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

ResNet-Variational AutoEncoder기반 변종 악성코드 패밀리 분류 연구☆

A Study on Classification of Variant Malware Family Based on ResNet-Variational AutoEncoder

이 영 전[1] 한 명 묵[1]*
Young-jeon Lee Myung-Mook Han

요 약

전통적으로 대부분의 악성코드는 도메인 전문가에 의해 추출된 특징 정보를 활용하여 분석되었다. 하지만 이러한 특징 기반의 분석방식은 분석가의 역량에 의존적이며 기존의 악성코드를 변형한 변종 악성코드를 탐지하는 데 한계를 가지고 있다. 본 연구에서는 도메인 전문가의 개입 없이도 변종 악성코드의 패밀리를 분류할 수 있는 ResNet-Variational AutoEncder 기반 변종 악성코드 분류 방법을 제안한다. Variational AutoEncoder 네트워크는 입력값으로 제공되는 훈련 데이터의 학습 과정에서 데이터의 특징을 잘 이해하며 정규 분포 내에서 새로운 데이터를 생성하는 특징을 가지고 있다. 본 연구에서는 Variational AutoEncoder의 학습 과정에서 잠재변수를 추출을 통해 악성코드의 중요 특징을 추출할 수 있었다. 또한 훈련 데이터의 특징을 더욱 잘 학습하고 학습의 효율성을 높이기 위해 전이 학습을 수행했다. ImageNet Dataset으로 사전학습된 ResNet-152 모델의 학습 파라미터를 Encoder Network의 학습 파라미터로 전이했다. 전이학습을 수행한 ResNet-Variational AutoEncoder의 경우 기존 Variational AutoEncoder에 비해 높은 성능을 보였으며 학습의 효율성을 제공하였다. 한편 변종 악성코드 분류를 위한 방법으로는 앙상블 모델인 Stacking Classifier가 사용되었다. ResNet-VAE 모델의 Encoder Network로 추출한 변종 악성코드 특징 데이터를 바탕으로 Stacking Classifier를 학습한 결과 98.66%의 Accuracy와 98.68의 F1-Score를 얻을 수 있었다.

☞ 주제어 : 변종 악성코드, 악성코드 분류, 변이 오토인코더, 전이학습, 앙상블 학습

ABSTRACT

Traditionally, most malicious codes have been analyzed using feature information extracted by domain experts. However, this feature-based analysis method depends on the analyst's capabilities and has limitations in detecting variant malicious codes that have modified existing malicious codes. In this study, we propose a ResNet-Variational AutoEncder-based variant malware classification method that can classify a family of variant malware without domain expert intervention. The Variational AutoEncoder network has the characteristics of creating new data within a normal distribution and understanding the characteristics of the data well in the learning process of training data provided as input values. In this study, important features of malicious code could be extracted by extracting latent variables in the learning process of Variational AutoEncoder. In addition, transfer learning was performed to better learn the characteristics of the training data and increase the efficiency of learning. The learning parameters of the ResNet-152 model pre-trained with the ImageNet Dataset were transferred to the learning parameters of the Encoder Network. The ResNet-Variational AutoEncoder that performed transfer learning showed higher performance than the existing Variational AutoEncoder and provided learning efficiency. Meanwhile, an ensemble model, Stacking Classifier, was used as a method for classifying variant malicious codes. As a result of learning the Stacking Classifier based on the characteristic data of the variant malware extracted by the Encoder Network of the ResNet-VAE model, an accuracy of 98.66% and an F1-Score of 98.68 were obtained.

☞ keyword : Variant Malware, Malware Classification, Variational AutoEncoder, Tranfer Learning, Ensemble Learning

1. 서 론

전통적으로 악성코드 분석은 악성코드 도메인 전문가에 의해 수행되었다. 이러한 특징 기반의 접근 방법은 도메인 전문가에 의존적이며 급변하는 변종 악성코드에 대응할 수 없다는 문제점을 가지고 있다[1].

[1] Department of Software, Gachon University, Seongnam-si, 13120, Korea.
* Corresponding author (mmhan@gachon.ac.kr)
[Received 26 October 2020, Reviewed 18 November 2020(R2 4 March 2021), Accepted 23 March 2021]
☆ 본 연구는 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행되었음 (NRF-2018R1D1A1B07050864)

본 연구에서는 악성코드 도메인 전문가의 분석 없이도 악성코드의 중요특징 정보를 추출하며 변종 악성코드의 패밀리를 분류할 수 있는 ResNet-Variational AutoEncoder 기반 변종 악성코드 군집 분류 연구를 제안한다.

Variational AutoEncoder 네트워크는 입력값으로 제공되는 훈련 데이터의 학습 과정에서 데이터의 특징을 잘 이해하며 정규 분포 내에서 새로운 데이터를 생성하는 특징을 가지고 있다. 본 연구에서는 Variational AutoEncoder를 활용해 새로운 데이터를 생성하며 잠재적인 특징을 학습하는 Encoder Network를 특징 추출기로 활용하여 변종 악성코드의 주요 특징을 추출할 수 있었다.

한편 본 연구에서는 Kingma, Diederik P[2]가 제안한 Variational AutoEncoder과 달리 Encoder Network에서 전이학습을 수행한다. ImageNet Datset으로 사전학습된 ResNet-152 모델의 파라미터를 Variational AutoEncoder의 Encoder Network로 전이하는 방법을 통해 기존 Variational AutoEncoder 대비 높은 성능과 효율적인 학습이 가능했다.

변종 악성코드 분류를 위한 방법으로는 앙상블 분류 모델인 Stacking Classifier가 사용된다. 사전학습된 ResNet-Variational AutoEncoder의 Encoder Network를 통해 추출한 특징 데이터에 Stacking Classifier를 학습하여 변종 악성코드 패밀리 분류를 수행했다. Stacking Classifier를 활용한 앙상블 학습 수행한 결과 98.66%의 높은 분류 정확도와 98.68의 F1-score를 얻을 수 있었다.

본 구성은 다음과 같다. 제 2장과 3장에서는 관련 연구와 제안하는 방법에 대하여 설명한다. 제 4장에서는 실험을 통한 제한 하는 방법의 검증과 평가를 수행한다. 5장에서는 결론 및 향후 연구 방향을 제안한다.

2. 관련 연구

2.1 악성코드 특징 추출 연구

전통적인 악성코드 분석 연구로는 Static Analysis, Dynamic Analysis, Hybrid Analysis 등이 있다. 각각의 방법은 도메인 전문가에 의한 특징 추출 방법, 악성코드를 직접 실행시켜 분석하는 방법 그리고 앞선 두 방법을 조합하여 더 좋은 성능을 내는 방법으로 정의할 수 있다.하지만 이러한 방법들은문제점들을 가지고 있다.

우선 Static Analysis의 경우 악성코드 전문가의 지식에 따라 특징이 추출되는 문제점이 있다. 즉 악성코드 분석자의 지식에 매우 의존적기 때문에 분석자에 따라 다른 결과가 발생한다는 문제점이 있다[1]. Dynamic Analysis의 경우 Static Analysis보다 성능이 좋은 것으로 알려져 있으나 Sand Box가상 환경을 구축해야 하는 비용적인 측면과 악성코드 행위를 탐지하기 위한 시간적, 비용적인 문제점이 존재한다. 이러한 두 분석 방법의 장점을 적절히 조합하여 Hybrid Analysis라는 방법이 제안되었지만 Hybrid Analysis역시 기존의 방법들이 가진 문제점을 해결하지 못한다는 문제점을 가지고 있다[3].

2.2 이미지 처리 기반 악성코드 분류 연구

Nataraji et al.[3, 4]연구팀은 악성코드의 시각화를 통한 binary texture기반의 악성코드 탐지 방법을 제안했다.

일반적으로 악성코드 작성자는 원본 악성코드의 작은 부분을 바꿔 새로운 변종 악성코드를 생성한다. 이러한 악성코드를 이미지로 변환할 경우 작은 변화를 탐지하면서 악성코드의 구조적인 정보는 유지할 수 있다. 또한 악성코드의 시각화를 통해 다양한 서브섹션들이 서로 다른 텍스쳐를 나타내는 특징을 발견하였다[3]. 특히 같은 군집 내에 속하는 악성코드의 텍스쳐는 시각적으로 유사한 특징을 보이며 군집이 다른 악성코드 간에는 텍스쳐가 다른 특징을 보이는 점에 영감을 받아 악성코드 이미지 유사도 기반의 특징 추출 연구를 제안했다. 이들의 연구는 향후 많은 연구자들에 의해 발전되었으며 악성코드 분석의 전통적인 Static Analysis, Dynamic Analysis에 이어 Binary Image Analysis라는 새로운 접근법으로 평가받고 있다[3, 6, 7, 9, 10].바이너리 악성코드를 이미지로 변환할 경우 악성코드의 구조적인 정보는 유지하면서도 작은 변화를 탐지할 수 있다는 장점이 있다.

2.3 딥러닝을 활용한 악성코드 분류 연구

최근 Anomaly Detection, Fraud Detection, Malware Detection 등 다양한 분야에서 딥러닝을 활용한 연구들이 활발하게 진행되고 있다. 이러한 딥러닝 기술의 발전은 기존의 악성코드 분석 방법이 가지고 있던 한계점을 극복하는 방안을 마련해주었다[5].

제로 데이 공격을 탐지하기 위한 연구로는 tDCGAN(Transferred Deep-Convolutional Generative Adversarial Network)을 활용한 연구가 있다[6]. 사전학습된 AutoEncoder 모델을 활용하여 GAN(Generative Adversarial Network)모델을 안정적으로 학습하였으며 학습된 GAN 모델의 Discriminator 파라미터를 제로 데이 공격 탐지를 위한 Detector로 전이하였다. tDCGAN을 활용한 악성코드 분류의 정확

도는 전통적인 머신러닝 기반의 방법보다 뛰어난 95.74%를 달성했다. 하지만 본 연구에서 사용한 딥러닝 생성 모델의 경우 랜덤 분포에서 데이터를 생성하는 것이 아닌 특정 분포에서 데이터를 생성하기에 생성된 악성 코드 데이터가 일반성을 갖지 못한다는 한계점을 갖고 있다.

또한 위협적인 악성코드 탐지를 위한 연구로는 LSC-GAN(Latent Semantic Controlling Generative Adversarial Networks)을 활용한 연구가 있다[7]. 기존 연구[6]가 가지고 있던 딥러닝 생성 모델이 랜덤 분포에서 데이터를 생성하지 못하는 문제를 해결하기 위해 학습 데이터를 Variational AutoEncoder의 잠재공간(Latent Space)로 투영하는 방법을 사용했다. 전통적인 방법과 비교했을 때 97%의 높은 정확도를 달성할 수 있었다.

3. ResNet-VAE 기반 변종 악성코드 패밀리 분류 방법

그림 1은 본 연구에서 제안하는 ResNet-Variational AutoEncoder를 활용한 변종 악성코드 패밀리 분류방법이다. 제안하는 방법은 (1) 데이터 수집 및 전처리 단계, (2) 변이 오토인코더 학습 단계 (3) 특징 추출 단계, (4) 변종 악성코드 분류 단계로 나누어진다.

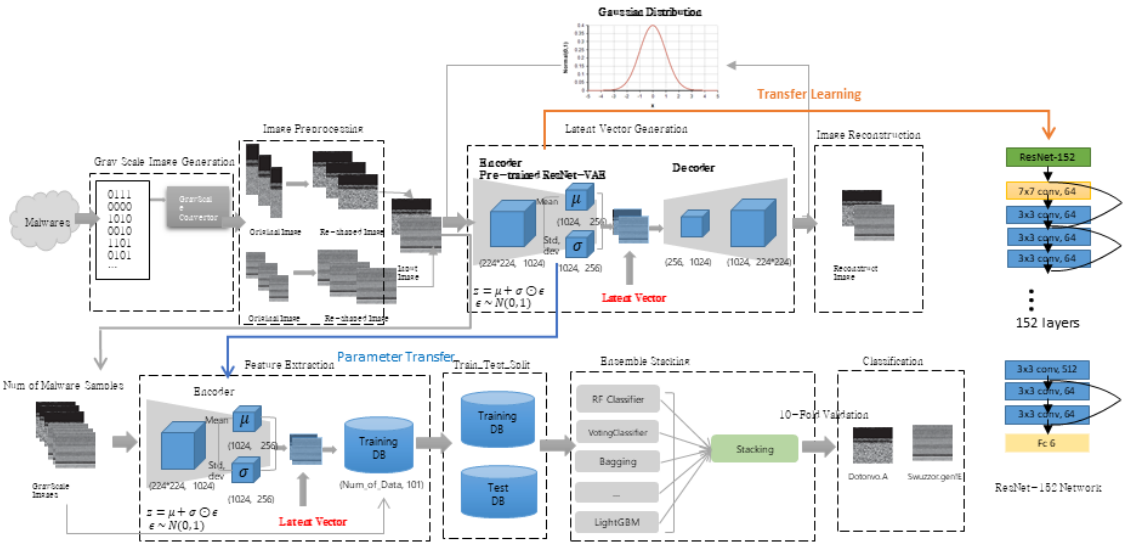
3.1 데이터 수집 및 전처리

우선 데이터 수집단계에서는 실험에 사용될 악성코드를 수집한다. 본 연구에서는 UC Santa Babara Vision Lab에서 제공하는 악성 코드의 Binary Code를 gray-scale Image로 변환시켜준 데이터를 사용했다[3]. Malimg Dataset은 고급 기계학습 알고리즘의 효율성을 평가하는데 사용된다[8, 9, 10]. Malimg Dataset에 다양한 신호처리 및 이미지 처리기술 외에도 딥러닝을 사용하여 악성코드 분류를 수행한다.

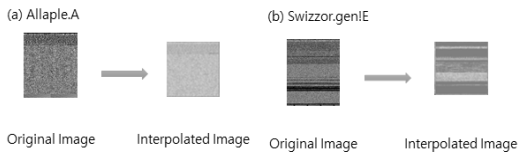
gray-scale Image생성 단계에서는 gray-scale Convertor를 이용해 악성 코드의 Binary Code를 8비트 단위로 나눠 gray-scale Image로 변경해준다. 본 연구에서는 사전에 gray-scale Convertor로 변환된 이미지를 사용하였다.

gray-scale Image 전처리 단계에서는 악성코드의 중요 정보는 보존하면서 딥러닝 모델의 입력값으로 사용할 수 있도록 데이터의 크기를 조정해준다. 본 연구에서는 중요 정보의 손실을 막으면서 악성코드 이미지 데이터의 주요 특징을 보존하기 위해 Interpolation 방법을 사용했다.

Interpolation이란 일반적으로 디지털 줌 또는 회전과 같은 공간 변환 작업 수행을 통해 이미지 데이터의 품질



(그림 1) 제안하는 방법
(Figure 1) Proposed Method



(그림 2) Bilinear Interpolation을 적용한 이미지 전처리
(Figure 2) Image preprocessing with bilinear interpolation

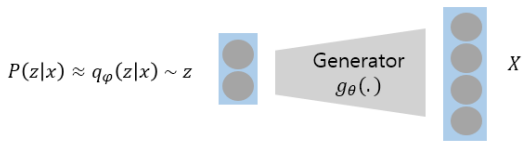
을 향상시키는데 사용된다. 많은 이미지 처리 응용프로그램에서 사용되는 방법은 Bilinear Interpolation이다[11]. 본 연구에서는 Bilinear Interpolation을 사용해 이미지 데이터 전처리를 통한 데이터 손실을 최소화 했다. 그림 2와 같이 Bilinear Interpolation을 통해 악성코드의 중요 정보를 보존하면서 Deep Variational AutoEncoder의 학습 데이터로 사용할 수 있도록 변환시켜 주었다.

또한 Variational AutoEncoder 학습 단계에서 전이 학습을 위해 데이터의 채널수를 3으로 변경시켰으며 크기를 (224, 224)로 변형해 주었다. 따라서 입력 데이터의 형상은 3차원의 224 x 224인 (3, 224, 224)가 된다.

3.2 변이 오토인코더 학습 단계

3.2.1 변이 오토인코더

Variational AutoEncoder(VAE)란 latent vector z로부터 입력데이터의 분포를 나타내는 데이터를 생성하는 생성 모델이다[2].



(그림 3) Variational AutoEncoder의 학습과정
(Figure 3) Learning process of Variational AutoEncoder

그림 3은 Variational AutoEncoder의 데이터 생성 과정을 나타낸다. 의미론적으로 가까운 샘플을 생성할 수 있는 z벡터를 정의하고 z를 생성할 수 있는 이상적인 샘플링 함수 $q_\phi(z|x)$를 정의한다. 그리고 학습을 위해 x를 given으로 주어 z벡터를 생성한다.

본 연구에서는 이러한 Variational AutoEncoder의 학습을 통해 훈련 데이터의 분포를 근사하는 변형된 데이터를 생성할 수 있었다. 학습이 완료된 Variational AutoEncoder 모델의 z벡터를 통해 훈련 데이터의 주요 특징 정보를 추출할 수 있었다.

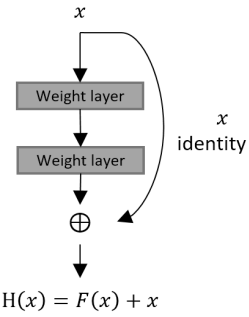
3.2.2 전이 학습

전통적인 기계 학습 방법론의 가정은 훈련 데이터와 테스트 데이터를 동일한 도메인에서 가져와 입력 특성 공간과 데이터 분포 특성이 동일하다는 것이다. 그러나 일부 실제 기계 학습 시나리오에서는 이러한 가정이 적용되지 않는다. 왜냐하면 훈련 데이터의 비용 문제와 수집의 어려움이 있기 때문이다. 따라서 서로 다른 도메인에서보다 쉽게 획득 할 수 있는 데이터로 훈련 된 고성능 학습자를 만들 필요가 있는데 이 방법론을 전이 학습이라고 한다[12].

본 연구에서는 ImageNet Dataset으로 사전학습된 ResNet-152 모델의 파라미터를 Variational AutoEncoder의 Encoder Network로 전이하는 전이 학습을 수행한다.

3.2.3 ResNet-Variational AutoEncoder

본 연구에서는 Variational AutoEncoder의 Encoder Network에서 Imagenet Data로 사전 학습된 ResNet-152모델 파라미터를 전이하는 전이 학습과 Decoder Network에서 Deconvolution 연산을 수행하는 ResNet-Variational AutoEncoder 모델을 구축하였다.



(그림 4) 잔차 블럭
(Figure 4) Residual block

ResNet은 2015년 The ImageNet Large Scale Visual Recognition Challenge(ILSVRC)에서 우승한 모델로 기존 22개의 layer로 우승한 모델보다 약 7배 많은 152개의 layer를

쌓아 더 좋은 결과를 얻을 수 있었다.

그림 4는 ResNet의 학습 방식이다[10]. ResNet 모델은 $H(x)=F(x)+x$ 함수를 최소화 하는 것을 목적으로 한다. $H(x)=F(x)+x$를 최소화 하는 것은 $F(x)=H(x)-x$를 최소화 시키는 방법과 같은데 $H(x)-x$를 잔차(Residual)라고 한다. ResNet은 Residual을 최소화 시키는 학습을 통해 152개의 깊은 레이어를 쌓을 수 있었고 높은 결과를 얻을 수 있었다[13].

본 연구에서는 Variational AutoEncoder의 Encoder Network에서 사전 학습된 ResNet-152 모델의 학습 파라미터를 전이하여 Variational AutoEncoder의 학습 성능을 향상시켰으며 훈련 과정에서 효율성도 증가시켰다. Decoder Network에서는 Deconvolution 연산을 수행해 학습 데이터의 공간 정보를 잘 보존하면서 복원의 성능을 높일 수 있었다.

3.3 특징 추출 단계

ResNet-Variational AutoEncoder의 전이 학습을 위해 1-channel gray-scale Image 데이터를 3-channel gray-scale Image 데이터로 변환해주었다.

ResNet-VAE모델의 학습 후 악성코드의 주요 특징을 추출하기 위한 방법으로 사전 학습된 ResNet-VAE 모델의 Encoder Network를 활용한다. 악성코드 패밀리 분류를 원하는 데이터를 Encoder Network에 통과시켜 악성코드 패밀리별 중요 특징값을 추출 할 수 있었다.

3.4 변종 악성코드 분류 단계

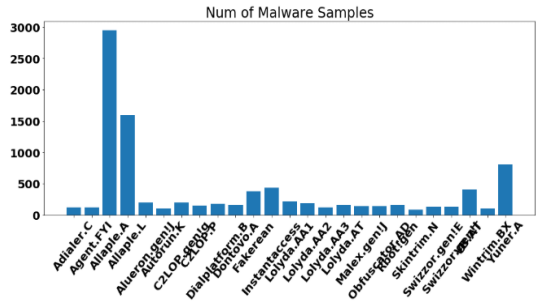
마지막으로 변종 악성코드 분류를 위해 앙상블 Stacking Classifier를 구성하였다. RandomForest Classifier, Voting Classifier, Bagging Classifier, Adaboost Classifier, Gradient Boost Classifier, XGBoost Classifier, LightGBM Classifier를 모두 조합한 Stacking Classifier를 통해 변종 악성코드 분류를 위한 학습을 수행 할 수 있었다.

4. 실험 및 평가

4.1 실험 환경 및 데이터

본 연구의 실험은 Google Colaboratory 환경에서 진행되었다. Google Colaboratory는 Google Research에서 제공하는 제품으로 웹 브라우저 환경에서 누구나 머신러닝과 데이터 분석을 할 수 있는 환경을 제공해준다[14].

본 연구에서는 Colab에서 제공하는 Tesla V100 GPU와 25.51GB의 고용량 RAM을 사용하여 연산을 수행하였다. OS는 Ubuntu 18.04.5 LTS를 사용하였다.



(그림 5) 실험에 사용된 악성코드 데이터
(Figure 5) Malware data used in the experiment

한편 본 연구에서는 실험을 위해 UC Santa Babara Vision Lab에서 제공하는 Malimg Dataset을 사용하였다.

그림 5은 실험에 사용된 악성코드 데이터를 나타낸다. 수집된 데이터는 총 9,339개의 악성코드 데이터로 구성되어 있으며 25개의 악성코드 패밀리 클래스 정보를 갖는다. 수집한 데이터는 클래스별 수집된 샘플의 수가 다른 클래스 불균형 문제를 갖고 있다.

4.2 데이터 전처리

전처리 단계에서는 두 가지 전처리를 수행했다. 우선 원본 이미지의 맥락정보는 유지하면서 데이터의 크기를 조정하기 위한 전처리를 수행하였다. 이를 위해 Interpolation 알고리즘을 사용하였다.

또한 ResNet-VAE의 Encoder Network 전이학습을 위해 채널수를 조정해 주었다. 전처리 결과 3-channel gray-scale Image data를 얻을 수 있었다. 따라서 학습에 사용되는 Malimg 데이터는 Interpolation과 3-channel 변환을

거쳐 (3, 224, 224)의 형상을 갖게 된다. 한편 연산을 위해 gray-scale Image 데이터의 값을 Tensor로 변환해 주었다. Tensor Scaling을 통해 각각의 픽셀 Tensor 값의 범위를 0에서 1사이의 값으로 scaling 하였다.

4.3 ResNet-Variational AutoEncoder 모형 구축

본 연구에서는 ResNet-VAE를 활용한 변종 악성코드 특징 추출을 위해 9,339개의 데이터를 7,471개의 훈련 데이터셋과 1,868개의 검증 데이터셋으로 나눠주었

(표 1) 변종악성코드 패밀리 분류기별 precision, recall, F1-score, Accuracy

	precision	recall	F1-score (weighted avg)	Accuracy
ResNet-VAE+SVC	97.54	97.54	97.54	97.54%
ResNet-VAE+RandomForest	98.58	98.45	98.49	98.45%
ResNet-VAE+XGBoost	98.74	98.61	98.63	98.61%
ResNet-VAE+LightGBM	98.68	98.55	98.57	98.55%
ResNet-VAE+Stacking	98.79	98.66	98.68	98.66%

다.이렇게 8대 2의 비율로 나눠진 데이터를 통해 ResNet-VAE를 학습하고 검증하는 과정을 거쳤다.

ResNet-VAE를 학습 과정에서 오버피팅을 방지하기 위해 Train Loss와 Test Loss를 비교해가며 모델을 학습시켰다. 그림 6은 ResNet-Variational AutoEncoder의 Epoch별 Train Loss, Test Loss를 출력한 그래프이다. 그Variational AutoEncoder 모델은 Generative Model의 특성상 학습 초반에는 Test Loss값이 발산하는 경향이 있었으나 40 Epoch을 기점으로 학습이 잘 수행되는 것을 확인할 수 있었다.

Test Image Data를 제공한 뒤 Reconstruction되는 Data를 확인해본 결과 그림 7과 같이 학습 데이터의 특징을 갖으면서 조금은 변형된 새로운 악성코드를 생성하는 것을 확인할 수 있었다.

4.4 변종 악성코드 특징 추출

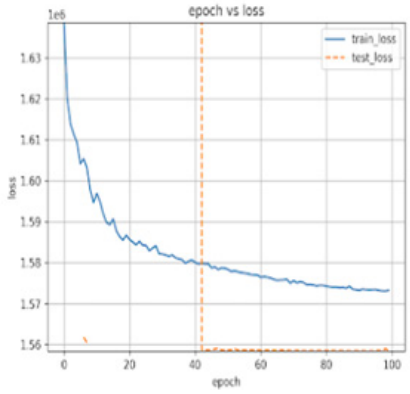
악성코드의 주요 특징을 추출하는 방법으로 사전 학습된 ResNet-VAE 모델의 Encoder Network를 활용하였다. Variational AutoEncoder의 Encoder Network는 학습 과정에서 학습 데이터의 잠재적 특징을 추출하고 이를 기반으로 하여 기존 데이터와 같은 분포를 갖으면서 조금은 변형된 새로운 데이터를 생성한다. 이러한 특성을 활용해 학습 데이터를 Encoder Network에 통과시켜 256개의 Feature를 갖는 데이터셋을 구성할 수 있었다.

4.5 변종 악성코드 분류

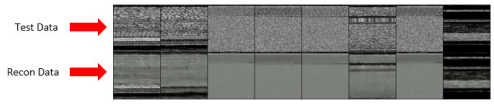
분류 단계에서는 9,339개의 악성코드 데이터를 6대 2대 2의 비율로 나눠주었다. 따라서 훈련 데이터로는 5,603개의 데이터가 사용되었으며 검증 데이터 1,868개와 테스트 데이터 1,868가 각각 모델 검증과 최종 결과 테스트에 사용되었다.

변종 악성코드 분류를 위한 방법으로 sklearn에서 제공하는 SVC, RandomForest, XGBoost, LightGBM, Stacking Classifier 등 다양한 분류기를 통해 실험을 수행하였다.

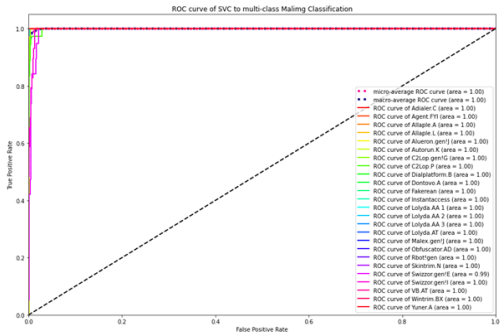
ResNet-VAE+SVC, ResNet-VAE+RandomForestm ResNet-VAE_XGBoost, ResNet-VAE+LightGBM, ResNet-VAE+Stacking 등 학습된 ResNet-VAE의 Encoder Network로부터 추출한 Feature 데이터를 기반으로 변종 악성코드를 SVM 분류기와 앙상블 분류기를 사용하여 분류하였다. 표 1은 분류기별 실험 결과를 나타낸다.



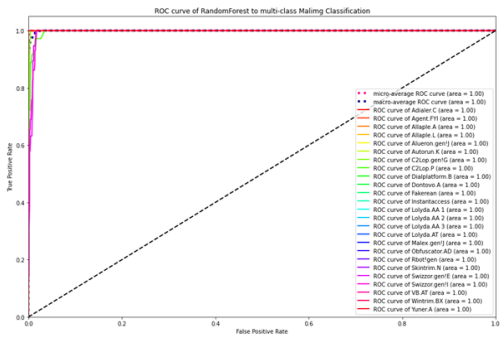
(그림 6) ResNet-VAE 학습데이터와 평가데이터의 손실값
(Figure 6) ResNet-VAE Train, Test Loss



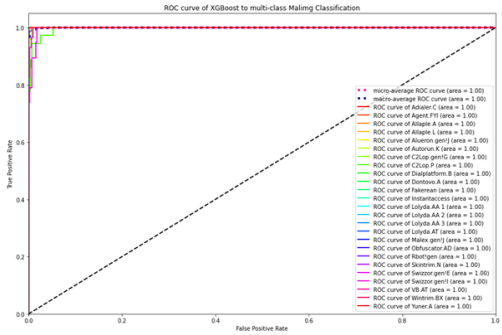
(그림 7) 전이 학습에 사용된 모델의 복원값
(Figure 7) Reconstuction result of the model used for transfer learning



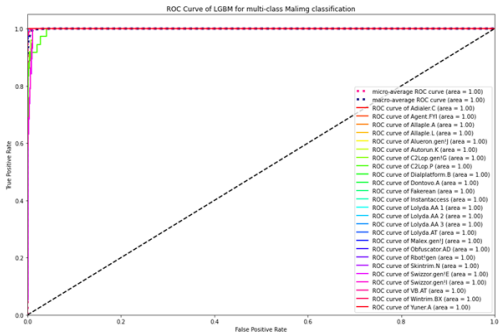
(그림 8) SVC의 ROC 곡선
(Figure 8) ROC curve of SVC to multi-class Malimg Classification



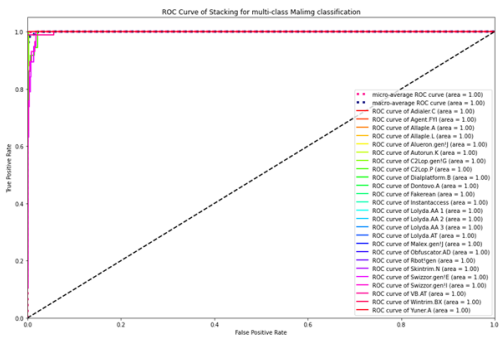
(그림 9) RandomForest의 ROC 곡선
(Figure 9) ROC curve of RandomForest to multi-class Malimg Classification



(그림 10) XGBoost의 ROC 곡선
(Figure 10) ROC curve of XGBoost to multi-class Malimg Classification



(그림 11) LightGBM의 ROC 곡선
(Figure 11) ROC curve of LightGBM to multi-class Malimg Classification



(그림 12) Stacking의 ROC 곡선
(Figure 12) ROC curve of Stacking to multi-class Malimg Classification

4.6 평가

변종 악성코드 분류 정확도 및 평가를 위해 각각의 분류기에 대한 Accuracy, Precision, Recall, F1-Score 등의 지표를 살펴보았다. 실험에서 사용한 데이터의 경우 클래스 불균형 문제를 갖고 있기 때문에 평가 지표로 F1-score와 ROC curve 등을 사용했다.

F1-score란 Precision과 Recall값의 가중평균이다. 따라서 F1-score는 위양성(False Positive)과 위음성(False Negative)를 고려한다. 수집한 데이터의 클래스가 불균형한 경우 모델을 평가하는 데 있어서 Accuracy보다 유용한 지표로 사용할 수 있다. 한편 ROC curve란 분류문제에서 모델을 평가하는 지표이다. ROC는 확률 곡선을 나타내며 확률 곡선 아래 영역을 나타내는 AUC는 분류의 정도를 나

타낸다. 본 연구에서 수집한 악성코드의 경우 클래스 불균형 문제가 있기 때문에 F1-score와 ROC curve를 통해 모델의 결과를 해석하였다.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_1 - score = 2 * \frac{precision * recall}{precision + recall}$$

표 1의 평가 지표에서 확인할 수 있듯이 ResNet-VAE+Stacking 모형의 경우 98.66%의 Accuracy와 98.68의 f1-score로 가장 성능이 좋은 것으로 나타났다. 선형 분류기인 SVM 모형부터 RandomForest, XGBoost, LightGBM에 이르는 앙상블 모형까지 살펴본 결과 선형 모형보다는 앙상블 모형의 결과가 좋은 것을 확인할 수 있었다. 특히 RandomForest Classifier, Voting Classifier, Bagging Classifier, Adaboost Classifier, Gradient Boost Classifier, XGBoost Classifier, LightGBM Classifier를 모두 조합한 Stacking Classifier는 Accuracy 뿐만아니라 F1-score에서도 조금 더 나은 결과를 가져다 주었다. 한편그림8, 9, 10, 11, 12, 13는 각각의 분류기에 대한 ROC curve 지표를 나타낸다. ROC 곡선을 살펴본 결과 Stacking Classifier가 변종 악성코드를 가장 잘 분류하는 것을 확인할 수 있었다.

5. 결론 및 향후 연구

본 연구에서는 악성코드 분석 전문가에 의한 전통적인 특징 기반 악성코드 분류 방법의 문제점을 해결하기 위해 ResNet-Variational AutoEncoder기반 변종 악성코드 분류 방법을 제안하였다. Variational AutoEncoder의 학습 과정에서 변종 악성코드의 특징을 학습할 수 있었으며 ImageNet data로 사전 학습 된 ResNet-152모델의 파라미터를 Encoder Network로 전이하여 Variational AutoEncoder의 학습 성능 향상과 학습 과정에서의 효율성을 높일 수 있었다. 변종 악성코드 분류를 위해 Encoder Network로 악성코드의 중요 특징을 추출하고 Stacking Classifier를 학습한 결과 98.66%의 Accuracy와 98.68의 F1-Score를 얻을 수 있었다.

하지만 본 연구에서는 악성코드의 주요 분석 방법인 Static Analysis만을 사용했기에 실제 악성코드 동작 환경에서는 악성코드의 모든 행위를 탐지 및 분류할 수 없다는 한계점을 가지고 있다. 향후 연구에서는 Sandbox 환경에서 악성코드를 실시간으로 분석하고 이를 Static Analysis 모델의 학습과 연계하는 Hybrid Method를 수행할 것이다 [15].

참고문헌(Reference)

[1] Moser, Andreas, Christopher Kruegel, and Engin Kirda, “Limits of static analysis for malware detection”, Twenty-Third Annual Computer Security Applicaitons Conference IEEE, 2007. https://doi.org/10.1109/acsac.2007.21

[2] Kingma, Diederik P, and Max Welling, ″Auto-encoding variational bayes″, arXiv preprint arXiv:1312.6114, 2013. https://doi.org/10.18653/v1/2020.coling-main.458

[3] Nataraj, Lakshmanan et al., “Malware images: visualization and automatic classification”, Proceedings of the 8th international symposium on visualization for cyber security, 2011. https://doi.org/10.1145/2016904.2016908

[4] Nataraj and B. S. Manjunath, “SPAM: Signal processing to analyze malware”, arXiv, 2016. https://doi.org/10.1109/msp.2015.2507185

[5] Chalapathy, Raghavendra, and Sanjay Chawla, “Deep learning for anomaly detection: A survey”, arXiv preprint arXiv:1901.03407, 2019. https://arxiv.org/abs/1901.03407

[6] Jin-Young Kim, Seok-Jun Bu, and Sung-Bae Cho, ″Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders″, Information Sciences 460, pp83-102, 2018. https://doi.org/10.1016/j.ins.2018.04.092

[7] Jin-Young Kim, and Sung-Bae Cho, ″Detecting intrusive malware with a hybrid generative deep learning model.″, International Conference on Intelligent Data Engineering and Automated Learning. Springer, 2018. https://doi.org/10.1007/978-3-030-03493-1_52

[8] Luo, Jhu-Sin, and Dan Chia-Tien Lo, “Binary malware image classification using machine learning with local binary pattern”, IEEE International Conference on Big Data, IEEE, 2017.
https://doi.org/10.1109/bigdata.2017.8258512

[9] Zhou, Xin, Jianmin Pang, and Guanghui Liang. "Image classification for malware detection using extremely randomized trees." 2017 11th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID). IEEE, 2017.
https://doi.org/10.1109/icasid.2017.8285743

[10] Vasan, Danish, et al. "IMCFN: Image-based malware classification using fine-tuned convolutional neural network architecture." Computer Networks 171, 2020.
https://doi.org/10.1016/j.comnet.2020.107138

[11] Gribbon, Kim T, and Donald G. Bailey, "A novel approach to real-time bilinear interpolation", Proceedings. DELTA 2004. Second IEEE International Workshop on Electronic Design, Test and Applications. IEEE, 2004.
https://doi.org/10.1109/delta.2004.10055

[12] Weiss, Karl, Taghi M. Khoshgoftaar, and DingDing Wang, "A survey of transfer learning.", Journal of Big data, 2016.
https://doi.org/10.1186/s40537-016-0043-6

[13] He, Kaiming, et al., "Deep residual learning for image recognition", Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.
https://doi.org/10.1109/cvpr.2016.90

[14] Bisong, Ekaba, "Google colaboratory", Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA, 2019.
https://doi.org/10.1007/978-1-4842-4470-8_7

[15] Venkatraman, Sitalakshmi, Mamoun Alazab, and R. Vinayakumar, "A hybrid deep learning image-based analysis for effective malware detection", Journal of Information Security and Applications 47, 2019.
https://doi.org/10.1016/j.jisa.2019.06.006

◐ 저 자 소 개 ◑

이 영 전(Young-jeon Lee)
2019년 가천대학교 컴퓨터공학과(공학사)
2019년~현재 가천대학교 일반대학원 소프트웨어학과 석사과정
관심분야 : 정보보호, 인공지능, 빅데이터
E-mail : leeyj0511@naver.com

한 명 묵(Myung-Mook Han)
1980년 연세대학교 공과대학(공학사)
1987년 뉴욕공과대학교 대학원 컴퓨터공학과(공학석사)
1997년 오사카시립대학교 대학원 정보공학부(공학박사)
1998년~현재 가천대학 소프트웨어학과 교수
관심분야 : 정보보호, 알고리즘, 데이터 마이닝
E-mail : mmhan@gachon.ac.kr