Journal of the Korea Academia-Industrial cooperation Society
Vol. 21, No. 9 pp. 634-642, 2020

https://doi.org/10.5762/KAIS.2020.21.9.634
ISSN 1975-4701 / eISSN 2288-4688

영상처리기법을 이용한 CNN 기반 리눅스 악성코드 분류 연구

김세진[1], 김도연[1], 이후기[2], 이태진[1*]
[1]호서대학교 정보보호학과, [2]건양대학교 사이버보안공학과

A Study on Classification of CNN-based Linux Malware using Image Processing Techniques

Se-Jin Kim[1], Do-Yeon Kim[1], Hoo-Ki Lee[2], Tae-Jin Lee[1*]
[1]Division of Information Security, Hoseo University,
[2]Department of Cyber Security Engineering, Konyang University

요 약 사물인터넷(IoT) 기기의 확산으로 인해 다양한 아키텍처가 존재하는 Linux 운영체제의 활용이 증가하였다. 이에 따라 Linux 기반의 IoT 기기에 대한 보안 위협이 증가하고 있으며 기존 악성코드를 기반으로 한 변종 악성코드도 꾸준히 등장하고 있다. 본 논문에서는 시각화한 ELF(Executable and Linkable Format) 파일의 바이너리 데이터를 영상처리 기법 중 LBP(Local Binary Pattern)와 Median Filter를 적용하여 CNN(Convolutional Neural Network)모델로 악성코드를 분류하는 시스템을 제안한다. 실험 결과 원본 이미지의 경우 98.77%의 점수로 가장 높은 정확도와 F1-score를 보였으며 재현율도 98.55%의 가장 높은 점수를 보였다. Median Filter의 경우 99.19%로 가장 높은 정밀도와 0.008%의 가장 낮은 위양성률을 확인하였으며 LBP의 경우 전반적으로 원본과 Median Filter보다 낮은 결과를 보였음을 확인하였다. 원본과 영상처리기법별 분류 결과를 다수결로 분류했을 경우 원본과 Median Filter의 결과보다 정확도, 정밀도, F1-score, 위양성률이 전반적으로 좋아졌음을 확인하였다. 향후 악성코드 패밀리 분류에 활용하거나 다른 영상처리기법을 추가하여 다수결 분류의 정확도를 높이는 연구를 진행할 예정이다.

Abstract With the proliferation of Internet of Things (IoT) devices, using the Linux operating system in various architectures has increased. Also, security threats against Linux-based IoT devices are increasing, and malware variants based on existing malware are constantly appearing. In this paper, we propose a system where the binary data of a visualized Executable and Linkable Format (ELF) file is applied to Local Binary Pattern (LBP) image processing techniques and a median filter to classify malware in a Convolutional Neural Network (CNN). As a result, the original image showed the highest accuracy and F1-score at 98.77%, and reproducibility also showed the highest score at 98.55%. For the median filter, the highest precision was 99.19%, and the lowest false positive rate was 0.008%. Using the LBP technique confirmed that the overall result was lower than putting the original ELF file through the median filter. When the results of putting the original file through image processing techniques were classified by majority, it was confirmed that the accuracy, precision, F1-score, and false positive rate were better than putting the original file through the median filter. In the future, the proposed system will be used to classify malware families or add other image processing techniques to improve the accuracy of majority vote classification. Or maybe we mean "the use of Linux O/S distributions for various architectures has increased" instead? If not, please rephrase as intended.

Keywords : Linux Malware, Machine Learning, CNN, LBP, Median Filter, Majority Voting Classifiers

본 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2018-0-00276, 딥러닝 기반 악성코드 패턴룰셋 생성 자동화 원천기술 개발).
*Corresponding Author : Tae-Jin Lee(Hoseo Univ.)
email: kinjecs0@gmail.com
Received April 20, 2020 Revised August 20, 2020
Accepted September 4, 2020 Published September 30, 2020

1. 서론

최근 사물인터넷(IoT) 시장이 빠르게 성장하면서 다양한 아키텍처가 존재하는 Linux 운영체제의 사용이 증가하였다. 이에 리눅스 기반 IoT 기기에 대한 보안 위협 또한 증가하고 있다[1]. 대규모 DDoS 공격을 일으켰던 미라이 악성코드의 소스 코드가 공개된 2016년 이후, IoT 기기를 대상으로 한 변종 미라이 악성코드가 꾸준히 등장하고 있으며 이는 또다시 대규모 DDoS 공격을 초래할 수 있다[2]. 이를 방지하기 위해 신·변종 악성코드가 유입되었을 경우 빠르고 정확하게 분류하기 위한 연구가 필요하다. 다양한 딥러닝 알고리즘을 이용하여 악성코드를 분류하는 연구 중에서도 이미지화한 악성코드를 CNN을 통해 분류하는 연구가 다수 진행되고 있다.

본 논문에서 제안하는 모델에서 또한 Convolutional Neural Network(CNN)을 이용하여 악성코드 분류기 학습모델에 적용했다. CNN은 딥러닝 알고리즘 중 하나이며 이미지를 통해 패턴을 찾아내고 이를 분류하는 데 유용하다. 얼굴 인식 애플리케이션 등에서 많이 사용되고 있으며, 이미지 특징을 직접 학습하기 때문에 특징을 수동으로 추출할 필요가 없다는 장점이 있다. 여러 층의 Convolutional 계층과 Neural Network로 이루어진 구조로 되어 있으며 Convolutional layer를 통해 이미지의 특징을 추출하고 Neural Network를 통하여 추출한 특징을 기반으로 이미지를 분류한다. 악성코드를 이미지로 시각화하였을 때 악성코드마다 고유한 특징이 나타나는 특징을 이용하여 파일을 실행시키지 않아도 분석이 가능한 ELF 파일의 바이너리 데이터를 gray-scale 이미지로 시각화한다. 시각화한 이미지를 영상처리 기법 중 LBP(Local Binary Pattern) 기법과 Median Filter 기법을 통해 전처리하여 이미지가 가지고 있는 특징을 부각시켜준다. 그 후 원본과 전처리한 이미지의 특징을 추출하여 분류한 결과를 비교 분석한다.

본 논문의 구성은 다음과 같다. 2장에서는 악성코드 분석 관련 연구를 제시하고 3장에서는 바이너리 데이터 시각화 및 CNN 기반 Linux 악성코드 분석 시스템을 제안한다. 4장에서는 이미지 변환 기법별 시험 결과와 더불어 다수결 분류 모델 성능 평가 결과를 제시하며 5장에서는 본 논문이 제시하는 CNN 기반의 Linux 악성코드 분석 연구의 주요 결론을 제시한다.

2. 관련 연구

리눅스 기반 IoT 기기 악성코드 중 Mirai 악성코드는 Linux 기반의 시스템을 원격으로 제어하는 악성코드로 IoT 기기를 감염시켜 DDoS 공격 시에 활용된다. Mirai 악성코드를 이용한 공격은 현재 진행형이며 규모가 점점 확장되고 있다[1-2].

Windows 악성코드 분석 연구에 비해 Linux 악성코드 분석에 대한 연구는 많이 찾아볼 수 없다. Windows PE(Portable Executable) 파일을 이용하여 악성코드를 분석하는 것처럼 Linux에서도 ELF(Executable and Linkable Format) 파일의 구조적 특징을 이용하여 분석이 가능하다. ELF 파일 또한 PE 파일과 같이 OS가 바이너리 실행 파일을 로드하는 데 사용하는 정보를 담고 있으며 ELF Header, Program Header table, Section Header table로 구성된다. ELF Header는 ELF 파일의 맨 앞에 위치하며 ELF 파일 포맷임을 표시하는 magic number, Architecture 등 파일 내용의 기본적인 정보를 포함한다. Program Header table은 실행 파일의 메모리 구조 내용을 표시하며 존재하는 Section의 정보를 포함한다. Fig. 1은 ELF의 전체적인 구조를 나타내며 Table 1은 ELF Header 영역에 포함되는 구조체를 나타낸 것이다.

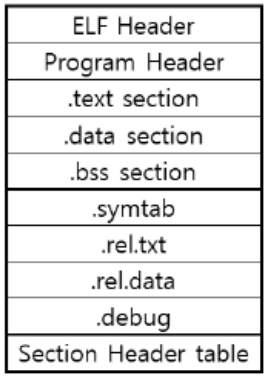


Fig. 1. ELF Structure

H. Kim et al.의 연구에서는 ELF 파일의 변종 악성코드가 원본 악성코드와 비슷한 바이너리 데이터를 갖는다는 특징을 이용하는 Dhash 기반 악성코드 탐지 기법 및 Dhash 알고리즘이 분석 시간이 느리다는 한계점을 개선하기 위해 10-gram 알고리즘을 제시한다[3].

Table 1. ELF Header Structure

Data	Explanation
e_ident	Magic number and other info
e_type	Object file type
e_machine	Architecture
e_version	Object file version
e_entry	Entry point virtual address
e_phoff	Program header table file offset
e_shoff	Section header table file offset
e_flags	Processor-specific flags
e_ehsize	ELF header size in bytes
e_phentsize	Program header table entry size
e_phnum	Program header entry count
e_shentsize	Section header table entry size
e_shnum	Section header table entry count
shstrndx	Section header string table index

J. Hwang et al.의 연구에서 또한 이런 ELF의 구조적 특징을 이용해 feature를 추출하고 이를 k-Nearest Neighbors(KNN), DNN 등 다양한 머신러닝 기법을 이용하여 Linux 악성코드를 분류한다. 또한 Linux 악성코드를 시각화하여 Convolutional Neural Network (CNN)을 이용하여 악성코드를 분류한 결과 95.05%의 분류 정확도를 검증하였다[4].

악성코드를 시각화하여 CNN을 이용해 분류하는 연구도 다수 진행되고 있다[5-9]. S. Seok et al.의 연구에서는 악성코드의 이미지화를 통한 패밀리를 분류하는 방법을 제안하였다[5]. 이미지 변환 알고리즘을 이용하여 이미지를 일정한 크기로 변환, CNN 알고리즘을 통해 악성코드 패밀리를 분류하였다. 분류기의 입력 이미지 형태를 맞춰주기 위해 이미지를 256x256 크기의 8-bit gray-scale로 변환한다. 변환 후 4개의 convolution layer와 2개의 full connect layer 2개로 구성된 CNN을 이용해 시각화된 악성코드를 분류하였다. 9개의 악성코드 패밀리로 구성된 Microsoft 데이터 셋 20,000개를 이용해 10-flod cross validation 기법을 적용하여 악성코드 분류 정확도를 검증한 결과 96.2%의 정확도가 도출되었다. 또한 많은 수의 악성코드 유형이 존재할 때의 분류 정확도를 확인하기 위하여 VXHeavens 데이터셋을 이용했다. 앞선 Microsoft 데이터 셋과 동일하게 10-flod cross validation 기법을 적용하여 분류 성능을 검증하였을 때 82.9%의 정확도가 도출되었다. 하루에도 셀 수 없이 많은 악성코드들이 발견되는 상황에서 S. Seok et al.의 모델은 샘플 데이터의 수가 충분히 확보되지 않는 이상 악성코드 패밀리 분류에 대한 학습이 충분하지 못해 분류 정확도가 낮아진다는 단점이 존재한다.

G. Lim et al.의 연구에서는 다양한 크기를 가진 악성코드 바이너리를 256 x 256 크기의 여러 장의 이미지로 전처리하여 Dynamic Recurrent Neural Network (RNN)와 CNN을 이용하여 분류할 수 있는 딥러닝 모델을 제안하였다[7]. 96%의 분류 정확도가 검증되었고 악성코드에 대한 복잡한 전처리가 없다는 장점이 존재하나 ConvLSTM Encoder, Residual Block 등을 사용하여 모델 학습 시 많은 연산을 필요로 하고 데이터 셋 불균형으로 인해 특정 클래스에서 분류 정확도가 떨어지는 현상을 보였다.

T. Kim et al. 연구에서는 실험을 위해 학습용 데이터와 더불어 패밀리 정보를 제공하지 않는 악성코드 데이터를 사용하였다. CNN과 Random forest 알고리즘을 통해 악성코드를 분류했을 때 각각 98.9%, 97.1%의 높은 분류 정확도를 검증하였다[10].

악성코드들에 대해 이미지화를 진행하였을 때 같은 패밀리의 악성코드들은 유사한 이미지 특성을 갖는다. W. Huang et al.의 연구에서는 악성코드가 갖는 이러한 특징을 이용하여 악성코드 이미지를 robust 한 전처리를 진행하였다[11]. Support Vector Machine(SVM) 분류기를 적용하여 분류할 수 있는 머신러닝 모델을 제안하였으며 2개의 local feature와 Histogram of Oriented Gradient(HOG), Pixel Intensity 등 6개의 global feature를 이용하여 이미지 전처리를 진행하였다. Fig. 2는 SVM을 이용하여 악성코드 분류 정확도를 그래프로 나타낸 것이다. 평균 84%의 정확도가 도출되었으며 패밀리 별 데이터의 수가 많을수록 높은 정확도를 보였다. Fig. 3는 이미지 robust hashing 과정을 나타내며 Fig. 4은 이미지화한 악성/정상 파일을 나타낸다.

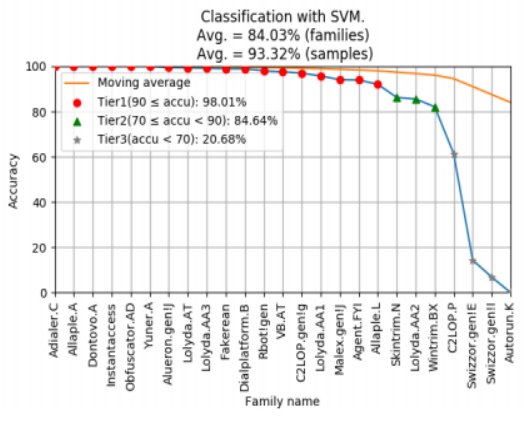


Fig. 2. Classification accuracy using SVM

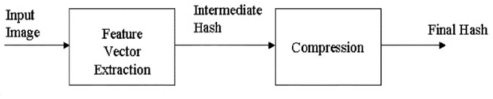
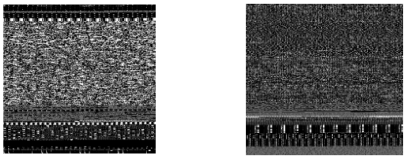


Fig. 3. Process of robust hashing



(a) (b)

Fig. 4. Example of Goodware and Malware Images
(a) Goodware (b) Malware

Luo et al. Ahonen et al. 및 Pietikainen의 연구에서 이미지 전처리 기법 중 Local Binary Pattern(LBP) 기법을 이용한 악성코드 및 이미지 분류 연구를 진행하였다. LBP의 질감 처리 특징을 이용하여 이미지 전처리를 통해 얼굴 인식 방법을 제시한다.

악성코드 이미지를 통한 악성코드 분류 정확도를 도출하였을 때 악성코드 이미지가 손상되거나 훼손될 경우 악성코드 분류가 제대로 되지 않는다는 문제점이 있다. Robust 한 악성코드 이미지 처리를 통해 악성코드의 전체적인 특징을 이용하여 분류 정확도를 높이는 기술이 필요하다. 본 연구에서는 시각화시킨 Linux 악성코드 원본을 Local Binary Pattern(LBP), Median Filter 처리를 한 후 CNN을 이용하여 악성코드 분류를 진행하고 원본 이미지와 더불어 분류 정확도를 도출하였다. 또한, 보다 나은 성능의 예측을 위해 다수결 분류 중 Hard Voting 방식을 사용하여 영상처리 기법별 분류 결과들에 대한 다수결 분류모델의 성능평가 결과를 제시한다.

3. 시스템 구성

본 논문에서는 영상처리기법을 적용한 Feature의 성능을 확인을 위해 다음과 같이 진행하였다. 시각화한 ELF 파일에 영상처리 기법 중 Median Filter 기법과 LBP 기법을 적용한다. 그 후 머신러닝 모델인 CNN 알고리즘을 이용하여 학습을 진행한 후 테스트 데이터의 악성 여부를 판단한다. 또한 영상처리 기법별 분류 결과를 집계하여 Voting Classifiers도 진행하였다. Fig. 5는 전체 시스템 흐름을 나타낸다.

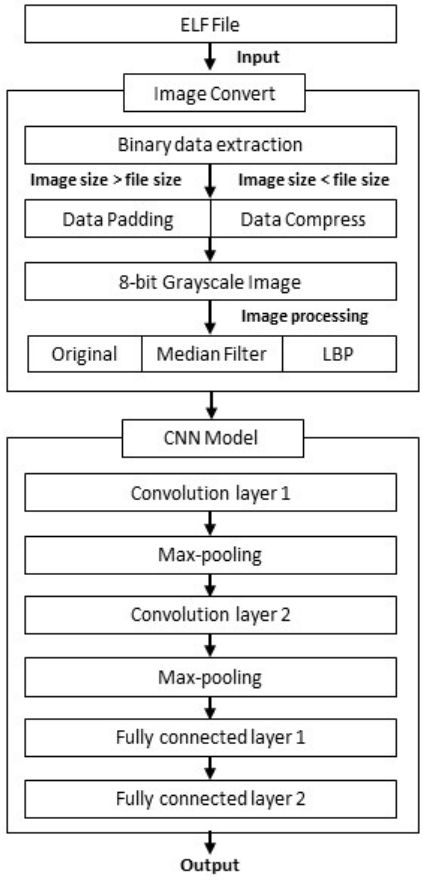


Fig. 5. Overall System Configuration

3.1 ELF 바이너리 데이터의 시각화

본 논문에서는 ELF 파일의 바이너리 데이터를 64x64

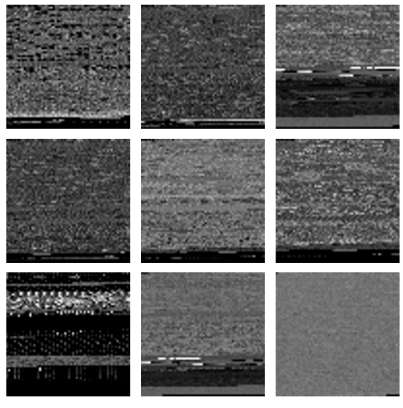


Fig. 6. Example visualization of ELF binary data

크기의 8bit gray-scale 이미지로 변환하였다. 이 과정에서 파일의 크기가 지정한 이미지 크기보다 큰 경우에는 압축, 작을 경우에는 제로 패딩을 통해 고정된 크기의 이미지를 생성한다. Fig. 6은 ELF 바이너리 데이터의 시각화 예시이다.

3.2 영상처리

악성코드 분류 시 outlier에 크게 영향을 받지 않도록 시각화한 ELF 파일에 영상처리 기법을 적용한다. 본 논문에서는 LBP 기법과 Median Filter 기법을 적용하였다. Fig. 7은 영상처리의 예시를 나타낸다.

(a) (b) (c)

Fig. 7. Image processing Example
(a) Original (b) LBP (c) Median Filter

3.2.1 LBP(Local Binary Pattern)

LBP는 이미지의 모든 픽셀에 대해 계산되는 값으로 각 픽셀의 3x3영역의 상대적인 밝기 변화를 2진수로 코딩한 인덱스 값이다. LBP의 전처리 과정은 다음과 같다. Pixel 하나를 기준으로 주위 8개의 픽셀을 가져온다. 중심 픽셀을 기준으로 주위 픽셀의 크기가 중심 픽셀보다 크거나 같을 경우 1, 작을 경우 0으로 threshold 해준다. 변환된 이진값을 반시계방향으로 8bit에 저장 후 10진수로 변환한 후 변환된 10진수 값을 중심 픽셀에 삽입한다. 모든 pixel에 대해서 위와 같은 과정을 반복한다.

이미지의 질감 표현 및 얼굴 인식 등에 활용되는 간단하면서도 효율적인 방법이면서 각 픽셀에 대한 상대적인 밝기 변화로 값을 나타내기 때문에 밝기가 변해도 Robust한 특징을 갖는다. Fig. 8은 gray-scale 된 이미지에 대해 LBP 기법을 적용하고 적용된 이미지를 histogram으로 표현한 것이다.

3.2.2 Median Filter

Pixel의 값들을 크기 순서대로 정렬한 후 중간 크기의 값을 선택하는 Median Filter는 비선형 디지털 필터 기술로 이미지나 기타 신호로부터 신호 잡음을 제거하는데 자주 이용되며, 통상적으로 이미지 프로세싱에서 윤곽선 감지 같은 높은 수준의 처리를 수행하기 전 단계인 이미지에 고성능 잡음 제거인 소금-후추 잡은 제거에 효과적이다. 이미지 프로세싱에서 자주 이용되며, 특히 스펙클 노이즈나 작은 반점들을 줄이는 데 유용하다. 그러나 pixel의 값이 변경되면서 이미지의 윤곽이 무너질 수 있으며 이로 인해 블러링 효과가 나타난다.

Median Filter의 전처리 과정은 다음과 같다. Pixel 하나를 기준으로 주위 8개의 pixel의 RGB 값을 도출한다. 도출한 pixel의 RGB 값을 Bubble Sorting을 거쳐 오름차순으로 나열한 후 나열한 값 중에 중간 크기의 값을 선택하여 기준 pixel에 삽입한다. 이러한 방법으로 주위 8개의 값과 비교하여 중간값을 찾아내어 출력할 이미지에 순차적으로 넣게 되면 Median Filter의 값을 얻을 수 있다.

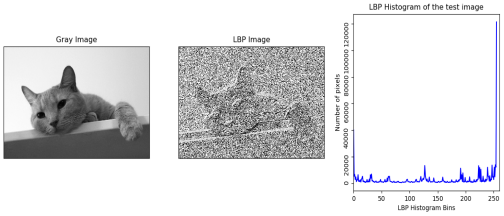


Fig. 8. LBP Conversion Example

3.3 CNN 기반 Linux 악성코드 분류기

영상처리 기법을 적용한 Feature의 성능을 확인하기 위해 머신러닝 모델인 CNN 알고리즘을 사용하였다. 본 논문에서 사용한 CNN 모델은 2개의 Convolution layer와 2개의 Fully Connected layer로 구성되어 있다. Convolution layer와 Fully Connected layer 사이에는 Max-pooling layer와 Dropout layer를 배치하여 차원을 줄이고 과적합을 방지하였다. 활성화 함수는 ReLu(Rectified Linear Unit)를 사용하였다. Table 2와 Table 3은 제안한 CNN 모델의 아키텍처와 파라미터의 설정값을 나타내며 Fig. 9는 CNN 모델의 아키텍처를 그림으로 나타낸 것이다.

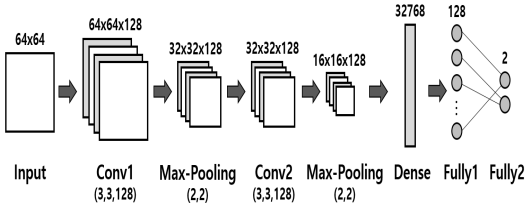


Fig. 9. Architecture of the CNN model

Table 2. CNN Architecture

Layer	Parameters	Value	Output
Input layer		64x64	64x64x1
Convolution layer1	Kernel_size	3x3	64x64x128
	Filter	128	
	Stride	1	
Max-Pooling 1	Kernel_size	2x2	32x32x128
	Stride	2	
Dropout layer			
Convolution layer2	Kernel_size	3x3	32x32x128
	Filter	128	
	Stride	1	
Max-Pooling 2	Kernel_size	2x2	16x16x128
	Stride	2	
Dropout layer			
Dense layer			32768
Fully Connected 1			128
Dropout layer			
Fully Connected 2			2

64x64 크기의 이미지를 입력으로 하는 첫 번째 Convolution layer는 128개의 3x3 kernel로 필터링하여 128개의 64x64 이미지를 생성한다. 생성된 이미지에 bias를 더하고 활성화 함수 ReLu를 적용한다. 이후 2x2 Max_pooling 필터를 진행하여 128개의 32x32의 이미지에 Dropout layer를 적용하여 과적합을 방지한다. 두 번째 Convolution layer에서는 이전단계에서 생성된 128개의 32x32 이미지를 입력으로 하여 128개의 3x3 kernel로 필터링을 진행한다. 생성된 이미지에 bias를 더해주고 Relu 함수를 적용한다. 이후 2x2의 Max_pooling을 진행하여 128개의 16x16 이미지들을 생성한다. 생성된 이미지들을 하나의 특징 데이터로 만든 32,768개의 값을 첫 번째 Fully Connected layer의 입력값으로 넣으면 128개의 값이 출력된다. 출력된 128개의 값을 두 번째 Fully Connected layer에 입력하면 2개의 값이 최종 출력으로 나온다. 본 논문에서는 batch size를 13, epoch 값을 100으로 설정하여 학습을 진행하였다.

Table 3. CNN Parameters

Parameters	value
Convolution layer	2
Max_Pooling layer	2
epochs	100
batch size	13
Leaming rate	0.001

3.4 다수결 분류

다수결 분류에는 단순하게 표를 많이 받은 값을 최종 예측값으로 정하는 Hard Voting과 가중치가 가장 높은 것을 예측값으로 정하는 Soft Voting이 있다. 본 논문에서는 Hard Voting 방식으로 영상처리 기법별 결과를 집계하여 표가 제일 많은 값을 예측값으로 정한다. Fig. 10은 Hard Voting 방식의 과정을 나타낸다.

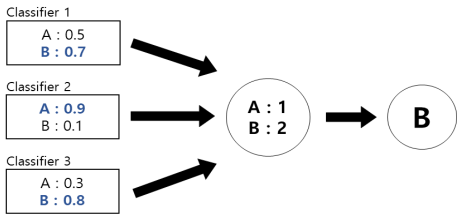


Fig. 10. Example of Hard Voting

4. 시험 결과

4.1 Dataset

본 논문에서 사용한 데이터 셋은 Virusshare에서 제공하는 Linux 악성코드 10,000개와 자체 수집한 정상파일 10,000개로 구성되어 있으며 악성/정상 각 8,000개씩 총 16,000개의 데이터를 학습 데이터로 사용하고 나머지 4,000개의 데이터를 테스트에 사용하였다.

4.2 영상처리 기법별 분류 결과

본 절에서는 영상처리기법을 이용한 Linux 악성코드 분류모델의 분석 결과를 제시한다. 본 논문에서는 시각화한 ELF 파일에 영상처리 기법인 LBP와 Median Filter를 적용하여 악성코드 분류를 진행했다. 그 결과 98.77%의 정확도로 원본이 가장 높은 정확도를 보였으며 Median Filter가 98.57%, LBP가 가장 낮은 96.47%의 정확도를 보였다. Median Filter의 경우 원본보다 낮은 정확도를 보였으나 99.19%의 높은 정밀도와 0.8%의 낮은 위양성률을 확인할 수 있었다. 분류 결과를 통해 영상처리 기법을 적용한 Feature가 악성코드 특징을 잘 나타내고 있음을 확인할 수 있었으며 악성코드 분석에 좋은 탐지율을 보임을 확인했다. Table 4는 원본, LBP, Median Filter 데이터의 분류 결과와 기법별 분류 결과를 집계하여 다수결로 분류한 결과를 나타낸다.

Table 4. Evaluation results of Hard Voting Classification and CNN-based Linux malware classification model using image processing

Performance evaluation	Original	LBP	Median Filter	Voting Classifiers
Accuracy	98.77%	96.47%	98.57%	98.87%
Precision	98.99%	96.87%	99.19%	99.39%
Recall	98.55%	96.05%	97.95%	98.35%
FPR	1%	3.1%	0.8%	0.6%
F1-score	98.77%	96.46%	98.57%	98.87%

4.3 다수결 분류 결과

4.2의 결과를 통해 Median Filter와 LBP를 적용한 Feature가 악성코드 분류에 좋은 성능을 보임을 확인하였다. 본 논문에서는 영상처리 기법별 분류 결과를 다수결로 분류했을 시, 분류 성능이 향상될 것으로 해석하여 다수결 분류를 진행하였다. 그 결과 4.2의 결과에서 정확도가 가장 높게 나왔던 원본 대비 0.1% 더 높은 98.87%의 정확도를 확인할 수 있었다. 또한, Median Filter의 정밀도와 위양성률보다 더 나은 점수를 보였으며 전체적으로 분류 성능이 향상되었음을 확인했다. 다른 영상처리 기법들을 추가하여 다수결 분류를 진행할 경우 악성코드 탐지율이 더욱 향상될 것으로 전망한다.

Fig. 11은 영상처리기법 및 다수결 분류에 대한 Confusion matrix를 나타낸다. Table 4는 영상처리기법별 분류 결과에 대한 다수결 분류 결과를 표로 나타낸 것이며 Fig. 12는 해당 결과를 ROC Curve로 나타낸 것이다.

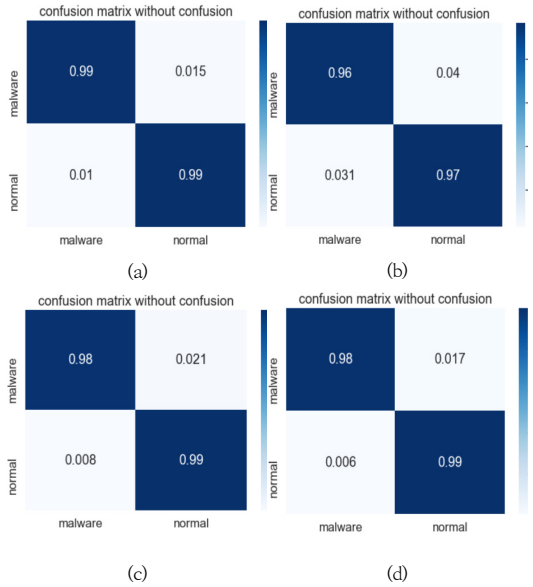


Fig. 11. Confusion matrix
(a) Original (b) LBP (c) Median Filter (d) Voting

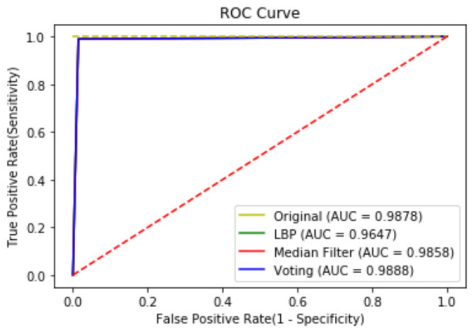


Fig. 12. ROC Curve

5. 결론

IoT 기기의 확산에 따라 Linux를 기반으로 하는 IoT 기기들에 대한 보안 위협이 증가하고 있다. 본 논문에서는 Linux 악성코드에 대응하기 위해 ELF 파일의 바이너리 데이터를 시각화한 후, Median Filter와 LBP 기법을 적용해 CNN 모델로 분류하는 시스템을 제안하였다. 제안한 모델로 악성코드를 분류한 결과 원본이 98.77%로 가장 높은 정확도를 보였으나 Median Filter를 적용하여 분류했을 경우, 재현율과 위양성률이 원본보다 개선되었음을 확인할 수 있었다. 또 한 영상처리 기법별 분류 결과에 대해 다수결 분류를 진행한 결과 98.87%의 정확도로 원본보다 높은 정확도를 보였으며 전체적으로 분류 성능이 향상되었음을 확인했다. 우리는 영상처리 기법을 적용한 Feature가 악성코드 특징을 잘 나타내고 있음을 확인했으며 다른 영상처리 기법을 추가하여 다수결 분류를 진행했을 경우 더 좋은 성능을 보일 것으로 기대한다. 향후, 제안한 시스템으로 악성코드 패밀리 분류를 진행하거나 다른 영상처리기법을 추가하여 다수결 분류의 정확도를 높이는 연구를 진행할 예정이다.

References

[1] KISA & KrCERT, “2016 Mirai Malware Trends Report”, Technical report, KISA, Republic of Korea, pp. 2-8

[2] B. N. Noh, Embedded Linux based IoT device malware analysis technology research, Technical Report, KISA, Korea, pp.1-7

[3] Hongbi, Kim, Hyunseok, Shin, Junho, Hwang, Taejin, Lee, "Malware Variants Detection based on Dhash", Journal of KIISE, Vol. 46, No. 11, pp. 1207-1214, 2019.11
DOI : https://dx.doi.org/10.5626/JOK.2019.46.11.1207

[4] Jun-ho, Hwang, Tae-jin, Lee, “Study of Static Analysis and Ensemble-Based Linux Malware Classification”, Journal of The Korea Institute of Information Security & Cryptology, VOL. 29, NO. 6, pp. 1327-1337, Dec. 2019
DOI : https://dx.doi.org/10.13089/JKIISC.2019.29.6.1327

[5] Seon-hee, Seok, Ho-won, Kim, “Visualized Malware Classification Based-on Convolutional Neural Network”, Journal of The Korea Institute of Information Security & Cryptology, VOL. 26, NO. 1, pp. 197-208, Feb. 2016
DOI : https://dx.doi.org/10.13089/JKIISC.2016.26.1.197

[6] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: visualization and automatic classification," Proc. of the 8th international symposium on visualization for cyber security, pp. 1-7, 2011
DOI : https://dx.doi.org/10.1145/2016904.2016908

[7] Geun-Youngm, Lim, Young-Bok, Cho, “Dynamic RNN-CNN malware clasifer corespond with Random Dimension Input Data”, Journal of the Korea Instiute of Information and Comunication Enginering, Vol. 23, No. 5, pp. 533~539, May 2019

[8] Ho-Sung, Woo, Geon-Ung, Cheong, Jun-Woo-Cho, Jae-Hyun, Kim, "Antivirus Software Using CNN", Proceedings of Symposium of the Korean Institute of communications and Information Sciences , pp. 385-386, 2018.11

[9] Jiawei Su, Danilo Vasconcellos Vargas, Sanjiva Prasad, Daniele Sgandurra, Yaokai Feng, Kouichi Sakurai, "Lightweight Classification of IoT Malware based on Image Recognition", 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 2018
DOI : https://dx.doi.org/10.1109/COMPSAC.2018.10315

[10] Tae-Guen, Kim, Hwan-Tae, Ji, Eul-Gyu, Im, “Malware Classification Using Machine Learning and Binary Visualization”, KIISE Transactions on Computing Practices, Vol. 24, No. 4, pp. 198-203, 2018.4
DOI : https://dx.doi.org/10.5626/KTCP.2018.24.4.198

[11] Wei-Chung, Huang, Fabio Di Troia, Mark Stamp, “Robust Hashing for Image-based Malware Classification”, Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018), Vol. 1, pp. 451-459, 2018.
DOI: https://dx.doi.org/10.5220/0006942204510459

[12] Jhu-Sin, Luo, Dan, Lo, Malware Image Classification using Machine Learning with Local Binary Pattern, Master's thesis, Kennesaw State University of Computer Science, 2018.5
DOI : https://dx.doi.org/10.1109/TPAMI.2006.244

[13] T. Ahonen, A. Hadid, M. Pietikainen, “Face Description with Local Binary Patterns: Application to Face Recognition”, IEEE Transactions on Pattern Analysis and Machine Intelligence Vol. 28 , Issue. 12 , pp. 2037-2041, Dec. 2006

[14] Matti Pietikäinen, Local Binary Patterns, Scholarpedia, 2010,
http://www.scholarpedia.org/article/Local_Binary_Patterns (accessed Apr. 3, 2020)

[15] Muhammad Furqan Rafique, Muhammad Ali, Aqsa Saeed Qureshi, Asifullah Khan, Jin Young Kim, Anwar Majid Mirza, “Malware Classification using Deep Learning based Feature Extraction and Wrapper based Feature Selection Technique”, 2019

김 세 진(Se-Jin Kim) [준회원]

- 2020년 2월 : 호서대학교 정보보호학과 (공학사)
- 2020년 3월 ~ 현재 : 호서대학교 일반대학원 정보보호학과 (석사과정)

〈관심분야〉
악성코드 분석, 정보보호, 머신러닝/딥러닝

김 도 연(Do-Yeon Kim) [준회원]

- 2018년 2월 : 건양대학교 정보보호학과 (공학사)
- 2020년 3월 ~ 현재 : 호서대학교 일반대학원 정보보호학과 (석사과정)

〈관심분야〉
정보보호, 악성코드 분석, 머신러닝

이 후 기(Hoo-Ki Lee) [정회원]

- 2018년 2월 : 숭실대학교 일반대학원 IT정책학과 (공학박사)
- 2018년 3월 ~ 현재 : 건양대학교 프라임창의융합대 사이버보안공학과 교수

〈관심분야〉
Data Science, Digital Forensic, CTA

이 태 진(Tae-Jin Lee) [정회원]

- 2008년 2월 : 연세대학교 공학대학원 컴퓨터공학과 (공학석사)
- 2013년 1월 ~ 2017년 2월 : 한국인터넷진흥원 팀장
- 2017년 2월 : 아주대학교 일반대학원 컴퓨터공학과 (공학박사)
- 2017년 3월 ~ 현재 : 호서대학교 정보보호학과 교수

〈관심분야〉
시스템 보안, 악성코드 분석, 침해사고 대응