성코드인지를 구별한다. 탐지한 악성코드는 악성코드 바이트 코드를 그레이스케일 이미지로 시각화 한다. 대부분 많은 악성코드 패밀리 군과 변종 패밀리 군의 경우 동일한 패밀리에 속하는 이미지가 레이아웃과 텍스처면에서 매우 유사한 결과를 나타낸다. 주어진 바이너리는 부호 없는 8비트 정수 2D 배열 [0.255]범위로 표현될 수 있다.

그림 2는 일반적인 트로이 Dontovo의 이미지 시각화 단계를 나타낸 것이다. 악성코드를 입력받아 8bit 바이너리를 생성하고 이것을 부호 없는 정수 2D 그레이스케일 배열을 생성한다. 그리고 2D 배열을 출력하면 이미지시각화가 끝나게 된다.

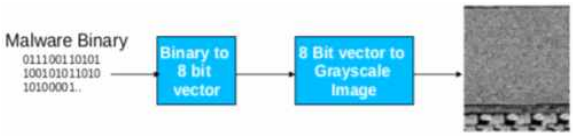

그림 2. 이미지 시각화 단계
Fig. 2. Image Visualization Step

Ⅲ. CNN을 이용한 딥러닝 기반의 악성코드 탐지 기법

3-1 시스템 구성

논문에서는 CNN을 이용해 딥러닝 기반의 악성코드 탐지를 위한 시스템 구성도를 그림 3과 같이 제안한다. 제안 논문은 악성코드 실행 없이 파일 형식이나 바이트, 추출 텍스트 등 수집 가능한 정보를 학습하여 악성코드를 탐지하고, 악성코드를 실행한 후 발생한 이벤트나 프로세스 행위 기록 등 수집된 정보를 기반으로 딥러닝 모델을 학습시켜 악성코드를 탐지하게 된다. 악성코드가 입력 시 특성을 추출하고 이미지화 한다.

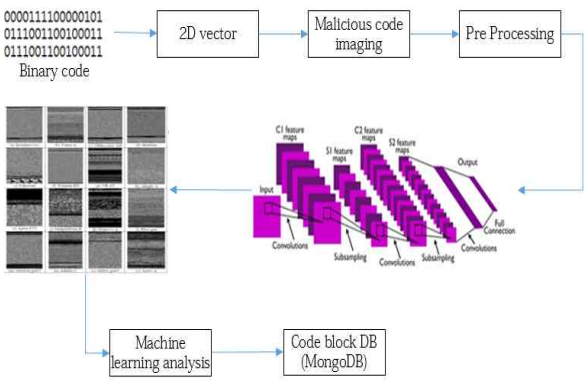

그림 3. 시스템 전체 구조도
Fig. 3. System Overall Structure

그림 3과 같이 제안 시스템에서 학습을 CNN 기반의 악성코드 탐지 단계는 악성코드의 정보에 따라 기본 탐지, 심화 탐지, 희귀 악성코드 학습 단계로 구성된다.

- 기본탐지는 실행파일의 기본적인 특징정보(Lightweight Features)로 유사성 해싱(Similarity Hashing) 함수를 학습시킨 후 정상 파일과 악성코드를 분류한다. 이때 유사성 해싱 함수는 특징정보의 해싱 값에 따라 악성코드를 적절한 버킷에 분류 하고 만일, 버킷이 정상파일과 악성코드가 혼합된 경우 심화 탐지 단계를 수행하게 된다.
- 심화 탐지는 실행파일에서 추출 가능한 모든 특징정보(Heavy Features)로 유사성 해싱 함수를 학습시킨 후 앙상블 알고리즘을 통해 정상파일과 악성코드를 분류한다. 만일 분류 결

www.dbpia.co.kr

과에 정상파일과 악성코드가 혼합된 경우 아래와 같은 희귀 악성코드 학습 단계를 수행하게 된다. 심화 학습을 위해서 은닉층에서는 ReLU 함수를 적용하는데 이데 액티베이션 $b^i_{(x,y)}$는 식 1과 같이 사용한다.

$$b^i_{(x,y)} = a^i_{(x,y)} \Big/ \left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a^j_{(x,y)})^2 \right)^{\beta} \quad (1)$$

식 1에서 n개의 합은 근접한 커널 맵의 부분 공간 합이고 N은 레이의 모든 커널 수를 의미한다. $a^i_{(x,y)}$는 i번째 커널을 (x,y)위치에 적용한 특징의 액티베이션을 의미한다.

– 희귀 악성코드는 탐지할 수 있도록 딥러닝 모델(CNN)로 악성코드의 특징정보를 추출하는 단계이다. 추출된 특징정보는 희귀 악성코드 및 이와 유사한 악성코드를 탐지하는 딥러닝 모델(R-CNN)에 사용 된다.

3-2 R-CNN을 이용한 악성코드 이미지화

제안 시스템은 악성코드 바이너리를 입력받아 R-CNN 을 이용해 학습하고 악성코드를 분류한다. 악성코드를 학습할 때 MS data set 10,868개 학습용 악성 코드 바이트를 이용한다. 식 2는 학습을 위한 강화학습에 사용될 테스트 셋을 나타낸다.

$$\delta = X^* - X_{test} \quad (2)$$
$$L(p_i,t_i) = \frac{1}{N_{ds}} \sum_i L_{ds}(p_i,p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i,t_i^*)$$

식 2에서 X_{test}은 원본 테스트 셋을 의미하고 X^*는 강화학습에 사용할 테스트 셋을 의미 한다. loss 함수로 $L(p_i,t_i)$에서 p_i 는 anchor의 예상확률, p^*는 이미지의 실제 값인 ground truth label로 1이면 anchor이 positive, 0이면 negative를 나타낸다. $L(p_i,t_i)$는 람다(lambda)로 N_{ds}와 N_{reg}의 차이로 발생하는 불균형을 방지하기 위해 사용한다. t_i는 predicted의 bounding box를 나타내고, t^*는 groud-truth box를 나타낸다.

그림 4는 R-CNN에서 악성코드 탐지를 위한 트레이닝 데이터 셋을 위한 코드를 파이썬으로 작성하고 학습 과정이 끝나면 result 폴더에 학습결과 파일을 생성한다.

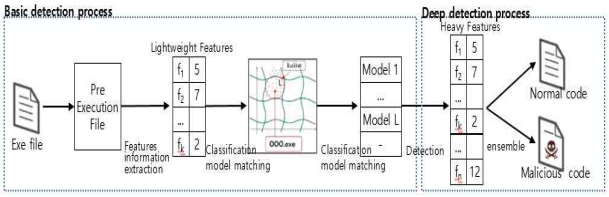

그림 4. 악성코드를 위한 R-CNN 모델
Fig. 4. R-Cnn Model For Malware

그림 5은 악성코드와 파일 단위별 분류의 정확도와 각종 수행간의 통계사항을 정리해 Numpy, Tensorflow의 라이브러리를 이용해 R-CNN을 설계하였다.

```
with tf.Session() as sess:
    saver=tf.train.Saver()
    if tf.gfile.Exists(FLAGS.result_dir + '/result.dat'):
        saver.restore(sess, FLAGS.result_dir '/result.data')
    else :
        init=tf.initinlize_all_variables()
        sess.run(init)
    print 'Total sample : %d' %total_sample_count
    print 'Precision %.5f'%(acc/total_sample_count)
    print 'Running time: %f secs'%((delta.seconds))

def test():
    fd=tf.placeholder(tf.float32)
    image.labels=td.batch_data('test',FLAGS.batch_size)
    logits = cnn.inference(images.fd)
    loss=cnn.loss(logits.labels)
    train=cnn.train(loss)

    with tf.Session() as sess
        saver=tf.train.Saver()
```

그림 5. 텐서플로우 구동을 위한 학습 알고리즘
Fig. 5. Learning Algorithm For Tensorflow Driving

악성코드를 시각적으로 이미지화 하는 경우 악성코드의 일반적 특성으로 이미지의 유사성으로 인한 이미지 구분이 또 다른 문제점이 될 수 있다.

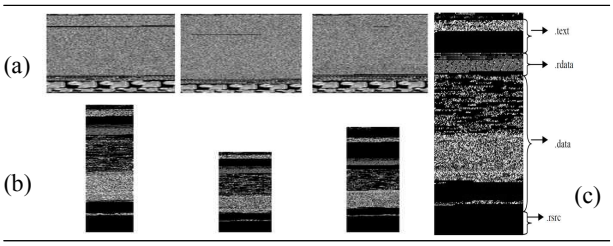

그림 6. 악성코드 이미지의 그룹별 구분
Fig. 6. Grouping Of Malware Images By Group.

그림 6의 첫 행(a)은 Fakerean,두 번째(b)행은 Dontovo의 3종 악성코드 인스턴스 이미지를 나타낸 것이다. 제안 논문에서는 텐서플로우 딥러닝 알고리즘을 이용해 알려진 멀웨어 악성코드를 학습시키고 악성코드를 분류하는 레이어를 구성하였다. 그림 6의 (c)에서 .text 섹션에는 실행가능코드가 들어있다. 그림 6에서 .text 섹션의 첫 번째 부분은 텍스처가 세분화된 코드를 포함하고 있음을 알 수 있다. 나머지 부분은 0으로 채워지며 이 부분의 끝에는 역시 0으로 채워진다. 다음 .data 섹션에는 초기화되지 않은 코드 (검은 색 패치)와 초기화 된 데이터 (세밀한 텍스처)가 모두 들어 있다. 마지막 세션은 모듈의 모든 리소스를 포함하는 .rsrc 섹션이다. 여기에는 응용 프로그램이 사용할 수 있는 아이콘도 포함될 수 있다. 악성코드를 분류하기 위해 컨볼루션 레이어는 여러 개의 필터를 이용해 악성코드 이미지에 대한 연산을 수행한다.

그림 7은 R-CNN을 위한 컨볼루션 레이는 이미지를 2*2의 필터를 이용해 이미지를 중첩되도록 상호 비교연산을 수행하면서 특징을 추출하게 된다. 또한 폴링 레이어(pooling layer)는

www.dbpia.co.kr

이미지를 겹치지 않는 정사각형 형태로 나누어 서브 영역에 최댓값을 출력하는 max pooling를 사용하는데 폴링 레이어는 일반적으로 계산량을 줄이고 입력데이터의 변화에 폴링 계층의 크기 값에 큰 영향을 받지 않는 다는 장점을 갖는다.

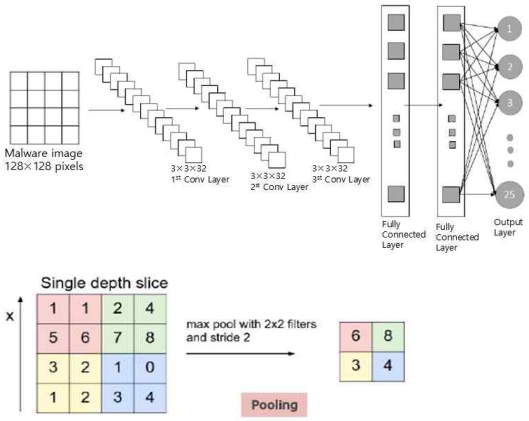

그림 7. R-CNN을 위한 컨볼루션 연산
Fig. 7. Convolution Calculation For R-CNN

FullConnect 레이어는 추상화된 이미지를 모두 연결하여 최종 패밀리를 판단하는 기능을 수행한다. 또한 학습시 learning rate는 학습에 사용되는 검색 비율의 크기로 너무 작으면 학습 단계가 너무 작아 많은 시간을 학습해도 학습이 이루어지지 않아 정확도가 매우 떨어지게 되는 결과를 가져온다. 반면 너무 작으면 오버피팅 문제가 발생되어 본 논문에서는 알고리즘 특성상 decay값은 시스템 성능에 큰 영향을 미지치 못하기 때문에 weight decay를 0.00004로 고정하고 실험한다.

Ⅳ. 실험 및 결과

4-1 실험환경

본 논문에서 악성코드를 위한 R-CNN을 이용한 딥러닝 기반의 탐지 알고리즘을 실험하기 위해 Microsoft에서 제공하는 data set 10,868개 중 3,260개를 학습 데이터로 사용하고, 6,520개를 테스트 데이터로 사용했다. 1,088개의 데이터는 검증용으로 사용한다. 이번 실험에서 우리는 데이터 셋이 가지고 있는 9종류의 악성코드를 분류하고 정확도를 판별한다. 정확도 판별을 위해 검출율(Recall)은 입력데이터를 성공적으로 분류한 비율, 미검출율(missing rate)은 입력 데이터를 판별하지 못하는 비율, 정확도(precision)은 정 분류로 9종을 정확히 분류한 비율, 오검출율(false alarm rate0은 분류된 악성코드 중 오분류된 비율을 의미한다. TP(True Positive)는 정상데이터를 정상으로 탐지한 결과, FN(False Negative)은 정상데이터를 악성코드로 탐지한 경우, FP(False Positive) 악성코드를 정상데이터로 분류한 경우, TN(True Negative)은 악성을 악성이라고 분류하는 경우를 의미한다. 제안 알고리즘을 평가한 결과 표 1과 같다.

표 1. R-CNN을 이용한 악성코드 분류 성능분석
Table 1. Performance Analysis Of Malware Classification Using R-CNN

Accuracy	Missing rate	Precision	Recall	False alarm
93.8%	5.2%	98.6%	93.4%	0.8%

표 1에서와 같이 검출율이 93.4%로 우수한 탐지 성능을 보였으며 정확도 98.6%로 악성코드 분류를 정확히 분류함을 확인할 수 있었다

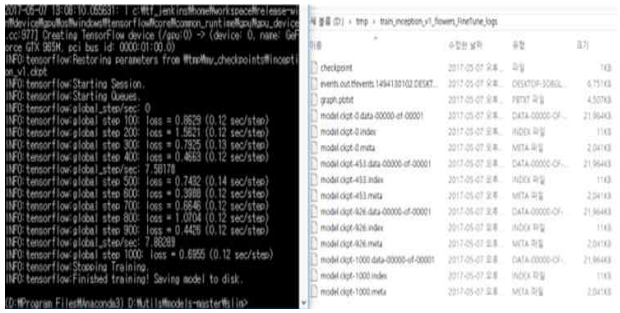

그림 8 악성 코드 패밀리 분류 결과
Fig. 8. Malware Code Family Classification Result

그림 8에서는 악성코드를 이미지화 하고 패밀리를 분류한 결과이다. 악성코드의 패밀리는 악성코드 유형을 분류한 것으로 테스트 데이터에는 9종의 패밀리를 포함하고 있었고 제안 모델을 이용한 경우 9종의 패밀리를 96.4%의 정확도로 분류해 냈음을 알 수 있다. 또한 제안 모델에서 하나의 악성코드를 이미지화 하고 악성코드 이미지로부터 특징점을 추출하는 과정에서 평균 23.3ms가 소요되었다. 그러나 하나의 샘플을 분류하기위해 학습된 일부 부분을 비교해야 하는 R-CNN의 특성상 4ms의 시간이 소요됨을 확인했다. 따라서 제안 알고리즘의 경우 R-CNN을 사용해도 일반 이미지 데이터와 달라 특별히 속도에 문제점을 보이지 않음을 실험을 통해 증명되었다.

Ⅴ. 결 론

본 논문에서는 최근 급격히 발전하고 있는 딥러닝을 정보보안 분야에 적용시켜 악성코드를 탐지하거나 악성코드를 분류함으로 사이버 공격에 대응하고자 딥러닝 기반의 R-CNN을 이용한 악성코드 탐지 기법을 제안하였다. 악성코드의 특성상 이진바이너리 코드로 구성된 악성코드는 특정 패밀리를 가지고 있어 패밀리끼리 특징을 가지고 있다. 관련 연구에서도 이미지 기반의 악성코드의 패밀리 분류 방법은 연구되고 있지만 새로운 악성코드에 대한 학습이나 특징점 추출을 위한 과정이 추가적으로 요구되었다. 또한 악성코드의 샘플수가 증가할수록

www.dbpia.co.kr

패밀리 분류에 소요되는 시간이 길어진다는 단점을 갖는다. 그러나 본 논문에서는 이미 알려진 악성코드 패밀리를 기반으로 새롭게 생성된 악성코드 학습시킬 수 있고, 스스로 특징점을 추출 할 수 있다는 점에서 차이점을 갖는다. 또한 CNN과 R-CNN을 이용해 악성코드를 이미지화 하는 단계와 패밀리 분류 단계의 딥러닝 알고리즘을 차별화함으로 처리 시간을 단축할 수 있었다.

본 논문에서는 기존 연구의 문제점을 파악하고 딥러닝에서 두 단계를 이용해 악성코드를 CNN을 이용해 이미지화하고 ,악성코드의 패밀리가 갖는 특징을 R-CNN을 이용해 분류함으로 악성코드를 이미지화하여 특징을 분류하고 패밀리를 분류한 후 악성코드의 진화를 자동 분류한다. 제안 기법은 검출율이 93.4%로 우수한 탐지 성능을 보였고 정확도는 98.6%로 매우 높은 성능을 보였다. 또한 악성코드를 이미지화 하는 CNN 처리속도가 23.3ms, 하나의 샘플을 분류하기 위해서 R-CNN처리 속도는 4ms로 비교적 빠르게 악성코드를 판별하고 분류할 수 있었다. 딥러닝의 특징상 향후 많은 악성코드를 이용해 강화학습이 지속된다면 정확도 및 검출율은 진화할 것으로 판단된다.

참고문헌

[1] Athiwaratkun,Ben, and Jack W, Stokes, "Malware classification with LSTM and GRU language models and a character-level CNN.", 2017, Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2017/07/LstmGruCnnMalwareClassifier.pdf, 2017

[2] Seok,Seon-Hee and Kim,Ho-Won "Visualized malware classification based-on convolutional neural network". Journal of the Korea Institute of Information Security and Cryptology, vol.26, no. 1, p.197, Available: http://www.koreascience.or.kr/article/ArticleFullRecord.jsp?cn=JBBHCB_2016_v26n1_197, Feb. 2016

[3] A. Test, "Malware Statistics.", Available: https://www.av-test.org/en/s tatistics/malware//,2015. September, 2015.

[4] Anderson,Hyrrum-S, Woodbridge,Jonathan and Filar,Bobby "DeepDGA: Adversarially-tuned domain generation and detection." In Proceedings of the 2016 ACM Workshop on Artificial Intelligence and Security, p.13. Vienna, Austria, 2016, Available: https://arxiv.org/pdf/1610.01969

[5] Razvan,Pascanu, Jack W,Stokes, Hermineh Sanossian, Mady Marinescu, Anil Thomas, "Malware classification with recurrent networks." in Proceeding of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, p. 1916, Queensland, Australia. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/pascanuIcassp2015.pdf, 2015

[6] Jack W.Stokes, De.Wang, Mady Marinescu, Marc Marino,and Brian Bussone, "Attack and Defense of Dynamic Analysis-Based, Adversarial Neural Malware Classification Models." Journal of the Cryptography and Security, Available: https://arxiv.org/pdf/1712.05919, 2017.

[7] Cho, Young-Bok, Woo,Sung-Hee, Lee, Sang-Ho and Han, Chang-Su, "CUDA based Medical Image High Speed Processing Algorithm," in Proceeding of the 2017 International Conference on Future Information Communication Engineering, vol 9, no.1, p. 213, Russia, 2017, Available: http://www.dbpia.co.kr/Journal/ArticleDetail/NODE07203503

[8] Giambattista Parascandolo, Heikki Huttunen and Tuomas Virtanen, “Recurrent neural networks for polyphonic sound event detection in real life recordings”, in Proceeding of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, p. 6440, Shanghai, China, 2016, Available: https://arxiv.org/pdf/1604.00861

[9] Cho,Young-Bok, Woo,Sung-Hee and Lee,Sang-Ho,"Security Issues Using Remote Medical Treatment in Health Care Information," in Proceeding of the 2014 International Conference On Future Information & Communication Engineering, vol. 6, no. 1, p.193, 2014. Available: http://www.dbpia.co.kr/Journal/ArticleDetail/NODE07221599

[10] Cho,Young-Bok, Woo,Sung-Hee and Lee,Sang-Hoo “Genetic lesion matching algorithm using medical image”, *Journal of the Korea Institute of Information and Communication Engineering*, vol. 21, no.5.p.960, May 2017

[11] Heaton,J.B,Polson,N.G and Witte,J.H "Deep learning for finance: deep portfolios." *Journal of the Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, p.3, October 2016, Available: https://doi.org/10.1002/asmb.2209

[12] Yoon,Hye-Jin, Kim,Chang-Sik, Kwahk Kee-Young,“Research Trends Investigation Using Text Mining Techniques: Focusing on Social Network Services”, *Journal of Digital Content Society(JDCS)*, Vol. 19, No. 3, March. 2018, Available: http://www.dbpia.co.kr/Journal/ArticleDetail/NODE07408880

www.dbpia.co.kr

조영복(Young-Bok Cho)

2005년 : 충북대학교 대학원 (이학석사)
2012년 : 충북대학교 대학원 (공학박사)
2016년 : 충북대학교 대학원 (의학 박사수료)

2012년~2018년: 충북대학교 소프트웨어학과 초빙교수
2012년~현 재: ㈜소노엠 기업부설연구소 연구소장 겸직
2018년~현 재: 대전대학교 정보보안학과 조교수
※관심분야 : 의료영상처리(Medical Image Processing), 의료정보보호(Medical Information Security), 모바일보안(Mobile Security) 등

www.dbpia.co.kr