https://doi.org/10.33778/kcsa.2021.21.4.069

CNN 기반 악성코드 탐지에서 이미지 형식이 탐지성능과 자원 사용에 미치는 영향 분석

변 성 현*, 김 영 원**, 고 관 섭**, 이 수 진***

요 약

CNN 기반의 악성코드 탐지모델을 활용하기 위해 다양한 이미지 형식을 사용할 수 있다. 하지만 대부분의 기존 연구들은 최종적인 악성코드 탐지 및 분류 성능을 주로 강조하고 있으며, CNN에 입력되는 이미지의 형식이 모델의 성능과 자원 사용량에 미칠 수 있는 영향은 거의 고려하지 않는다. 이에 본 논문에서는 CNN을 기반으로 안드로이드 악성코드를 탐지하는 모델을 구축함에 있어 입력되는 이미지 형식이 탐지성능과 학습에 소요되는 자원의 사용량에 어떠한 영향을 미치는지를 분석하였다. CICAndMal2017 데이터세트를 사용하여 BMP, JPG, PNG 및 TIFF 4가지 형식의 이미지로 변환하고, 자체적으로 구축한 CNN 모델에 학습시킨 후 악성코드 탐지성능과 자원 사용량을 측정하였다. 그 결과 이미지 형식에 따른 이진분류 및 다중분류 성능과 GPU 및 RAM 사용량은 큰 차이를 보이지 않았다. 그러나 생성된 이미지의 파일 크기는 이미지 형식에 따라 최대 6배까지 차이가 났으며, 학습에 소요되는 시간에서도 유의미한 차이가 발생함을 확인하였다.

Analysis of Effects of Image Format on Detection Performance and Resource Usage in CNN-Based Malware Detection

Seong-hyeon Byeon*, Young-won Kim**, Kwan-seob Ko**, Soo-jin Lee***

ABSTRACT

Various image formats are being used when attempting to construct a malware detection model based on CNN. However, most previous studies emphasize only the detection or classification performance, and do not take into account the possible impact of image format on detection performance and resource usage. Therefore, in this paper, we analyze how the input image formats affect detection performance and resources usage when detecting android malware based on CNN. The dataset used in the experiment is the CICAndMal2017 Dataset. Subdataset extracted from the CICAndMal2017 Dataset were converted into images in four formats: BMP, JPG, PNG, and TIFF. We then trained our CNN model and measured malware detection performance and resource usage. As a result, there was no sifnificant difference between detection performance and the GPU/RAM usage, even if the image format changed. However, we found that the file size of the generated images varied by up to six times depending on the image format, and that significant differences occurred in the training time.

Key words : CNN, Image, Malware, Detection, Classification, Resource Usage

접수일(2021년 9월 28일), 게재확정일(2021년 10월 18일)

* 국방대학교 국방과학학과 석사과정(주저자)
** 국방대학교 국방과학학과 박사/석사과정(공동저자)
*** 국방대학교 국방과학학과 교수(교신저자)

1. 서 론

국방부는 19년 1월 출범한 '4차 산업혁명 스마트 국방혁신추진단'을 중심으로 4차 산업혁명 기술 기반의 「디지털 강군, 스마트 국방」 구현을 추진 중이다. 세부과제로는 '초연결 네트워크 구현을 위한 국방 모바일 및 클라우드 환경 구축'과 '지능형 스마트부대 구축' 등을 추진하고 있다.

이러한 추세로 인해 '20년에는 병사들의 스마트폰 부대 반입을 전면 허용하고 재난안전통신망을 활용하여 구축 가능한 국방 모바일업무 체계를 도입하려는 등 국방 모바일 환경을 키우기 위한 행보를 시작하고 있다.

반면, 국제적인 모바일 위협은 점차 거세지고 있다. 북한의 경우 해커 그룹인 '김수키'와 '금성121 조직' 등이 스마트폰을 해킹하여 사진, 연락처, 대화기록 등을 수집하고 있으며, 실제로 태영호 前 영국 주재 북한공사의 핸드폰에 저장된 연락처, 문자메시지, 사진 등이 북한 해커의 서버에서 발견되기도 하였다. 또한, 18년에는 전·현직 군관계자의 개인정보 6,000여건이 중국 등 해외로 유출되기도 하였던 만큼 국방 모바일 환경에 대한 보안시스템 구축이 시급한 실정이다.

한편, 기존의 보안인력에 의한 시그니처 기반 악성코드 탐지 방식으로는 많은 인력과 시간, 비용이 소모되고 악성코드에 인공지능을 적용하며 지능화·고도화되는 사이버 공격에 효과적으로 대응할 수 없다. 이에, 최근에는 기존 악성코드 탐지기법의 한계를 극복할 대안으로 머신러닝 및 딥러닝 기반의 악성코드 탐지기법이 활발하게 연구되고 있다. 또한, 사이버 공간 우위 확보를 위한 인공지능 기반의 사이버 방어 체계 고도화도 추진 중이다. 그러나 머신러닝 및 딥러닝을 기반으로 악성코드를 정확하게 탐지해내기 위해서는 대상이 될 악성코드의 특성을 모델에 정확하게 학습시키는 과정이 선행되어야 한다.

악성코드가 가진 특성을 정확하게 추출하는 과정은 특성공학(feature engineering)이라고 하며, 생성되는 탐지모델의 성능을 좌우하는 핵심 요소이다. 따라서 대부분 악성코드나 네트워크 및 시스템에 대한 전문지식을 가진 사람에 의해 수행되고 있지만, 쉽지 않은 과정이며 많은 시간이 소요된다. 특성공학 과정에서 잘못된 특성이 추출되었을 경우에는 모델 학습의 복잡도가 증가하거나 탐지성능이 현저하게 저하될 수도 있다. 이러한 이유로 최근 들어서는 특성공학을 자동으로 수행하는 CNN(Convolutional Neural Network) 알고리즘이 많은 관심을 받고 있다. CNN은 샘플이 가진 모든 특성을 이미지화하여 신경망의 입력으로 받고 학습 및 분류를 수행한다.

현재까지 제안된 CNN 기반의 악성코드 탐지기법들은 입력으로 사용할 이미지를 다양한 형태로 생성하였다. 주로 사용된 이미지 형식은 PNG, JPG, BMP 및 TIFF 등이며, 8비트 혹은 16비트 심도의 그레이스케일 이미지가 가장 많이 사용되었다. 그러나 대부분 연구는 단일 형태의 이미지를 생성 후 악성코드 탐지 성능을 측정하였을뿐, 생성되는 이미지의 형식이 탐지 성능이나 학습모델 생성에 필요한 컴퓨팅 자원 등에 어떠한 영향을 미치는지는 전혀 고려하지 않고 있다.

이러한 인식하에 본 논문에서는 인공지능 기술을 국방 모바일 보안분야에 적용함에 있어 성능과 자원 투입을 판단할 수 있는 근거를 제시하고자 한다. 특히, CNN 알고리즘에서 입력으로 사용되는 이미지 형식이 악성코드 탐지 성능(소요시간)과 자원사용(CPU, GPU, 메모리)에 미치는 영향을 분석한다. 데이터세트 CICAndMal2017[1]을 사용하며, 이미지 형식은 BMP, JPG, PNG 및 TIFF 4가지로 생성하여 학습 및 테스트에 적용하고 탐지 성능과 자원 사용량을 측정한다.

논문의 구성은 다음과 같다. 2장에서는 CNN을 기반으로 CICAndMal2017 데이터세트를 이용하여 안드로이드 악성코드 탐지를 시도했던 기존 연구를 정리하고, 3장에서는 제안하는 실험절차에 대해 설명한다. 4장에서는 실험결과를 분석하고, 마지막으로 5장에서 연구결과를 요약한 후 결론을 맺는다.

2. 관련 연구

A. H. Lashkari 등[2]은 실제 안드로이드 스마트폰을 이용하여 모바일 트래픽과 악성코드 트래픽을 수집하여 CICAndMal2017 데이터세트를 구축하였다. 그리고 특성추출(feature selection)을

활용하여 악성코드 분류를 시도하였다. 그 결과 악성코드와 정상 트래픽을 분류하는 이진분류에서는 정밀도(precision) 85.80%, 재현율(recall) 88.30%를 달성하였으나, 카테고리를 분류하는 다중분류는 50% 미만의 성능이 나타났다.

S. Fallah 등[3]은 악성코드가 악의적 행동 수행을 위해 C&C(Command & Control) 서버와의 통신으로 발생되는 네트워크 트래픽을 이용하면 ISP(Internet Service Provider)나 관리자가 전반적인 네트워크 성능을 관리할 수 있음에 착안하였다. 데이터세트에서 특성집합(feature set) 구성방안을 제안하고, KNN, RF, DT, SVM(Support Vector Machine) 등의 알고리즘을 이용하여 성능을 검증한 결과 이진분류에서 90%의 F1-Score를 달성하였다.

Kang and Lee[4]는 모바일 네트워크 트래픽 정보를 16bit의 PNG 이미지로 변환하여 CNN 기반의 안드로이드 악성코드 탐지 방안을 제시하였다. 이진분류에서 모든 성능평가 지표가 99.9% 이상으로 나타났고, 특성공학을 생략하더라도 CNN을 이용하여 안드로이드 악성코드를 효율적으로 탐지할 수 있음을 검증하였다. 다중분류에서도 전반적으로 높은 성능을 달성하였으나, 'adware' 및 'smsmalware'에서 F1-score가 각각 91.98%, 92.85%로 상대적으로 낮게 나타났다.

이외에도 ElMouatez Billah Karbab 등[5]이 DEX로부터 추출한 API Call을 이용해 CNN 기반 안드로이드 악성코드 탐지 기법을 제안하였으며, Y. Kim[6]은 IoT 기기 등의 경량화된 플랫폼에서 실행되도록 텍스트 기반의 신경망을 활용하여 악성코드를 분류하였다. X. Su 등[7]은 정상적인 애플리케이션에서 빈번히 사용하는 API 50개를 기반으로 해당 API 사용 여부를 추출하여 Random Forest를 사용하여 악성코드를 탐지하였으며, Z. Ma 등[8]은 제어 흐름 그래프를 구성하고, 실제 사용되는 API를 목록화하여 애플리케이션 분석을 통해 목록 내의 API call, 사용 빈도를 수집하여 DNN, LSTM을 이용하여 악성코드를 탐지하였다. 또한 Amin M 등[9]은 Opcode에서 사용되는 명령어 집합을 선정하고, Dex를 분석하여 선정된 명령어에 속하는 명령들을 모아 BiLSTM을 이용하였으며, H. Han 등[10]은 안드로이드 API에 존재하는 것 중 액세스 빈도가 매우 낮은 API를 제거하였다.

3. CNN 적용을 위한 이미지 생성

3.1 이미지 형식별 특징

이미지 형식별 탐지성능과 자원 사용량 비교를 위해 실험에 사용한 이미지 형식은 BMP, JPG, PNG 및 TIFF 4가지이며, 모든 형식에서 8비트 그레이스케일 이미지로 생성한다.

4가지 이미지 형식의 특징은 다음과 같다. 우선 BMP 이미지는 가장 단순한 구조를 가지지만 화질이 매우 우수하다. 화질이 우수하다는 것은 특성 정보를 이미지로 변환했을 때 정보의 왜곡 및 손실이 그만큼 적다는 것을 의미한다. JPG는 웹과 멀티미디어 환경에서 널리 사용되는 이미지 형식으로, 이미지 화질과 파일 크기를 손실압축을 통해 조정할 수 있다. 그러나 그 과정에서 일부 특성 정보의 왜곡이나 손실이 발생할 수 있다. PNG는 비손실 압축이 가능하며, TIFF는 여러 색상 모드를 유지한 상태로 저장이 가능하면서도 다양한 압축 방법을 제공한다.

3.2 데이터세트 및 전처리

실험에 사용된 데이터세트는 CICAndMal2017 데이터세트이다. 이는 구글마켓(Google Market)을 통해 수집한 악성 애플리케이션 429개와 정상 애플리케이션 5,065개를 실제 안드로이드 스마트폰에 설치한 후 다양한 동작을 수행하면서 발생되는 트래픽을 수집한 것이다. 악성코드는 4개의 카테고리에 총 42개의 패밀리로 분류하였으며, 세부 현황은 <표 1>과 같다.

<표 1> CICAndMal2017 Category & Families

Category	Family(42)
Adware (401,559)	Dowgin, Ewind, Feiwo 등 10종
Ransomware (348,943)	Charger, Jisut, Koler 등 10종
Scareware (360,970)	AVforAndroid, AVpass 등 11종
SMSmalware (222,411)	Beanbot, Biige, Zsone 등 11종

* Benign Data Set : 1,048,575개

데이터 전처리 단계에서는 CICAndMal2017 데이터세트에 포함된 특성 85개 중 Flow ID, Source IP /Port, Destination IP/Port, Protocol, Timestamp 등을 제외하고 77개의 특성 정보만을 이용하였다. 77개의 특성을 7×11의 행렬로 변환하여 0~1 사이의 값으로 정규화(normalization)한 후 8비트 그레이스케일 이미지를 생성하였다.

데이터세트에서 임의 선정된 하나의 트래픽 정보를 이용하여 각각의 형식으로 생성한 이미지 샘플은 <표 2>에서 보는 바와 같다. 육안으로는 4개의 이미지가 거의 유사하게 보이지만, 이미지의 각 픽셀이 저장한 속성값은 다르기 때문에 이미지 크기도 다름을 확인할 수 있다.

<표 2> 이미지 별 데이터 크기

	BMP	JPG	PNG	TIFF
Data Size	5,174B	860B	1,964B	4,218B
Image				

4. 실험 및 평가

BMP, JPG, PNG 및 TIFF 4가지 형식으로 8비트 그레이스케일 이미지를 생성한 후 이진분류 및 다중분류 실험을 진행하였다. 실험 결과 이진분류는 성능평가 지표가 모두 99.9% 이상을 달성하였으며, 본 장에서는 카테고리를 분류하는 다중분류 결과만을 중점적으로 기술한다.

4.1 실험 환경

본 논문의 모든 실험은 Window 10 Home 64비트 운영체제, Intel Core i5-8265U CPU, 8GB RAM 사양의 Laptop에서 Python, Keras 및 Google Colab(PRO)를 이용하여 진행하였다.

4.2 서브 데이터세트

CICAndMal2017 데이터세트를 대상으로 전처리를 수행하는 과정에서 일부 오류가 있음을 확인하였다. ‘Scareware’ 카테고리에 ‘Benign’이 2,235개 포함되어 있고, ‘Smsmalware’ 카테고리에도 ‘Benign’과 ‘Scareware’가 4,137개 오분류되어 있어 이러한 오분류 데이터는 삭제하고 서브 데이터세트 구성을 수행하였다.

전체 데이터세트에 대해 Microsoft Azure Machine Learning Studio를 이용하여 무작위로 샘플을 추출하였다. 그 결과 구성된 서브 데이터세트는 Adware 12,725개, Ransomware 15,703개, Scareware 11,819개, SMSMalware 11,970개를 포함하고 있으며, Train, Validation 및 Test를 위한 서브 데이터세트는 8:1:1 비율로 구성하여 실험에 적용하였다.

4.3 신경망 구조

실험에 사용된 모델의 신경망은 3개의 Convolution layer - Flatten layer - 2개의 Dense layer로 구성하였다. Convolution layer에서는 활성화함수로 ‘Relu’를 사용하고, 필터 수는 64 → 128 → 256개로 증가시켜 구성하였다. 각 Convolution layer층 사이에 과적합 방지 및 특성 추출을 위해 Maxpooling과 Dropout을 추가하고, 최적화 알고리즘은 ‘Adam’을 사용하였다.

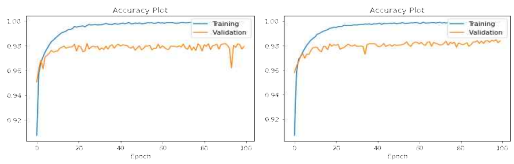
4.4 Malware 다중분류

생성된 이미지 형식에 맞춰 4가지로 구분하여 다중분류 실험을 진행하였으며, 각 실험의 세부 내용은 <표 3>에서 보는 바와 같다. 성능평가를 위해 사용된 지표는 Precision, Recall 및 F1-score 3가지이다.

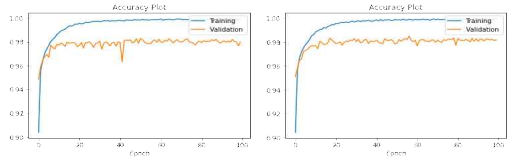
<표 3> 실험의 세부 내용

구 분	내 용
실험 1	BMP 이미지 기반의 다중분류
실험 2	JPG 이미지 기반의 다중분류
실험 3	PNG 이미지 기반의 다중분류
실험 4	TIFF 이미지 기반의 다중분류

모델 학습 시 100 epoch가 진행된 이후 과적합이 관찰되어 모든 실험의 epoch는 100으로 설정하였다. 각 실험별 Train 및 Validation 정확도는 (그림 1)에 나타난 바와 같다.



a) 실험 1(BMP) b) 실험 2(JPG)



c) 실험 3(PNG) d) 실험 4(TIFF)

(그림 1) 각 실험별 Train 및 Validation 정확도

학습 모델의 분류 정확도는 4개 이미지 형식에서 유의미한 차이를 보이지 않았으며, 테스트 결과 역시 <표 4>에서 보는 바와 같이 대부분 지표가 98% 이상으로 유사하게 나타났다. 다만 JPG 및 PNG 이미지를 이용했던 실험 2와 실험 3의 경우 Precision 값이 97.75%로 미세하게 낮게 나타났다. 각 실험별 오차행렬은 (그림 2)에서 확인할 수 있다.

<표 4> 실험별 다중분류 결과

	Category	Precision	Recall	F1-score
실험 1 (BMP)	adware	98%	98%	98%
	ransomware	99%	100%	100%
	scareware	97%	97%	97%
	smsmalware	98%	98%	98%
	Average	98%	98.25%	98.25%
실험 2 (JPG)	adware	98%	98%	98%
	ransomware	99%	100%	100%
	scareware	98%	96%	97%
	smsmalware	96%	98%	97%
	Average	97.75%	98.14	98%
실험 3 (PNG)	adware	97%	98%	98%
	ransomware	99%	100%	100%
	scareware	97%	97%	97%
	smsmalware	98%	97%	97%
	Average	97.75%	98%	98%
실험 4 (TIFF)	adware	98%	97%	98%
	ransomware	99%	100%	100%
	scareware	97%	97%	97%
	smsmalware	98%	98%	98%
	Average	98%	98%	98.25%

실험 1 (BMP)

True label	adware	ransom	scare	sms
adware	1242	2	15	13
ransom	0	1568	0	2
scare	24	3	1142	12
sms	6	6	15	1170

Predicted label

실험 2 (JPG)

True label	adware	ransom	scare	sms
adware	1244	2	11	15
ransom	0	1569	1	0
scare	17	4	1132	28
sms	4	5	13	1175

Predicted label

실험 3 (PNG)

True label	adware	ransom	scare	sms
adware	1243	2	16	11
ransom	1	1568	0	1
scare	20	4	1145	12
sms	12	6	23	1156

Predicted label

실험 4 (TIFF)

True label	adware	ransom	scare	sms
adware	1237	3	21	11
ransom	1	1568	1	0
scare	14	3	1149	15
sms	7	4	18	1168

Predicted label

(그림 2) 실험별 다중분류 오차행렬

각 실험에서 100 epoch 학습에 소요된 시간과 실험 진행 간 사용된 GPU 및 RAM 사용량을 정리한 결과는 <표 5> 및 <표 6>에서 보는 바와 같다.

<표 5> 실험 소요 시간(s)

Image	실험 1 (BMP)	실험 2 (JPG)	실험 3 (PNG)	실험 4 (TIFF)
100 epoch	958	925	953	918

<표 6> GPU 및 RAM 사용량(GB)

Image	실험 1 (BMP)	실험 2 (JPG)	실험 3 (PNG)	실험 4 (TIFF)
GPU	1.528	1.528	1.528	1.528
RAM	4.083	4.091	4.087	4.085

4.6 실험 결과 분석

본 논문의 실험을 위해 생성한 이미지는 7×11 픽셀 크기의 8비트 그레이스케일 이미지이다. 그러나 이미지 형식별 파일 크기는 <표 2>에서 확인할 수 있는 바와 같이 큰 차이를 보인다. JPG 이미지가 가장 작은 파일 크기를 가지며, BMP 이미지는 JPG 이미지 대비 평균 6배 이상 크고, TIFF 이미지는 5배 이상 크다.

개별 이미지의 파일 크기로 비교했을 때에는 불과 1~4KB 정도의 차이가 발생하는 것으로 보일 수 있지만, 모델의 학습에 사용할 데이터세트의 크기가 수백만 개 이상일 경우에는 학습 데이터 저장을 위해 요구되는 저장공간의 차이가 상당히 커진다. 실험에 사용된 악성코드 서브 데이터세트 52,217개를 JPG 이미지로 변환했을 경우 45MB 정도의 저장공간이 필요하였으나, BMP 이미지로 변환했을 경우에는 270MB 이상의 저장공간이 필요하였다. 실험에 사용된 서브 데이터세트의 크기가 전체 데이터의 4% 미만인 점을 고려하면, 전체 데이터세트를 서로 다른 이미지 형식으로 변환했을 경우 저장공간 요구량의 차이는 더욱 커질 것이다. 단순 비교를 위해 악성코드 데이터세트 전체를 BMP와 JPG 이미지로 변환해 본 결과, BMP 이미지로 변환했을 때 5GB 이상의 추가 저장공간이 필요하였다.

이미지 형식에 따른 악성코드 이진분류 성능은 99.9% 이상으로 나타났으며, 다중분류 성능도 유의미한 차이를 보이지는 않았다. 그러나 학습 소요 시간 측면에서는 유의미한 차이가 발생하였다. BMP 이미지가 가장 짧은 학습시간을 기록한 TIFF 이미지의 학습시간 대비 1 epoch 당 평균 0.4초 정도 더 소요되었으며, JPG 이미지와 대비해서는 평균 0.33초 정도 더 소요되었다. 보유한 PC 및 Google Colab에서 지원하는 가상 플랫폼의 성능 한계로 인해 전체 데이터세트를 이용한 학습시간을 측정하기는 불가능하였으나, 기 실시한 실험 결과를 바탕으로 유추했을 때 데이터세트의 크기가 커지면 학습모델 생성에 소요될 시간 역시 큰 차이를 보일 것으로 판단된다.

자원 사용량 측면에서는 모든 이미지 형식에서 GPU 사용량과 RAM 사용량이 유의미한 차이를 보이지 않았다. 이러한 결과는 앞서 언급했던 바와 같이 실험에 사용된 이미지의 크기가 7×11로 작아 학습이 진행되면서 처리해야 할 하이퍼파라미터의 수가 크게 증가하지 않았기 때문으로 판단된다.

5. 결 론

본 논문에서는 CNN을 기반으로 안드로이드 악성코드 탐지를 시도함에 있어 CNN 모델의 입력으로 사용될 이미지의 형식이 탐지 성능과 자원 사용량에 어떤 영향을 미칠 수 있는지를 분석하였다.

안드로이드 악성코드 탐지를 위해 널리 사용되고 있는 CICAndMal2017 데이터세트를 이용하여 BMP, JPG, PNG 및 TIFF 4가지 형식의 이미지로 변환하고 간단한 신경망을 구성하여 학습을 실시한 후 영향을 분석하였다. 그 결과 이미지 형식에 따른 이진분류 및 다중분류 성능은 유의미한 차이를 보이지 않았다. 그러나 생성된 개별 이미지의 파일 크기를 비교한 결과 BMP 이미지가 JPG 이미지에 비해 6배 이상 크다는 점을 확인하였다. 학습에 소요된 시간을 측정한 결과 TIFF 이미지가 가장 빠른 시간에 학습을 종료하기는 하였지만, 이미지 파일 크기에 있어서는 JPG 이미지 대비 5배 이상 크다. 이러한 결과를 바탕으로 판단해 보면 CNN을 기반으로 악성코드 탐지를 시도할 경우 탐지성능과 자원 사용량을 종합적으로 고려했을 때 JPG 이미지를 사용하는 것이 가장 바람직할 것이다.

본 연구를 통해 도출된 결과는 군이 모바일 보안에 딥러닝, 특히 CNN 알고리즘을 적용함에 있어 요망하는 탐지성능이나 자원 할당 등을 고려할 때 유용한 참고자료가 될 것으로 판단된다.

참 고 문 헌

[1] Canadian Institute for Cybersecurity, https://www.unb.ca/cic/datasets/andmal2017

[2] A. H. Lashkari, A. F. A. Kadir, L. Taheri and A. A. Ghorbani. "Toward Developing a Systematic Approach to Generate Benchmark Android malware Datasets and Classification." 2018 International Carnahan Conference on Security

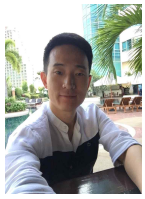
Technology (ICCST), pp. 1-7, 2018.
[3] S. Fallah and A. J. Bidgoly. "Benchmarking machine learning algorithms for android malware detection." Jordanian Journal of Computers and Information Technology (JJCIT), pp. 216-230. 2019.
[4] Jinwon Kang, Soojin Lee, "Android Malware Detection Through the Conversion of Network Traffic to Images." The Korean Institute of Information Scientists and Engineers, vol. 47, pp. 761-768, 2020.
[5] E. B. Karbab, M. Debbabi, A. Derhab, and D. Mouheb, "MalDozer: Automatic Framework for Android Malware Detection Using Deep Learning." Digital Investigation, vol. 24, pp.S48-S59, Mar.2018.
[6] Y. Kim, "Convolutional Neural Networks for Sentence Classification." Proc. of the Empirical Methods in Natural Language Processing, pp. 1746-1751, 2014.
[7] X. Su, D. Zhang, W. Li and K. Zhao, "A Deep Learning Approach to Android Malware Feature Learning and Detection." 2016 IEEE Trustcom/BigDataSE/ISPA, pp. 244-251, Tianjin, 2016.
[8] Z. Ma, H. Ge, Y. Liu, M. Zhao and J. Ma, "A Combination Method for Android Malware Detection Based on Control Flow Graphs and Machine Learning Algorithms." in IEEE Access, 7, pp.21235-21245, 2019.
[9] Amin M., Tanveer T.A., Tehseen M., Khan M. and Khan F.A., "Anwar, S. Static Malware Detection and Attribution in Android Byte-code through an End-to-end Deep System." Future Gen. Comput. Syst. 102, pp. 112 - 126. 2020.
[10] H. Han, S. Lim, K. Suh, S. Park, S. Cho and M. Park, "Enhanced Android Malware Detection: An SVM-Based Machine Learning Approach." 2020 IEEE International Conference on Big Data and Smart Computing(BigComp), pp. 75-81, Busan, Korea (South), 2020.

〔저자소개〕

변 성 현 (Seong-hyeon Byeon)

2010년 3월 해군사관학교
해양학과 학사
2020년 1월 ~ 현재 국방대학교
국방과학학과 석사과정

email : haesa64@gmail.com



김 영 원 (Young-won Kim)

2009년 3월 해군사관학교
전산학과 학사
2021년 1월 국방대학교
국방과학학과 석사
2020년 1월 ~ 현재 국방대학교
국방과학학과 박사과정

email : headsun21@gmail.com

고 관 섭 (Kwan-seob Ko)

2009년 3월 조선대학교
군사학과 학사
2020년 1월 ~ 현재 국방대학교
국방과학학과 석사과정

email : gosan8850@gmail.com

이 수 진 (Soo-jin Lee)

1992년 3월 육군사관학교
전산학과 학사
1996년 2월 연세대학교
컴퓨터과학과 석사
2006년 2월 한국과학기술원
전산학과 박사
2006년 ~ 현재
국방대학교 국방과학학과 교수

email : cyberkma@gmail.com