Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.6, Dec. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.6.1067

하이브리드 데이터셋을 이용한 악성코드 패밀리 분류*

최 서 우,[1†] 한 명 진,[1] 이 연 지,[2] 이 일 구[3‡]
[1,2,3]성신여자대학교 (학생, 대학원생, 교수)

Classification of Malware Families Using Hybrid Datasets*

Seo-Woo Choi,[1†] Myeong-Jin Han,[1] Yeon-Ji Lee,[2] Il-Gu Lee[3‡]
[1,2,3]Sungshin Women's University (Undergraduate student, Graduate student, Professor)

요 약

최근 변종 악성코드가 증가하면서 사이버 해킹 침해사고 규모가 확대되고 있다. 그리고 지능형 사이버 해킹 공격에 대응하기 위해 악성코드 패밀리를 효과적으로 분류하기 위한 기계학습 기반 연구가 활발히 진행되고 있다. 그러나 기존의 분류 모델은 데이터셋이 난독화되거나, 희소한 경우에 성능이 저하되는 문제가 있었다. 본 논문에서는 ASM 파일과 BYTES 파일에서 추출한 특징을 결합한 하이브리드 데이터셋을 제안하고, FNN을 사용하여 분류 성능을 평가한다. 실험 결과에 따르면 제안하는 방법은 단일 데이터셋에 비해 약 4% 향상된 성능을 보였으며, 특히 희소한 패밀리에 대해서는 약 30%의 성능 향상을 보였다.

ABSTRACT

Recently, as variant malware has increased, the scale of cyber hacking incidents is expanding. To respond to intelligent cyberhacking attack, machine learning-based research is actively underway to effectively classify malware families. However, existing classification models have problems where performance deteriorates when the dataset is obfuscated or sparse. In this paper, we propose a hybrid dataset that combines features extracted from ASM files and BYTES files, and evaluate classification performance using FNN. As a result of the experiment, the proposed method showed performance improvement of about 4% compared to a single dataset, and in particular, performance improvement of about 30% for rare families.

Keywords: Malware Classification, Hybrid data, Feature Selection

I. 서 론

정보통신기술의 발전과 함께 인간은 언제 어디서나 사이버 세상에 연결되어 정보를 주고 받을 수 있게 되었고, 디지털화된 전통 산업은 사이버 범죄의 주요 공격 대상이 되고 있다[1]. 특히 IoT (Internet of Things) 기술이 상용화되고 수요가 증가함에 따라 2025년에는 220억대의 IoT 장치가 산업에 활용될 거라고 추산되고 있다. 이러한 IoT 기기는 계산 비용 및 메모리와 같은 자원이 제한되어 있어 다양한 사이버 공격에 취약하며, 이에 따라 IoT 환경에서의 악성코드 탐지 연구의 필요성이 대두되고 있다[2][3].

2022년 상반기에 발생한 변종 악성코드의 수는 27억 5천만 건 이상이며[4], 발전하는 악성코드로 인한 사이버 범죄 피해는 2025년에 10조 5천억 달러에 달할 것이라 분석되고 있다[5]. 증가하는 악성코드를 탐지하기 위해 다양한 선행 연구들이 진행되었다[6-11]. 그중에서도 가장 널리 알려진 기술은

Received(10. 17. 2023), Modified(11. 27. 2023), Accepted(11. 27. 2023)
* 본 논문은 2023년도 정부(산업통상자원부)의 재원으로 한국산업기술진흥원의 지원(P0008703, 2023년 산업혁신인재성장지원사업), 2023년도 과학기술정보통신부 및 정보통신기획평가원의 ICT혁신인재4.0 사업(IITP-2022-RS-2022-00156310)의 지원을 받아 연구되었음.
† 주저자, 20211104@sungshin.ac.kr
‡ 교신저자, iglee@sungshin.ac.kr(Corresponding author)

악성코드의 동작 패턴을 데이터베이스에 저장하고 악성코드의 행위와 데이터베이스를 대조하여 일치할 경우에 악성으로 분류하는 시그니처 기반 악성코드 탐지 방식이다[6]. 시그니처 기반 탐지 방법은 데이터베이스에 존재하는 악성코드를 빠르고 단순하게 탐지할 수 있다는 장점이 있지만, 빠르게 진화하는 변종 악성코드 탐지에 취약하다는 한계점이 있다[7].

변종 악성코드를 정확하고 빠르게 탐지하기 위해 악성코드를 직접 실행하지 않고 소스코드를 분석하는 정적 분석 방식에 기계학습을 접목하는 연구가 주목받고 있다[8-12]. 특히 시각화한 이미지를 학습하는 합성곱 신경망(CNN, Convolutional Neural Network) 분류 알고리즘의 활용도가 높다[10-12]. 그러나 종래의 CNN 기반 이미지 분류 연구들은 희소한 라벨의 편향 데이터 분류 성능이 낮은 문제가 존재하며[14][15], 악성코드 탐지 성능에만 초점을 맞추고 한정된 자원을 고려하지 않아서 IoT 장치에는 적합하지 않다.

본 연구에서는 앞서 언급한 종래 연구들의 한계점을 해결하기 위해 악성코드의 BYTES 파일과 ASM 파일을 결합한 하이브리드 데이터셋을 제안한다. 이는 BYTES 파일을 Gray-scale 이미지로 만들고 1차 학습을 진행해 나온 결과 값과 ASM 파일에서 추출한 OP Code sequence에 2-gram을 적용한 값을 결합한다. 이후 FNN(Feed-forward Neural Network) 모델로 2차 학습을 진행한 뒤 다양한 평가 지표로 성능을 평가한다.

본 논문의 주요 기여점은 다음과 같다.

- 종래의 희소 데이터 및 난독화 데이터의 분류 성능 저하 문제를 극복하는 새로운 데이터 전처리 방법을 제안한다.
- IoT 환경의 한계점인 한정된 자원을 고려하여 경량 악성코드 분류 모델을 제안하고 기존의 성능-자원 간의 trade-off 문제를 개선한다.
- 악성코드 패밀리 분류 모델 평가 프레임워크를 제안하고, 제안한 방법이 종래 방법 대비 희소 데이터 및 난독화된 데이터의 분류 정확도가 개선됨을 보였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 악성코드 분류에 관한 종래 연구를 분석한다. Ⅲ장에서는 제안하는 하이브리드 기반 전처리 방법을 소개하며, Ⅳ장에서는 하이브리드 데이터셋 기반의 악성코드 패밀리 분류 모델을 제안한다. Ⅴ장에서는 실험 결과를 분석하고, 마지막으로 Ⅵ장에서 결론을 맺는다.

Ⅱ. 관련 연구

최근 악성코드를 시각화하여 분류하는 연구가 활발히 이루어지고 있다. Ahmed Bensaoud의 연구[12]에서는 악성코드 BYTES 파일을 Gray-scale로 시각화하는 전처리 방법을 제안했다. 제안하는 방법은 높은 분류 정확도를 보였지만 파일을 이미지화하는 과정에서 불필요한 데이터까지 학습하였으며, 이미지 파일의 특성상 BYTES 파일에 비해 큰 메모리를 사용하므로 자원 측면에서도 한계를 갖는다.

이러한 종래의 문제를 해결하는 데 필요한 특징만 추출하는 방법에 관한 다양한 연구가 이루어지고 있다. Hanqi Zhang의 연구[13]에서는 ASM 파일의 OP Code sequence에서 2, 3, 4-gram을 추출하고, TF-IDF를 계산해 데이터셋의 feature로 사용하는 전처리 방법을 제안하였다. 이러한 전처리 과정은 학습에 효과적인 feature만을 선별하여 메모리, 학습시간 측면의 효율성을 보완하였지만, 91.43%의 분류 정확도로 전체 데이터셋을 사용하는 방법에 비해 비교적 낮은 분류 성능을 보인다.

악성코드 분류 성능을 향상시키기 위한 Rajasekhar Chagantia의 연구[14]에서는 이미지화한 BYTES 파일에 이미지 크기 조정 작업을 적용함으로써 [13]의 연구 대비 향상된 분류 성능을 갖는 악성코드 분류 모델을 제안하였다. 해당 실험은 이미지 폭이 고정된 악성코드에서 최대 99%의 전체 분류 성능을 보였으나, 난독화된 Obfuscator.ACY 패밀리의 특성을 충분히 분석하지 못해 이를 Ramnit 패밀리로 오분류하는 한계점이 존재한다. Ketan Gupta의 연구[15]에서는 BYTES 파일을 이미지화하고 ANN 알고리즘을 사용하여 패밀리를 분류하는 연구를 수행하였다. 해당 모델은 전체 데이터셋에서 90%에 가까운 분류 성능을 갖는다. 그러나 불균형한 데이터 중 42개의 데이터가 포함된 희소한 simda 패밀리에 대해 약 2배의 성능 저하를 보인다. [14]와 [15] 연구는 공통적으로 범용적인 데이터셋에서는 좋은 성능을 유지하지만 희소하거나 난독화된 데이터 등 열악한 데이터에 대해서는 좋은 성능을 유지하지 못했다.

희소 데이터셋의 분류 성능 개선을 위해 Xuejin Zhu의 연구[16]에서는 BYTES 파일과 OP Code 두 가지 데이터의 특징을 융합하고 이미지화하는 하

이브리드 데이터셋을 제안하였다. 제안하는 모델은 희소한 패밀리의 성능을 약 2배 개선하여 83%의 정확도를 달성하였으며, BYTES 파일만을 사용하는 모델에 비해 난독화된 패밀리의 분류 성능을 9% 개선하여 제안한 하이브리드 데이터셋의 성능을 입증하였다. 그러나 ASM의 전처리 과정 중 추출된 n-gram의 과도한 Feature 수 증가에 따른 계산 처리부하 문제의 해결방안을 언급하지 않았다는 한계가 존재한다.

이와 같이 종래의 연구 중 단일 데이터셋을 이용한 분류 모델의 경우 희소하거나 난독화된 데이터에 대해 전반적으로 낮은 분류 성능을 갖는다. 반면 두 가지 데이터의 특징을 융합하여 다중 데이터를 사용한 연구의 경우 희소하거나 난독화된 데이터에 대한 성능이 보완되어 높은 분류 정확도를 보였다.

III. 하이브리드 기반 전처리 기법

종래의 BYTES 파일을 이미지화하는 전처리 방식을 단독으로 사용하였을 때, 전체적으로 높은 성능을 보이지만 희소하거나 난독화된 데이터셋에 대해서 성능이 저하되었다. 또한 ASM 파일의 OP Code sequence에서 특징을 추출하는 전처리 방법을 단독으로 사용하였을 때는 메모리 사용량 측면에서 개선되었지만 BYTES 파일을 이미지화하는 방식보다 낮은 성능을 보였다.

따라서 본 연구에서는 앞서 언급된 종래 연구의 한계점을 보완하기 위해 BYTES파일과, ASM파일의 각 특징을 결합한 하이브리드 데이터셋을 제안한다. 제안하는 전처리된 하이브리드 데이터셋은 다양한 특징을 고려하여 높은 악성코드 분류 정확도를 달성하며, 희소하거나 난독화된 데이터셋에서의 분류 성능 향상을 보인다.

3.1 BYTES 파일 전처리

본 연구에서 제안하는 BYTES 파일의 전처리 기법은 Fig. 1.과 같다. 우선 악성코드의 바이너리 데이터를 Gray-scale 이미지로 변환하고 CNN 모델로 학습시킨다. 악성코드의 바이너리는 0과 1로 구성되며 이를 8비트 단위의 Gray-scale 이미지로 변환한다[16].

본 연구에서 사용한 Microsoft Malware Classification Challenge (BIG 2015) 데이터셋[23]의 각 패밀리를 시각화한 결과는 Fig. 2. 와 같다. Gray-scale 이미지는 CNN의 학습 데이터로

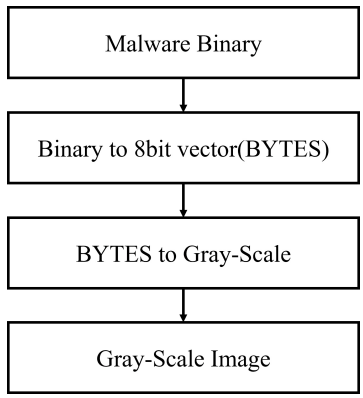

Fig. 1. BYTES file preprocessing process

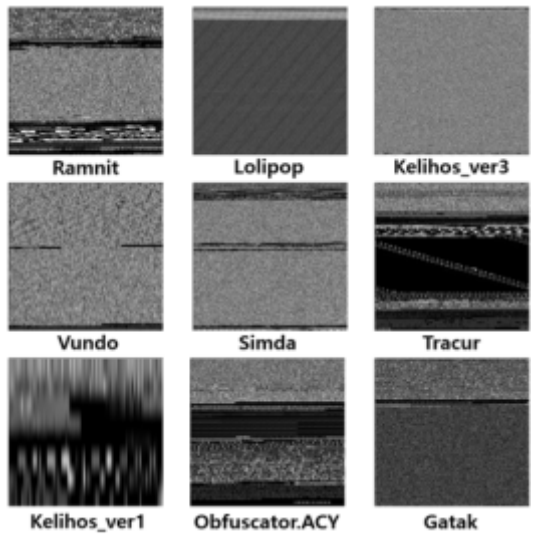

Fig. 2. Malware family Gray-scale imaging

Table 1. Structure of CNN

Convolution Neural Networks	
Layer (type)	**Output Shape**
Conv2D	(None, 64, 64, 32)
MaxPooling2D	(None, 32, 32, 32)
Dropout	(None, 32, 32, 32)
Conv2D	(None, 16, 16, 64)
MaxPooling2D	(None, 16, 16, 64)
Dropout	(None, 16384)
Flatten	(None, 128)
Dense	(None, 128)
Dropout	**(None, 9)**

사용하기 위해 고정된 64×64의 크기로 조정하고 해당 이미지를 CNN 모델로 학습하여 라벨 별 분류 확률을 추출한다[16]. 특징 추출에 사용한 CNN의 구조는 Table 1.와 같다.

3.2 ASM 파일 전처리

ASM 파일의 전처리 방법은 Fig. 3.과 같다. 정적 특징에서 상위 10,000개의 2-gram을 추출한 후 TF-IDF (Term Frequency - Inverse Document Frequency) 빈도 행렬을 계산한다. 이후 고차원 행렬에 LSA(Latent Semantic Analysis)를 적용하여 중요한 특징을 선택한다. N-gram은 N 개의 단어의 sequence로 N-gram을 사용하여 악성코드의 특징을 추출하는 것이 효과적인 분석 방법으로 알려져 있다[17].

TF-IDF는 단어의 빈도를 기반으로 중요도를 계산하여 가중치로 부여하는 방법이다. TF는 단어의 빈도, DF는 전체 문서 중 해당 문서가 출현한 문서의 수를 의미한다.

식(1)은 TF 값 계산 과정으로, $f(t,d)$는 문서(d)에 출현한 특정 단어(t)의 빈도수를, $\max\{f(w,d):w\in d\}$ 는 문서 내에 포함된 모든 단어(w)의 빈도수 중 가장 큰 값을 나타낸다. 식(2)는 DF의 역수인 IDF를 표현한 것이다. 식 (2)에서 $|D|$는 전체 문서의 수를, $|d\in D: t\in d|$는 특정 단어(t)가 출현한 문서의 수를 나타낸다. $|d\in D: t\in d|$을 $|D|$로 나눈 뒤 역수를 취하고, 값 간의 편차를 줄이기 위해 로그를 취한다[18]. 식(3)은 TF-IDF의 계산식으로, 단어의 빈도 값과 문서 빈도에 역수를 취한 값을 곱하여 특정 문서 내에서 단어의 중요도를 고려할 수 있다[19].

$$tf(t,d) = \frac{f(t,d)}{\max\{f(w,d) : w \in d\}} \tag{1}$$

$$idf(t,D) = \log\frac{|D|}{|\{d\in D : t\in d\}|} \tag{2}$$

$$tf-idf(t,d,D) = tf(t,d)\times idf(t,D) \tag{3}$$

TF-IDF와 같은 단어 빈도수를 기반으로 하는 가중치 계산 기법은 단어가 속한 토픽을 고려하지 못한다는 한계점이 있다[20]. 이러한 한계점을 개선한 알고리즘인 LSA 기반의 전처리는 잠재된 의미를 분석함으로써 상관성이 높은 데이터를 추출할 수 있고, 학습시간과 메모리 사용량을 효과적으로 줄일 수 있다[21].

SVD(Singular Value Decomposition)는 입력 행렬을 단어 벡터 행렬(U), 문서 벡터 행렬(V), 대각행렬($\sum$)로 분해하는 행렬 분해 방법이다. 이에 대해 k개의 차원으로 축소하는 방법을 TSVD (Truncated Singular Value Decomposition)라고 한다. TF-IDF를 통해 생성한 행렬에 대해 TSVD를 적용하여 차원을 축소한다. 여기서 m은 단어의 수이며, n은 문서의 수를 나타낸다[22].

식 (4)은 행렬 A와 TSVD의 관계를 나타낸다. SVD를 통해 $m\times t$인 직교 행렬 U_k와 $n\times t$인 행렬 V_k, $t\times t$인 대각 행렬 S_k를 얻는다. 여기서 t는 $\min(m,n)$을 의미하며, U_k 와 V_k는 행렬 A의 최상의 근사치를 생성하는 k차원 행렬이다. 행렬 V_k의 각 행은 축소한 k차원 문서에 대한 특징 벡터이며 이때 k 값을 패밀리의 개수로 지정한다. 본 논문에서는 9개의 차원으로 축소하였다.

$$A \approx U_k.S_k.V_k^T \tag{4}$$

패밀리에 관한 정보를 가지고 있는 행렬 V_k^T의 각 행에 대하여 상위 F개의 특징을 추출하는 과정을 Top F라고 명명한다. 이를 통해 최적의 성능을 보이는 특징 F값을 결정하고 차원 수를 줄여 주요 특징만을 학습에 반영한다.

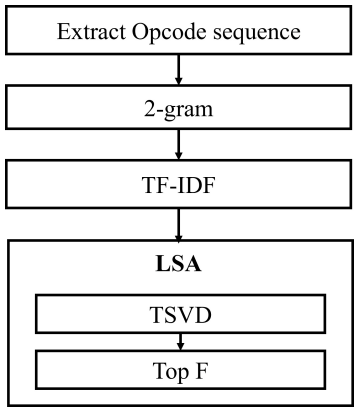

Fig. 3. ASM file preprocessing sequence

IV. 제안 모델

4.1 데이터셋

제안한 방법과 종래의 방법의 성능을 비교 평가하기 위해 Microsoft Malware Classification Challenge (BIG 2015)를 사용했다[23]. 이 데이터셋은 악성코드 변종 파일을 16진수로 변환한 10,868개의 BYTES 파일과 10,868개의 ASM 파일로 구성되어 있다. 이 데이터셋에서는 Ramnit, Lollipop, Kelihos_ver3, Vundo, Simda, Tracur, Kelihos_ver1, Obfuscator.ACY, Gatak의 9가지 악성코드 파일 패밀리로 구성된다.

본 연구에서는 이 데이터셋 중 값이 불명확한 8개의 파일을 제외한 10,860개의 BYTES 파일과 ASM 파일을 사용하였다. 패밀리 별 데이터의 분포는 Fig. 4.와 같이 나타나며, 패밀리 별 데이터의 개수와 라벨링은 Table 2.와 같다. 특히, Class 5는 총 데이터셋의 개수가 42개로 희소한 패밀리이며, Class 8은 난독화된 파일에 대한 패밀리이다.

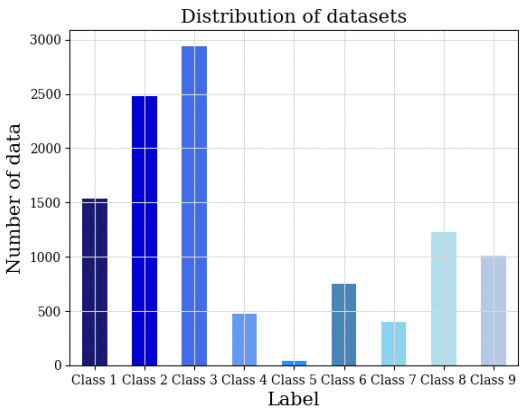

Fig. 4. Distribution of the datasets

Table 2. Number and labeling of samples

Family	Number	Type	Label
Ramnit	1,533	Worm	Class 1
Lolipop	2,478	Adware	Calss 2
Kelihos_ver3	2,942	Backdoor	Class 3
Vundo	475	Trojan	Class 4
Simda	42	Backdoor	Class 5
Tracur	751	Trojan Downloader	Class 6
Kelihos_ver1	398	Backdoor	Class 7
Obfuscator.ACY	1,228	Obfuscated malware	Class 8
Gatak	1,013	Backdoor	Class 9
Total	10,860		

4.2 악성코드 패밀리 분류

원본 데이터셋에서 추출한 특징을 CSV 파일 형식으로 병합하여 하이브리드 데이터셋을 생성한다. 이 데이터셋은 최종 학습 모델에 입력으로 사용되기 위해 기본적인 신경망인 FNN을 활용하여 평가된다. 전체 과정의 구조도는 Fig. 5.와 같다.

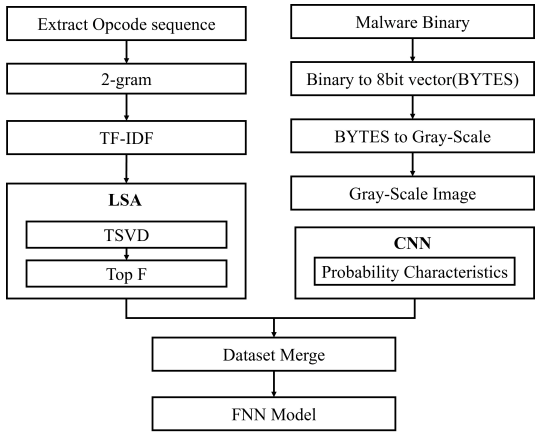

Fig. 5. Overall process architecture

V. 실험 및 평가

5.1 실험 환경

제안한 방법의 효과를 검증하기 위해 제안한 전처리 방법을 단독으로 사용한 단일 데이터셋의 성능과 두 가지 전처리 방법을 병합한 하이브리드 데이터셋의 성능을 비교 분석한다.

실험은 총 세 단계로 구성된다. 첫 번째 실험은 악성코드의 BYTES 파일을 Gray-scale 이미지로 변환하여 CNN으로 학습시킨 분류기의 성능을 평가해 각 패밀리로 분류할 확률인 probability를 추출한다. 이때 분류 확률은 입력된 이미지가 각 라벨로 분류될 확률을 의미하며 scikit-learn 모듈의 predict_proba 함수를 사용하였다. 두 번째 실험은 악성코드의 ASM 파일에서 OP Code sequence를

추출하여 2-gram과 TF-IDF 계산을 수행하고 LSA를 통해 feature를 선택한다. 세 번째 실험은 첫 번째 실험에서 추출한 probability와 두 번째 실험에서 선택된 feature들을 결합한 하이브리드 데이터셋을 FNN 모델로 학습하고 성능을 평가한다.

종래 기법 및 제안하는 기법은 windows 11에서 파이썬으로 구현하였으며, 상세 실험 환경은 Table 3.과 같다. 두 번째 실험과 세 번째 실험에 사용한 FNN의 구조는 Table 4.와 같다.

Table 3. Experiment environment

Classification	Version
OS	Windows 11
CPU	Intel(R) Core(TM) i9-10850K
RAM	32GB
Python	3.10.5
Tensorflow	2.11.0
Keras	2.11.0

Table 4. Structure of FNN

Feedforward Neural Network	
Layer (type)	Output Shape
Flatten	(None, 38)
Dense	(None, 128)
Dropout	(None, 128)
Dense	(None, 9)

5.2 실험 결과

본 논문에서는 정확도, 정밀도, 재현율, f1-score 등의 평가 지표를 통해 성능을 평가하였다. TP(True Positive)는 실제 양성 라벨을 양성으로 바르게 예측한 결과를 의미하며, TN(True Negative)은 실제 음성 라벨을 모델이 음성이라 바르게 예측한 경우를 의미한다. FP(False Positive)는 실제 음성 라벨을 모델이 양성이라 잘못 예측한 경우를 의미하며, FN(False Negative)은 실제 양성 라벨을 모델이 음성이라 잘못 예측한 경우를 의미한다[24]. 평가에 사용된 수식(5)는 다음과 같다.

$$
\begin{aligned}
\text{정확도} &= \frac{TP+TN}{TP+TN+FP+FN} \\
\text{정확도} &= \frac{TP}{TP+FP} \\
\text{재현율} &= \frac{TP}{TP+FN} \\
f1-score &= 2\times\frac{\text{정밀도}\times\text{재현율}}{\text{정밀도}+\text{재현율}}
\end{aligned}
\tag{5}
$$

앞서 언급한 수식들로 평가한 제안 모델의 전반적인 분류 성능은 Fig. 6.과 같으며, 손실 함수로 평가한 성능은 Fig. 7.과 같다. 본 연구에서 제안한 하이브리드 데이터셋의 정확도는 BYTES Gray-scale보다 약 3% 증가하였으며, ASM 전처리 방법에 비해 정확도가 약 2% 증가하였다. Fig. 7.의 epoch에 따른 loss값을 보면 하이브리드 데이터셋은 BYTES Gray-scale에 비해 손실 값이 약 4배 감소하였고, ASM 전처리 방법 보다 약 2.5배 감소하였다.

학습한 모델에 대해 각 패밀리 별로 분류한 정확도는 Fig. 8.과 같다. 희소한 패밀리인 Class 5의

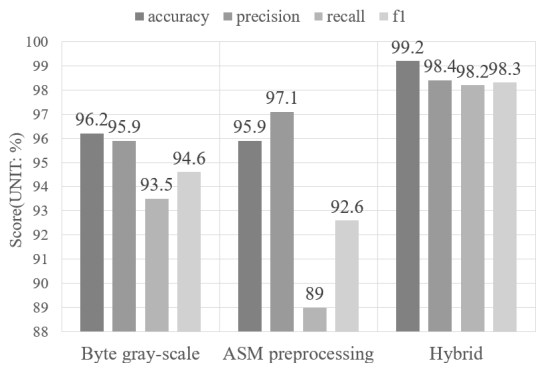

Fig. 6. Experimental group performance evaluation results

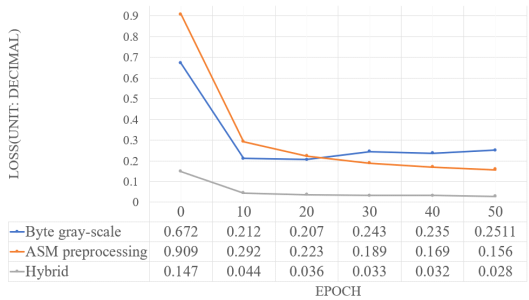

Fig. 7. Loss performance according to epoch

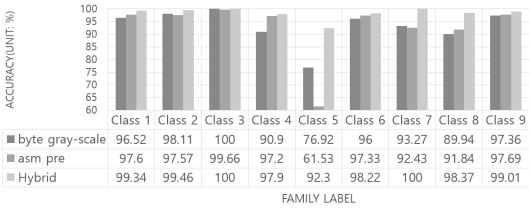

Fig. 8. Accuracy of Family

경우 BYTES Gray-scale 데이터셋보다 제안한 하이브리드 데이터셋이 약 15% 높은 정확도를 보였다. 종래 모델인 ASM 전처리된 데이터셋과 비교했을 때, 제안하는 하이브리드 데이터셋은 약 30% 개선된 정확도를 보였다.

난독화된 패밀리인 Class 8의 경우 BYTES Gray-scale 데이터셋의 성능 대비 제안한 하이브리드 데이터셋의 정확도가 약 8% 향상되었다. ASM 전처리된 데이터셋에 대해서는 제안한 하이브리드 데이터셋이 약 6% 정도 높은 정확도를 보였다.

하이브리드 데이터셋은 악성코드 파일을 단독으로 전처리했을 때보다 높은 성능을 보였다. 또한 희소한 패밀리인 Class 5에 대하여 약 30% 개선된 분류 성능을 보였으며, 난독화된 패밀리인 Class 8에 대해서는 분류 성능이 약 8% 개선되었다.

VI. 결 론

본 논문에서는 하이브리드 데이터셋을 생성하기 위한 전처리 방법을 제안하였으며, 악성코드의 ASM 파일과 BYTES 파일을 동시에 사용한 하이브리드 데이터셋을 분류하였다. 실험 결과 제안하는 하이브리드 데이터셋을 사용한 모델이 단일 데이터셋만 사용하는 방식보다 정확도와 손실에서 우수함을 보였다. 또한 하이브리드 데이터셋은 단일 데이터셋과 비교하여 희소한 패밀리에 대해서는 약 30% 우수한 성능을 보였고, 난독화된 패밀리에 대해서는 약 8% 우수한 성능을 나타낸다.

하지만 본 연구에서 제안하는 하이브리드 데이터셋은 두 개의 분류 모델을 사용하는 복잡한 구조이므로 모델 구축시에 시간과 비용이 더 많이 소모될 수 있다. 따라서 향후에는 이러한 복잡성을 줄이고 더 효율적인 모델을 개발하기 위해 CNN 모델에서 추출한 특징을 클러스터링 알고리즘의 유사도로 대체하여 새로운 하이브리드 기반 전처리 기법을 연구할 계획이다.

References

[1] Faitouri A. Aboaoja, Anazida Zainal, Fuad A. Ghaleb, Bander Ali Saleh Al-rimy, Taiseer Abdalla Elfadil Eisa, and Asma Abbas Hassan Elnour, "Malware Detection Is-sues, Challenges, and Future Directions: A Survey," Applied Sciences 12, No. 17, pp. 8482, Aug. 2022.

[2] Gaurav, Akshat, Brij B. Gupta, and Prabin Kumar Panigrahi, "A comprehensive survey on machine learning approaches for malware detection in IoT-based enterprise information system," Enterprise Information Systems Vol. 17, No. 3 pp. 2023764, Jan. 2023.

[3] Baoguo Yuan, Junfeng Wang, Peng Wu, and Xianguo Qing, "IoT Malware Classification Based on Lightweight Convolutional Neural Networks," IEEE Internet of Things Journal, Vol. 9, No. 5, pp. 3770-3783, March. 2022.

[4] Malware statistics and facts for 2023, https://www.comparitech.com/antivirus/malware-statistics-facts/, last accessed 2023/09/20.

[5] Cybercrime To Cost The World $10.5 Trillion Annually By 2025, https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/, last ac-cessed 2023/09/20.

[6] Ansam Khraisat, Iqbal Gondal, Peter Vamplew and Joarder Kamruzzaman, "Survey of intrusion detection systems: techniques, datasets and challenges," Cybersecurity, Vol. 2, July, 2019.

[7] Fahad Alswaina and Khaled Elleithy, "Android Malware Family Classification and Analysis: Current Status and Future Directions," Electronics 9, No. 6, pp. 942, June.

2020.
[8] Chae-rim Han, Su-hyun Yun, Myeong-jin Han, and Il-Gu Lee, “Machine Learning-based Malicious URL Detection Technique,” Journal of the Korea Institute of Information Security & Cryptology, 32(3), pp. 555-564, June. 2022.
[9] EunJi Lim, EunYoung Lee, and Il-Gu Lee, “Behavior and Script Similarity-based Cryptojacking Detection Framework Using Machine Learning,” Journal of the Korea Institute of Information Security & Cryptology, 31(6), pp. 1105-1114, Dec. 2021.
[10] Jiang, J., Zhang, Y. “A pyramid stripe pooling-based convolutional neural network for malware detection and classification,” J Ambient Intell Human Comput 14, pp. 2785–2796, Feb. 2023.
[11] Y.C. Qiao, Q.S. Jiang, L. Gu, X.M. Wu. “Research on malicious code classification based on assembly instruction word vector and convolutional neural network,” Infor-mation network security, 04, pp. 20-28, Jan. 2019.
[12] Ahmed Bensaoud, Nawaf Abudawaood, and Jugal Kalita, “Classifying Malware Images with Convolutional Neural Network Models,” ArXiv, Oct. 2020.
[13] Hanqi Zhang, Xi Xiao, Francesco Mercaldo, Shiguang Ni, Fabio Martinelli and Arun Ku-mar Sangaiah, “Classification of ransomware families with machine learning based on N-gram of opcodes,” Fusture Generation Computer Systems, Vol 90, pp. 211-221, Jan. 2019.
[14] Rajasekhar Chagantia, Vinayakumar Ravib, and Tuan D. Pham, “Image-based Malware Representation Approach with EfficientNet ConvolutionalNeural Networks for Eff ective Malware Classification,” Journal of Information Security and Applications, Vol. 69, pp. 103306, Sep. 2022.
[15] Ketan Gupta, Nasmin Jiwani, Md Haris Uddin Sharif, Ripon Datta, and Neda Afreen, “A Neural Network Approach For Malware Classification,” 2022 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), pp. 681-684, Nov. 2022.
[16] Zhu X, Huang J and Wang B, Qi C. “Malware homology determination using visualized images and feature fusion,” PeerJ Computer Sci. Vol. 7, Apr. 2021.
[17] Parvin, H., Minaei, B., Karshenas H. and Beigi, A., “A New N-gram Feature Extraction-Selection Method for Malicious Code,” Lecture Notes in Computer Science, pp. 98-107, Apr. 2011.
[18] Aninditya, Annisa, Muhammad Azani Hasibuan, and Edi Sutoyo, “Text mining ap-proach using TF-IDF and naive Bayes for classification of exam questions based on cognitive level of bloom′s taxonomy,” 2019 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS), pp. 112-117, Nov. 2019.
[19] Akuma, Stephen, Tyosar Lubem, and Isaac Terngu Adom, ″Comparing Bag of Words and TF-IDF with different models for hate speech detection from live tweets,″ International Journal of Information Technology, Vol. 14, pp. 3629-3635, Sep, 2022.
[20] Liu, Yipeng, Junwu Wang, Shanrong

Tang, Jiaji Zhang, and Jinyingjun Wan, "Integrating Information Entropy and Latent Dirichlet Allocation Models for Analysis of Safety Accidents in the Construction Industry," Buildings 13, No. 7, pp. 1831, July. 2023.

[21] Y. Li and B. Shen, "Research on sentiment analysis of microblogging based on LSA and TF-IDF," 2017 3rd IEEE International Conference on Computer and Communications (ICCC), pp. 2584-2588, Dec. 2017.

[22] Adarsh Kumar Singh, Gandharv Wadhwa, Mayank Ahuja, Keshav Soni and Kapil Sharma, "Android Malware Detection using LSI-based Reduced Opcode Feature Vector," Procedia Computer Science, Vol. 173, pp. 291-298, July. 2020.

[23] Ronen, R., Radu, M., Feuerstein, C., Yom-Tov, E. and Ahmadi, M., "Microsoft malware classification challenge," ArXiv, Feb. 2018.

[24] Aman, N., Saleem, Y., Abbasi, F.H. and Shahzad, F. "A Hybrid Approach for Malware Family Classification," Communications in Computer and Information Science, Vol. 719, Springer, June. 2017.

〈저 자 소 개〉

최 서 우 (Seo-Woo Choi) 학생회원
2021년 3월~현재: 성신여자대학교 융합보안공학과 학사
2023년 3월~현재: 성신여자대학교 CSE LAB 연구원
〈관심분야〉 정보보호, 악성코드 분석, 침해사고대응

한 명 진 (Myeong-Jin Han) 학생회원
2021년 3월~현재: 성신여자대학교 융합보안공학과 학사
2023년 3월~현재: 성신여자대학교 CSE LAB 연구원
〈관심분야〉 정보보호, 모의해킹, 취약점 분석

이 연 지 (Yeon-Ji Lee) 학생회원
2022년 2월: 성신여자대학교 융합보안공학과 졸업
2022년 3월~현재: 성신여자대학교 미래융합기술공학과 석사과정
〈관심분야〉 융합보안, 이상행위탐지, 기계학습

이 일 구 (Il-Gu Lee) 종신회원
2003년 2월: 서강대학교 전자공학과 졸업
2005년 2월: KAIST 정보통신대학원 석사
2016년 2월: KAIST 전산학부 박사
2005년 2월~2017년 2월: 한국전자통신연구원 5G기가통신시스템연구본부 선임연구원
2017년 3월~현재: 성신여자대학교 미래융합기술공학과/융합보안공학과 부교수
〈관심분야〉 융합보안, 정보보호, 정보통신