JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 27 No. 11, pp. 123-130, November 2022
https://doi.org/10.9708/jksci.2022.27.11.123

Light-weight Classification Model for Android Malware through the Dimensional Reduction of API Call Sequence using PCA

Dong-Ha Jeon*, Soo-Jin Lee*

*Graduate Student, Dept. of Defense Science, Korea National Defense University, Nonsan, Korea
*Professor, Dept. of Defense Science, Korea National Defense University, Nonsan, Korea

[Abstract]

Recently, studies on the detection and classification of Android malware based on API Call sequence have been actively carried out. However, API Call sequence based malware classification has serious limitations such as excessive time and resource consumption in terms of malware analysis and learning model construction due to the vast amount of data and high-dimensional characteristic of features. In this study, we analyzed various classification models such as LightGBM, Random Forest, and k-Nearest Neighbors after significantly reducing the dimension of features using PCA(Principal Component Analysis) for CICAndMal2020 dataset containing vast API Call information. The experimental result shows that PCA significantly reduces the dimension of features while maintaining the characteristics of the original data and achieves efficient malware classification performance. Both binary classification and multi-class classification achieve higher levels of accuracy than previous studies, even if the data characteristics were reduced to less than 1% of the total size.

▸ **Key words**: API-Call, PCA, Dimensional Reduction, LGBM, RF, KNN

[요 약]

최근 API Call 정보를 기반으로 안드로이드 악성코드를 탐지 및 분류하는 연구가 활발하게 진행되고 있다. 그러나 API Call 기반의 악성코드 분류는 방대한 데이터 양과 높은 차원 특성으로 인해 악성코드 분석과 학습 모델 구축 과정에서 과도한 시간과 자원이 소모된다는 심각한 제한사항을 가진다. 이에 본 연구에서는 방대한 API Call 정보를 포함하고 있는 CICAndMal2020 데이터세트를 대상으로 PCA(Principal Component Analysis, 주성분분석)를 사용하여 차원을 대폭 축소시킨 후 LightGBM, Random Forest, k-Nearest Neighbors 등의 다양한 분류 기법 모델을 적용하여 결과를 분석하였다. 그 결과 PCA가 원본 데이터의 특성을 유지하면서 데이터 특성의 차원은 획기적으로 감소시키고 우수한 악성코드 분류 성능을 달성함을 확인하였다. 이진분류 및 다중분류 모두 데이터 특성을 전체 크기의 1% 수준 이하로 줄이더라도 이전 연구 결과보다 높은 수준의 정확도를 나타내었다.

▸ **주제어**: API-Call, 주성분분석, 차원축소, LGBM, RF, KNN

• First Author: Dong-Ha Jeon, Corresponding Author: Soo-Jin Lee
*Dong-Ha Jeon (acenoma@naver.com), Dept. of Defense Science, Korea National Defense University
*Soo-Jin Lee (cyberkma@gmail.com), Dept. of Defense Science, Korea National Defense University
• Received: 2022. 10. 12, Revised: 2022. 11. 11, Accepted: 2022. 11. 11.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

2008년 첫 번째 상용버전이 출시된 안드로이드 운영체제는 2009년 전 세계 운영체제 시장에서 4%의 점유율을 기록하기는 했지만, 이후 3년 동안 매년 약 20%씩 시장 점유율을 확대시켜 왔다[1]. Statcounter의 조사 결과에 의하면, 2022년 7월 기준으로 안드로이드의 모바일 운영체제의 시장 점유율은 71.85%, 한국에서의 시장 점유율은 70.15%로 나타났다[2].

안드로이드 운영체제가 이처럼 운영체제 시장에서 성공할 수 있었던 주 요인은 오픈 소스 소프트웨어(open source software)라는 점이다. 오픈 소스는 구글이 개발하였지만, 스마트폰 제조사들은 자사의 기기에 어떤 앱을 설치할지 선택할 수 있으며, 자사만의 레이아웃과 인터페이스를 적용해 사용자들이 유일한 경험을 누릴 수 있도록 해 준다. 한편 오픈 소스 소프트웨어라는 특성으로 인해 다양한 보안위협에 노출되고 있기도 하다.

독일 보안업체 짐페리움의 발표자료[3]에 따르면, 2021년 9월 전 세계 70개의 국가 및 지역에서 1천만대 이상의 안드로이드 스마트폰을 공격해 수백만 유로를 탈취한 악성코드가 발견됐다. 이 악성코드는 2020년 말부터 200여 개 이상의 앱을 감염시킨 후 구글플레이와 서드파티 앱스토어를 통해 퍼진 것으로 확인되었다. 이처럼 안드로이드 운영체제는 특유의 개방성으로 모바일 악성코드 유포에 최적화된 수단이 되고 있어 안드로이드 운영체제를 대상으로 한 공격은 그 수와 방법이 더욱 다양해지고 있다.

안드로이드 운영체제는 사전 정의된 API(Application Programming Interface) 호출을 통해 컴퓨터 응용프로그램을 동작시킨다. 안드로이드 기반 악성코드 역시 API 호출을 통하여 코드 실행이 결정되기 때문에 API Call 분석은 악성코드 탐지에 중요한 역할을 할 수 있다. 그러나 일반적으로 API Call 정보를 포함하고 있는 악성코드는 고차원의 특성(feature)을 포함하고 있고 데이터의 용량도 크기 때문에 분석에 많은 시간을 투자해야 한다. 그리고 데이터 값이 대부분 '0' 또는 동일 값으로 이루어져 있어 정확한 분석이 제한될 수 있다는 점을 유의해야 한다.

본 연구에서 사용되는 CICAndMal2020 데이터세트의 경우에도 총 특성의 수가 9,503개이고, 데이터의 값 또한 대부분 '0' 또는 '1'을 가지고 있다. 이 중 '0'의 값을 갖는 9,400개 이상의 특성이 일반적인 악성코드뿐만 아니라 해당 악성코드와 관련된 API Call과도 관련이 없다. 이러한 데이터세트 특징으로 인해 일반적인 컴퓨팅 환경에서의 악성코드 분석이나 분류 및 탐지모델 구축은 쉽지 않으며, 분류 및 탐지 정확도 역시 영향을 받는 등 시간과 자원 측면에서 심각한 한계에 직면할 수밖에 없다.

이에 본 논문에서는 주성분분석(Principal Component Analysis, 이하 PCA)[4]을 적용하여 새로운 주성분(PC) 추출을 통해 API Call 정보에 대한 차원을 대폭 축소시킨 후 다양한 분류 알고리즘을 적용하여 효율적으로 안드로이드 악성코드를 탐지 및 분류하는 방안을 제시한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 PCA를 활용한 악성코드 탐지 및 API Call 정보를 기반으로 악성코드 분류를 시도한 선행연구들을 정리하고, Ⅲ장에서는 데이터세트, PCA 및 분류 알고리즘 적용 방안에 대해 설명한다. Ⅳ장에서는 데이터세트 전처리 과정을 기술하고, 다양한 분류 알고리즘을 적용하여 수행한 실험 결과를 분석한다. 끝으로 Ⅴ장에서 연구결과를 정리하고 결론을 맺는다.

II. Previous Works

PCA는 대량의 특성을 포함하고 있는 고차원/대규모의 데이터세트를 분석하는 과정에서 원본 데이터의 특성을 최대한 보존하면서 해석 가능성을 높이고, 다차원의 데이터를 시각화하는데 널리 사용되는 기술이다. 최근에는 이러한 PCA를 악성코드 탐지 및 분류에 적용하려는 시도가 활발하게 진행되고 있다. L. Shilpa 등[5]은 PCA를 사용하여 네트워크 데이터세트를 정상파일과 악성파일의 공간으로 구분하는 방안을 제시하였고, Y. Liu 등[6]은 [5]에서 제시된 접근 알고리즘을 개선하여 악성코드 탐지 속도 측면에서 향상된 결과를 달성하였다. 그러나 두 연구는 각각 데이터 차원의 크기를 축소하기 위해 주성분을 적절하게 선택하는 방법과 탐지 성능을 향상시키기 위한 거리 측정 방식에 대해서는 구체적으로 설명하지 못하고 있다.

선행연구의 문제점을 보완하기 위하여 H. Kye 등[7]은 다변량 통계 네트워크 모니터링 방식 등의 방법을 통하여 적절한 주성분의 개수를 선택하는 방안을 제시했다. 또한, PCA 적용 과정에서 공분산 행렬 기반의 마할라노비스 거리(mahalanobis distance) 계산 방식을 적용하여 PCA의 이상탐지 성능을 향상시켰다.

S. Waskle 등[8]은 NSL KDD 데이터세트를 대상으로 PCA를 사용하여 데이터 차원을 감소시키고 RF 분류 알고리즘을 적용하여 수행시간을 단축하였고, 약 97% 수준의 정확도를 달성하여 SVM, Naïve Bayes, Decision Tree 분류 기법에 비해 향상된 탐지 능력을 확인하였다.

B. Dissanayake 등[9]은 MLP(Multi Layer Perception)

를 이용해 RF 기반 특성 중요도(feature importance) 산출을 통해 30% 수준의 특성 선택으로 90%의 정확도를 달성하였다. 이후 RF 분류 알고리즘을 이용한 실험 결과 동일한 수준의 정확도를 지속적으로 유지하였다.

본 연구의 데이터세트인 CICAndMal2020을 대상으로 특성 선택 및 추출 기법을 사용하여 악성코드 탐지 및 분류 성능을 향상시키는 연구도 다양하게 진행되었다.

A. Rahali 등[10]은 CICAndMal2020 데이터세트를 직접 생성한 후 Extra-Tree Classifier를 사용하여 특성 개수를 줄이고 CNN을 적용하여 악성코드 탐지 성능을 분석하였다. 실험 결과 다중분류 정확도는 82% 수준을 달성하였고 이진분류는 실시하지 않았다. 그리고 데이터의 특성 차원을 대폭 감소시켰다고 주장하였으나 구체적인 방법을 제시하지 않았으며, 무엇보다도 제시된 방법론을 구현하기 위해서는 고성능의 컴퓨팅 환경(CPU 50개, 512GB 메모리)이 필요하기 때문에 일반적 환경에서 적용이 어렵다는 한계가 있다. N. Peiravian 등[11]은 클래스 파일 기반 API Call 특성과 안드로이드 Manifest 권한 특성을 결합시킨 후 Bagging Classifier를 사용하여 이진분류를 시도하였고 약 96%의 정확도를 달성하였다.

동적 분석 연구에서는 A. D. Lorenzo 등[12]이 안드로이드 애플리케이션 기반 악성 프로그램 대상 악성코드 탐지 분석 결과를 VizMal이라는 도구를 사용하여 시각화하였다. D. S. Keyes 등[13]은 가상환경에서의 동적 분석을 통해 140여개의 특성을 추출하고, Decision Tree 분류 기법으로 이진분류를 시도한 결과 약 98%의 정밀도와 0.983의 Recall 값을 달성하였다. 또한, H. Hwang 등[14]은 다양한 특성 선택 방법을 적용하여 데이터세트의 차원을 축소하여 핵심 특성 집합을 추출하는 방안을 제시하였고 CNN을 이용하여 악성코드 분류를 시도한 결과 이진분류의 경우 97% 수준의 정확도를, 다중분류의 경우 83% 수준의 정확도를 달성하였다.

III. The Proposed Scheme

본 논문에서 제안하는 접근방법의 전체 흐름은 Fig. 1에서 보는 바와 같다. 먼저 API Call 정보를 포함하고 있는 CICAndMal2020 데이터세트와 정상파일인 Androzoo를 대상으로 PCA를 통하여 데이터세트 특성 차원의 크기를 대폭 축소하면서 주성분을 추출한다. 다음으로 추출된 주성분 특성을 기반으로 다양한 분류 기법을 적용하여 악성코드를 이진분류 및 다중분류를 실시한다.

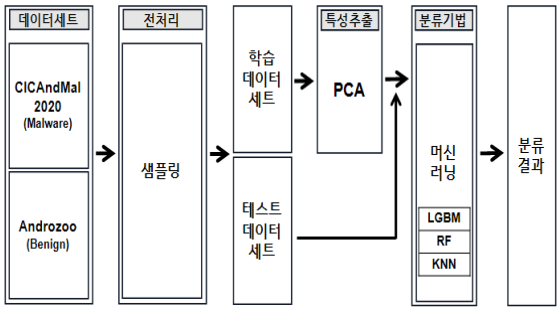

Fig. 1. Overview of Proposed Approach

1. Dataset

Canadian Institute for Cybersecurity에서 발표한 CICAndMal2020은 대표적인 안드로이드 악성코드 데이터세트로 총 195,623개의 악성코드로 이루어져 있다. 악성코드(Malware)는 총 9,503개의 특성 정보를 가지고 있고 14개의 카테고리로 분류되어 있다. 해당 악성코드는 다른 데이터세트에 비해 대용량, 고차원 데이터의 성격을 가지고 있고 API Call에 해당하는 대부분의 정보는 '0'의 값을 나타낸다.

Table 1에서 보는 바와 같이 실험에 사용된 카테고리는 총 12개(179,988개)로 No_Category 클래스와 Zero_day 클래스에 해당하는 악성코드는 제외하였다.

Table 1. Number of Malware in CICAndMal2020 Dataset

Class	the Number of samples
Adware	47,210
Backdoor	1,538
Fileinfector	669
(No_Category)	(2,295)
PUA	2,051
Ransomware	6,202
Riskware	97,349
Scareware	1,556
Trojan	13,559
Trojan_Banker	887
Trojan_Dropper	2,302
Trojan_SMS	3,125
Trojan_Spy	3,540
(Zero_day)	(13,340)

No_Category의 경우 데이터세트를 제작하는 과정에서 다른 13개의 클래스처럼 특성 정보를 명확하게 분류하는 것이 곤란한 악성코드들만을 별도로 모아둔 클래스이기 때문에 특성 학습을 통한 분류 자체가 불가능하다. 이러한 이유로 데이터세트를 제작했던 연구자들도 이 클래스는 실험에서 배제하였다. Zero_day는 클래스를 구성하는 하위 패밀리가 매우 다양하기 때문에 해당 클래스의 고유한 특성을 파악하는 것이 제한되어 학습 이후 모델의 정확한

성능 검증을 방해한다. 이에 본 연구에서는 No_Category와 Zero_day 두 클래스를 제외하고 실험을 진행하였다.

정상파일(Benign)은 CICAndMal2020 악성코드(Malware) 데이터 수를 고려하여 Androzoo(http://androzoo.uni.lu) 데이터 162,901개를 사용하였다. Table 2에서 보는 바와 같이 정상파일은 총 5개의 하위 데이터세트로 분류되어 있지만, 각 데이터세트별로 구분되는 특성을 가지지 않으며 단지 데이터 수집 시기에서만 차이가 있다.

Table 2. Number of Benign Dataset

Class	the Number of samples
Benign0	32,804
Benign1	47,861
Benign2	42,635
Benign3	7,847
Benign4	31,754

2. PCA

CICAndMal2020 데이터세트는 총 9,503개의 특성으로 이루어져 있기 때문에 방대한 양과 고차원의 데이터로 볼 수 있고, 악성코드 분석 및 학습을 위해서는 많은 계산 비용과 처리시간을 감당할 수 있도록 [10]에서 제시된 바와 같이 다수의 CPU와 대용량 메모리를 갖춘 고성능 컴퓨팅 환경이 필요하다.

이러한 문제점을 해결하고 저사양의 컴퓨팅 환경에서도 API Call 정보를 기반으로 한 안드로이드 악성코드 탐지 및 분류가 가능하도록 경량화하기 위해 본 연구에서는 고차원 데이터를 효과적으로 분석해주는 PCA를 활용하였다. 데이터 시각화, 군집화, 압축 등에 광범위하게 활용되는 PCA는 고차원의 데이터세트를 특성 왜곡을 최소화하면서 축소하는 효과적인 기법으로 주어진 데이터세트의 무수히 많은 속성에서 전체 데이터의 분산을 가장 잘 설명해주는 주성분이라는 새로운 속성을 적절한 수만큼 뽑아낸다.

PCA 동작 원리는 Fig. 2에서 보는 것과 같이 데이터의 차원을 낮추기 위해 원래의 데이터 구조를 가장 잘 유지할 수 있는 벡터를 구하고, 그 벡터에 데이터들을 정사영(projection)하는 방식으로 진행된다. 즉 원래 데이터의 분산을 최대한 보존하는 새로운 축을 찾고, 그 축에 데이터를 정사영한다. 따라서 전체 특성 중 일부 특성만 선택하여 사용하는 특성 선택(feature selection)과는 특성의 차원을 축소한다는 점은 유사하지만 새로운 특성을 찾는다는 점에서 차이가 있다. 세부 동작 방식은 아래와 같다.

(1) 모든 차원(d)의 데이터세트 취합 및 각 차원(d)의 평균 벡터(mean vector) 계산

(2) 데이터세트 공변 행렬(covarian matrix) 계산

(3) 고유벡터(eigenvectors)와 고유값(eigenvalues) 계산

(4) 고유값(eigenvalues) 정렬 후 높은 순으로 n개의 고유 벡터(eigenvectors) 선택

(5) M 행렬(M=d*m) 산출

(6) M 행렬로 새로운 표본 공간 형성(=주성분)

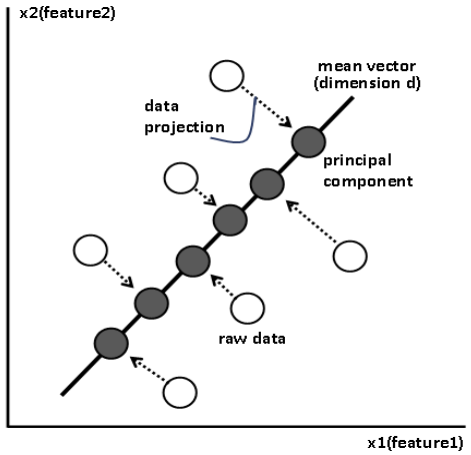

Fig. 2. Data Dimension Reduction Process Using PCA

본 연구에서는 PCA를 활용하여 원래 데이터의 특성 총 9, 503개에서 주성분(PC)의 숫자를 늘려나가며 분산도(새롭게 추출된 주성분이 기존 데이터의 전체 특성을 설명할 수 있는 정도)를 측정하였고, 새로운 주성분을 추출하여 다양한 악성코드 분류 기법 적용을 통해 탐지 성능을 확인하는 방식으로 연구를 진행하였다.

3. Classification Model

초도 실험에서는 XGBoost, LightGBM, KNN, SVM 및 Random Forest 총 5가지의 분류모델을 적용하여 실험을 진행하였으나, XGBoost는 LightGBM에 비해 탐지 성능이 많이 저하되었으며 SVM 또한 만족할만큼의 탐지 성능이 관찰되지 않았다. 이러한 이유로 본 실험은 LightGBM, Random Forest 및 KNN 3가지의 분류모델만 적용하여 진행하고 결과를 분석하였다.

트리를 수평으로 확장하는 방식의 학습 알고리즘인 LightGBM은 Gradient Boosting 방식의 프레임워크로서, 적은 메모리를 사용하면서 높은 정확도를 보장하고 속도 역시 빠르기 때문에 데이터세트의 크기가 커질수록 효율적인 성능을 보인다. Random Forest는 결정트리 기반의

학습 알고리즘으로서, Bagging 방식으로 직관적인 계산을 이용하며 비교적 빠른 속도를 보장하기 때문에 다양한 컴퓨팅 분야에 활용되고 있다. KNN은 거리 기반 학습 알고리즘으로서, 특성 정보와 클래스 간 관계를 이해하는 것은 제안되지만, 구현이 단순하여 수행 속도가 빠르다.

IV. Experimental Results

1. Environment

본 실험은 Window 10 64bit 운영체제, Intel(R) Core(TM) i5-8250U CPU, RAM 8GB의 컴퓨팅 환경에서 진행하였다. 개발언어는 Python 3.7.10 버전, 알고리즘은 Scikit Learn(Python) 라이브러리를 사용하였다.

2. Experimental Dataset

Malware에 해당하는 CICAndMal2020 데이터세트와 Benign에 해당하는 Androzoo 데이터세트를 사용하여 악성코드 여부를 분류하는 이진분류와 악성코드 카테고리를 분류하는 다중분류로 구분하여 실험을 진행하였다. 앞서 언급한 실험 환경에서 효율적으로 해당 고차원 데이터세트를 학습하고 성능을 검증하기 위해 각 데이터세트에서 일부 데이터를 추출하여 서브데이터세트를 구성하였다.

먼저 이진분류를 수행하기 위해서 Benign에서 1개의 클래스당 1,400개씩 총 5개 클래스(총 7,000개)로 서브데이터를 추출하였다. Malware 또한 Benign 서브데이터의 수를 고려하여 각 클래스당 600개씩 총 12개 클래스에서 서브데이터를 추출하였다. 이 때, 기존 14개 클래스에서 No_Category 및 Zero_day 클래스는 제외하였고, 가장 적은 수의 악성코드를 가지고 있는 FileInfactor(669개) 클래스를 기준으로 추출할 서브데이터의 수를 판단하였다.

서브데이터 추출 과정은 10번 반복 수행하여 총 10회의 실험을 실시하고 평균값을 산출하여 결과를 분석하였다. 즉 10회 반복 실험을 진행하면서 사용된 서브데이터세트는 모두 상이하며, 원본 데이터 전체는 아니지만 최대한 많은 수의 데이터가 실험에 사용될 수 있도록 노력하였다. 위 과정을 통해 구성된 서브데이터세트는 다시 학습 서브데이터세트와 검증 서브데이터세트로 8:2의 비율로 구분하여 실험에 적용하였다.

학습 및 성능 검증 서브데이터세트 각각에서 추출한 주성분은 성능 검증 과정에 이용하지 않고, 학습 서브 데이터세트에서 추출한 주성분만을 성능 검증 서브 데이터세트에 적용하여 실험을 진행하였다. 이는 주성분을 추출하는 데이터세트와 주성분을 적용하는 데이터세트를 달리함으로써 객관적인 주성분 추출 과정을 만들기 위함이다.

3. Results

Benign과 Malware 학습 서브데이터에서 주성분을 10개부터 증가시키면서 추출하고 3가지의 분류모델에 학습시킨 후 성능 검증 데이터세트를 대상으로 성능 검증을 실시하였다. 실험 결과, 주성분이 100개 이하일 때 가장 높은 정확도의 분류 성능을 나타냈으며 이진분류 및 다중분류 결과는 Table 3 및 Table 4에서 확인할 수 있다. (-) 표시는 차원 축소 없이 9,503개의 특성 모두를 사용했을 때의 분류 성능을 의미한다.

Table 3. Results of Binary Classification

구분	# of PC	Accuracy	Precision	Recall	F1-score
Light GBM	-	0.9820	0.9821	0.9819	0.9820
	10	0.9566	0.9576	0.9560	0.9565
	50	0.9640	0.9653	0.9632	0.9639
	100	0.9680	0.9693	0.9673	0.9679
RF	-	0.9832	0.9839	0.9831	0.9834
	10	0.9544	0.9603	0.9602	0.9602
	50	0.9610	0.9632	0.9600	0.9609
	100	0.9603	0.9623	0.9593	0.9601
KNN	-	0.9625	0.9629	0.9621	0.9624
	10	0.9441	0.9443	0.9438	0.9440
	50	0.9618	0.9617	0.9617	0.9617
	100	0.9621	0.9621	0.9620	0.9621

Table 4. Results of Multi-class Classification

구분	# of PC	Accuracy	Precision	Recall	F1-score
Light GBM	-	0.8785	0.8832	0.8769	0.8783
	10	0.8396	0.8481	0.8368	0.8400
	50	0.8556	0.8654	0.8555	0.8562
	70	0.8597	0.8655	0.8576	0.8598
RF	-	0.8778	0.8868	0.8758	0.8788
	10	0.8465	0.8564	0.8446	0.8480
	50	0.8667	0.8775	0.8652	0.8686
	70	0.8681	0.8772	0.8662	0.8694
KNN	-	0.8306	0.8354	0.8308	0.8306
	10	0.8014	0.8081	0.7993	0.8015
	50	0.8296	0.8323	0.8287	0.8301
	70	0.8063	0.8156	0.8065	0.8073

이진분류의 경우 대부분의 분류 모델이 PC 개수를 늘릴수록 평가 결과가 높은 값을 나타냈으며, LightGBM이 PC 100개일 때 96.8%로 가장 높은 수준의 정확도를 달성하였다. 해당 결과에 대한 이진분류 오차행렬(Confusion Matrix)은 Fig. 3과 같다. 실험 결과를 통해 주성분 추출 과정에서 기존 9,503개의 특성 중에서 약 1% 수준의 특성 개수만으로 이전 연구들과 비슷한 수준의 정확도를 나타냄을 확인하였다.

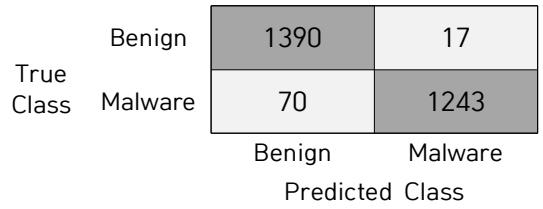

Fig. 3. Confusion Matrix of Binary Classification

다중분류의 경우 LightGBM과 Random Forest의 예측 결과값이 상대적으로 높게 나왔으며, Random Forest를 적용하였을 때 PC 70개에서의 정확도가 86.8%로 가장 높게 나타났다. 해당 결과에 대한 다중분류 오차행렬은 Fig. 4와 같다. 다중분류에서도 주성분 추출을 통해 9,503개의 특성 차원이 약 0.7% 수준으로 대폭 감소되었고, 정확도 측면에서도 약 87%로 Table 5에서 보는 바와 같이 CICAndMal2020 데이터세트를 대상으로 분류를 시도한 이전 연구들에 비해 매우 향상된 결과를 달성하였다.

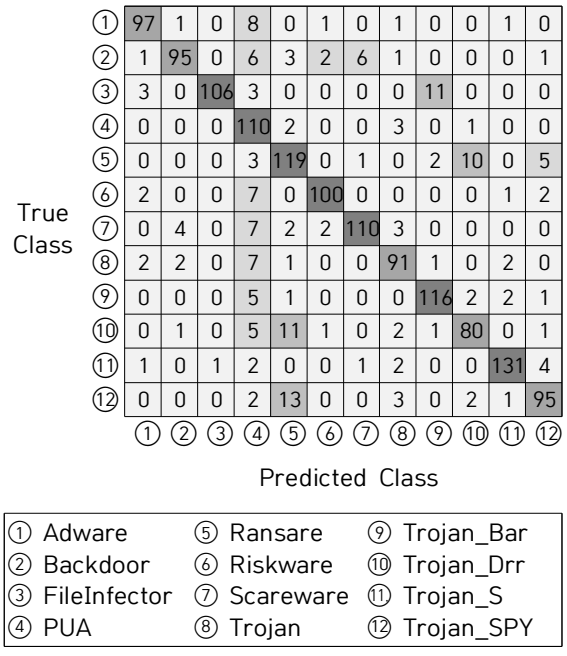

Fig. 4. Confusion Matrix of Multi-class Classification

Table 5. Comparision with Previous Studies

	Main Approach	# of Used Features	ACC
Proposed	PCA + Random Forest	70	86.81%
[10]	Feature Selection by Extra-Tree Classifier	2,237	82.22%
[14]	Feature Selection by Mutual Classifier	593	83.40%
	Feature Selection by Chi-Square Classifier	781	83.07%

이상과 같은 실험 결과를 바탕으로 PCA가 기존의 다양한 특성 추출 및 선택 방식에 비해 원본 데이터의 특성을 유지하면서 데이터 특성 차원을 대폭 감소시킬 수 있음은 물론, 우수한 탐지 성능도 보장함을 알 수 있다. 또한, PCA를 사용하면 고성능 컴퓨팅 환경이 아닌 일반적인 환경에서도 고차원의 데이터를 대상으로 효율적인 탐지모델 구축이 가능하다는 점도 확인하였다.

V. Conclusion

본 논문에서는 CICAndMal2020 데이터세트를 대상으로 안드로이드 악성코드 탐지 및 분류를 시도하였다.

API Call 정보를 포함하고 있는 데이터세트는 단위 데이터가 각각 수천 개에 달하는 특성을 가지고 있어 특성의 차원이 매우 고차원이며 데이터 양 또한 방대하다. 따라서 일반적인 컴퓨팅 환경에서는 분석이나 학습모델 구축이 심각하게 제한되어 과도한 시간과 자원 소모를 방지하고 효율적인 성능 도출을 보장하기 위해서는 악성코드 분류 이전에 차원 축소가 반드시 선행되어야 한다. 이러한 문제점을 해결하기 위해 본 논문에서는 기존에 널리 활용되던 특성 선택 기법이 아닌 PCA를 사용하여 차원을 대폭 축소시킨 후 LightGBM, Random Forest 및 KNN 등의 3가지 머신러닝 기법을 적용하여 분류를 시도하였다.

제안한 방안을 검증하기 위하여 CICAndMal2020 데이터세트의 특성 총 9,503개에 대해 PCA를 적용하여 새로운 주성분을 추출하였다. 추출된 주성분을 개수 별로 구분하여 3가지 분류모델에 학습시킨 후 악성코드 분류 성능을 측정한 결과, 데이터 특성 차원은 획기적으로 감소시키면서도 기존 연구들에 비해 높은 분류 성능을 달성하였다.

이진분류에서는 LightGBM이 원본 데이터세트 특성을 약 1% 수준의 크기로 줄이면서 가장 높은 96.8%의 정확도를 달성하였다. 다중분류의 경우에는 Random Forest 기준 약 0.7% 수준으로 특성 차원을 줄이면서 약 87%의 정확도를 달성하였다. 이러한 결과는 동일 데이터세트를 사용하여 가장 높은 정확도를 달성했던 이전 연구[14]보다 더 향상된 결과이다. 따라서 PCA는 기존의 다양한 특성 선택 방식에 비해 원본 데이터의 특성을 유지하면서 데이터의 특성 차원을 대폭 감소시켜 저사양의 컴퓨팅 환경에서도 동작 가능한 경량화된 분류모델을 구축할 수 있게 해주고 분류 성능 향상에도 결정적으로 기여하는 효과적인 접근방법이라고 할 수 있다.

본 논문에서 제안하는 이러한 접근방법은 API Call을 통해 응용프로그램을 동작시키는 안드로이드 운영체제를 대상으로 하여 경량화된 악성코드 탐지 및 분류모델 구축 가능성을 확인하고, 분류모델의 성능 또한 크게 향상시켰다는 점에서 향후 안드로이드 악성코드 대응방안 고도화에 많은 도움이 될 것이라 확신한다.

향후에는 PCA를 기반으로 대용량 및 고차원 데이터세트의 차원은 줄이면서도 원본 데이터세트의 전체 특성을 사용한 경우보다 탐지 및 분류 성능을 향상시킬 수 있는 방안을 집중 연구할 예정이다. 또한 PCA를 통해 추출된 새로운 PC 특성의 정보를 분석하여 PC와 실제 악성코드 동작에 미치는 특성과의 관계를 보다 세부적으로 분석하고, 제안하는 방안을 API Call 정보를 포함하고 있는 다른 형태의 데이터세트에도 적용하여 탐지 성능을 비교하면서 연구범위를 확장시켜 나가고자 한다.

REFERENCES

[1] Statista Research Department, Global market share smartphone operating systems of unit shipments 2014-2023, https://www.statista.com/statistics/272307/market-share-forecast-for-smartphone-operating-systems/

[2] Statcounter, Mobile Operating System Market Share Worldwide, https://gs.statcounter.com/os-market-share/mobile/south-korea/#monthly-202108-202208

[3] Zimperium, Financially Motivated Mobile Scamware Exceeds 100M Installations, https://blog.zimperium.com/dark-herring-android-.scamware-exceeds-100m-installations

[4] H. Abdi and L. J. Williams, Principal component analysis, Wiley interdisciplinary reviews: computational statistics 2 (4), 433-459, 2010.

[5] L. Shilpa, J. Sini, and V. Bhupendra, "Feature Reduction using Principal Component Analysis for Anomaly-Based Intrusion Detection on NSL-KDD", International Journal of Engineering Science and Technology, Vol. 2, No. 6, pp.1790-1799, July. 2010, DOI: 10.1.1.168.1957

[6] Y. Liu, L. Zhang, and Y. Guan, "Sketch-based streaming PCA algorithm for network-wide traffic anomaly detection ", 2010 IEEE 30th International Conference on Distributed Computing Systems, pp.807-816, Jun. 2010, DOI: 10.1109/ ICDCS.2010245

[7] Hyoseon Kyew and Minhae Kwon, "PCA-Based Low-Complexity Anomaly", KCIS, Vol. 46, No. 6, pp.941-955, June. 2021, DOI: 10.7840/kics.2021.46.6.941

[8] W. Subhash, L. Parashar, and U. Singh. "Intrusion detection system using PCA with random forest approach", 2020 International Conference on Electronics and Sustainable Communication Systems (ICESC), IEEE, pp.803-808, Aug. 2020, DOI: 10.1109/ICESC48915.2020.9155656

[9] Dissanayake, Maheshi B. "Feature Engineering for Cyber-attack detection in Internet of Things.", I.J Wireless and Microwave Technologies, Vol. 6, pp.46-54, Dec. 2021, DOI: 10.5815/ijwmt.2021.06.05.

[10] A. Rahali, A. H. Lashkari, G. Kaur, L. Taheri, F. Gagnon, and F. Massicotte, "DIDroid: Android Malware Classification and Characterization Using Deep Image Learning", Proc. of the 10th International Conference on Communication and Network Security (ICCNS2020), pp.70-82, Nov. 2020, DOI: 10.1145/3442520.3442522

[11] N. Peiravian and X. Zhu, "Machine Learning for Android Malware Detection Using Permission and API Calls", Proc. of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence, pp.300-305, Feb. 2014, DOI: 10.1109/ICTAI.2013.53

[12] A. D. Lorenzo, F. Martinelli, E. Medvet, F. Mercaldo and A. Santone, "Visualizing the outcome of dynamic analysis of Android malware with VizMal", Journal of Information Security and Applications, Vol. 50, Feb. 2020, DOI: 10.1016/j.jisa.2019.102423

[13] D. S. Keyes, B. Li, G. Kaur, A. H. Lashkari, F. Gagnon and F. Massicotte, "EntropLyzer: Android Malware Classification and Characterization Using Entropy Analysis of Dynamic Characteristics", Proc. of the 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), pp.1-8, May. 2021, DOI: 10.1109/RDAAPS48126.2021.9452002

[14] Hee-Jin Hwang and Soojin Lee, "Dimensionality Reduction of Feature Set for API Call based Android Malware Classification", Journal of The Korea Society of Computer and Information, Vol. 26, No. 11, pp.41-49, Nov. 2010, DOI: 10.9708/jksci.2021.26.11.041

Authors

Dong-Ha Jeon received B.S. degree in 2013 from the Department of International Relations, Social Humanities Republic of Korea Naval Academy. He is currently a graduate student in the Department of Defense Science, Korea National Defense University. His research interests include Machine Learning and Intrusion Detection System.

Soo-Jin Lee received B.S., M.S. and Ph.D. degrees in Computer Science from Korea Military Academy, Yonsei University and Korea Advanced Institute of Science and Technology(KAIST) in 1992, 1996 and 2006. He is currently a professor of the Department of Defense Science, Korea National Defense University from 2006. His research interests include National Cybersecurity Policy, Intrusion Detection System, Mobile Network Security, Machine Learning, Encryption theory and applications.