JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 30 No. 11, pp. 179-189, November 2025
https://doi.org/10.9708/jksci.2025.30.11.179

Transformer-based Android Malware Classification using Multi-stage Feature Selection

Gwang-Ho Kim*, 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]

High-dimensional features in API Call-based Android malware detection lead to high computational costs when applying Transformer models. To address this, this paper proposes a multi-stage feature selection pipeline combining LightGBM and Transformer's tokenization method to achieve both model lightweightness and high performance. The proposed method ranks feature importance using LightGBM and then dynamically constructs a final feature set constrained by each Transformer's maximum input token limit. Experimental results show that despite dramatically reducing 9,503 original features to 80-95, our model achieved up to 98.28% accuracy in binary classification and an 83.66% Macro F1-Score in multi-class classification. This demonstrates that our methodology provides comparable performance to previous studies with significantly fewer features, proving it to be an effective solution for ensuring both efficiency and high detection rates in high-dimensional data analysis.

▸ **Key words**: Android Malware, API-Call, Feature Selection, LightGBM, Transformer

[요 약]

API Call 기반 안드로이드 악성코드 탐지에서 고차원 특성은 트랜스포머 모델 적용에 적용 시 높은 계산 비용을 유발한다. 본 연구는 이 문제를 해결하기 위해 LightGBM과 트랜스포머 모델의 토큰화 방식을 결합한 다단계 특성 선택 파이프라인을 제안하여 모델 경량화와 성능을 동시에 달성하고자 한다. 제안하는 방법은 LightGBM으로 특성 중요도를 산출한 뒤, 각 트랜스포머 모델의 최대 입력 토큰 제약에 맞춰 최종 특성집합을 동적으로 구성한다. 실험 결과, 9,503개의 원본 특성을 80~95개로 획기적으로 축소하였음에도 불구하고 이진분류에서 최대 98.28%의 정확도를, 다중분류에서 83.66%의 Macro F1-Score를 달성하였다. 이는 제안 방법론이 훨씬 적은 특성으로 기존 연구들과 대등한 성능을 보이며, 고차원 데이터 분석에서 효율성과 높은 탐지 성능을 모두 확보할 수 있는 효과적인 해결책임을 입증한다.

▸ **주제어**: 안드로이드 악성코드, API-Call, 특성 선택, LightGBM, 트랜스포머

• First Author: Gwang-Ho Kim, Corresponding Author: Soo-Jin Lee
*Gwang-Ho Kim (champion2409@naver.com), Dept. of Defense Science, Korea National Defense University
**Soo-Jin Lee (cyberma@gmail.com), Dept. of Defense Science, Korea National Defense University
• Received: 2025. 10. 02, Revised: 2025. 11. 03, Accepted: 2025. 11. 17.

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

I. Introduction

안드로이드 운영체제는 2008년 첫 출시된 이후 세계 스마트폰 시장의 주요 동력으로 자리 잡았다. 2009년 당시 약 4%에 불과하던 세계 시장 점유율은 이후 3년간 연평균 20% 이상 증가하였다. 그리고 Statcounter 조사 결과에 따르면, 2025년 8월 기준 안드로이드의 전 세계 모바일 운영체제 점유율은 약 74%, 국내는 약 78%로 나타났다[1].

이러한 안드로이드 운영체제의 급성장은 구글이 개발한 오픈소스 기반 구조 덕분에 가능했으며 기기 제조사들은 이를 통해 애플리케이션 구성, 사용자 인터페이스, 레이아웃 등을 자유롭게 커스터마이징하여 차별화된 사용 경험을 제공할 수 있었다. 안드로이드의 이러한 개방성과 확장성은 기기 제조사와 개발자들에게 혁신의 기회를 제공하고 있지만 그와 동시에 다양한 보안 취약점을 확대하는 양날의 검이 되고 있다. 즉, 제조사와 개발자가 참여하는 생태계에서는 개방성과 확장성이 기능 다양성을 향상하는 원동력이 되었지만, 악성코드 제작자에게는 손쉬운 진입 경로를 제공하였다.

최신 보고서들은 안드로이드 운영체제를 대상으로 한 보안 위협이 더욱 정교하고 광범위하게 진화하고 있음을 잘 보여준다. 2024년 발표된 McAfee 보고서[2]에 따르면 'Goldoson' 악성코드는 구글 플레이에 등록된 60개 이상의 합법적인 앱을 통해 확산되었고 이후 1억 건 이상의 다운로드를 기록하면서 대규모 개인정보 유출을 야기하였다. 또한, Zimperium이 2024년 발표한 보고서[3]에 의하면 'SpinOk' 스파이웨어를 포함한 SDK가 100개 이상의 앱에 숨겨져 4억 회 이상 다운로드되며 사용자의 데이터를 탈취하였다. 이러한 사례들은 안드로이드 운영체제의 개방형 생태계가 여전히 대규모 악성코드 확산의 주요 경로로 악용될 수 있음을 명확하게 보여준다.

안드로이드를 겨냥한 악성코드를 탐지하는 기법 중 API Call 정보를 이용하는 탐지는 운영체제 실행 과정에서 호출되는 시스템·라이브러리 함수 정보를 기반으로 악성코드를 식별하는 방법이다. API Call 정보는 악성코드의 행위 패턴을 명확하게 반영하고 있어 정확한 악성코드 탐지 및 분류를 가능하게 해 준다. 그러나 최근 적용이 확대되고 있는 인공지능 기반 모델들로 API Call 정보를 활용하여 악성코드 탐지를 시도하는 것은 쉽지 않다. API Call 정보를 포함하고 있는 데이터세트는 대부분 수천 개의 특성으로 구성되어 있어 데이터 분석 및 학습 과정에서 대규모 연산 자원과 시간이 소요된다. 또한, 특성값 대부분이 0 또는 동일한 값을 포함하고 있어 학습 효율이나 탐지 정확도가 저하될 가능성이 있다.

본 연구에서 사용하는 CCCS-CIC-AndMal-2020 데이터세트[4] 역시 9,503개의 고차원 특성을 포함하고 있으며, 특성 대부분이 악성코드 실행 시 호출되는 전체 API Call 정보이다. 그리고 수천 개의 API 중 악성코드가 실행 시 실제로 호출되는 API는 극히 일부에 불과해 특성값 대부분이 '0'으로 채워진 희소행렬 구조를 가진다. 따라서 특성 간 높은 상관관계로 인한 중복성 문제를 내포할 수 있으며, 모델의 학습 효율과 탐지 성능이 심각하게 저하될 수 있다.

이러한 문제를 해결하기 위해 본 연구는 모델의 예측 성능과 아키텍처 제약조건을 동시에 고려하는 다단계 특성 선택 파이프라인을 제안한다. 제안하는 파이프라인은 LightGBM 기반의 특성 중요도 필터링과 트랜스포머의 고유한 토큰화 메커니즘을 결합하는 새로운 접근방법으로서 원본 데이터세트의 방대한 특성 공간을 최종 분류 모델인 트랜스포머 모델에 최적화된 1% 미만의 핵심 특성집합으로 효율적으로 압축한다. 본 연구는 이렇게 생성된 특성집합을 이진분류와 다중분류에 적용하여 모델의 경량화 가능성과 탐지 성능의 안정성을 검증한다.

이후 논문의 구성은 다음과 같다. Ⅱ장에서 본 연구에 사용된 CCCS-CIC-AndMal-2020 데이터세트에 대해 살펴본 후 동일 데이터세트를 기반으로 악성코드 분류를 시도했던 선행연구를 정리한다. Ⅲ장에서는 제안하는 다단계 특성 선택 파이프라인과 실험 과정에 대해 설명한다. Ⅳ장에서는 실험 결과를 분석하며 Ⅴ장에서 결론과 향후 연구 방향을 제시한다.

II. Preliminaries

1. Dataset

CCCS-CIC-AndMal-2020 데이터세트는 캐나다 사이버보안센터(CCCS)와 사이버보안연구소(CIC)가 협력하여 구축한 대규모 안드로이드 악성코드 공개 데이터세트로서 안드로이드 악성코드 탐지 및 분류 연구를 위해 널리 활용되고 있다. 데이터세트에는 정상파일 162,901개, 악성코드 데이터 195,624개를 포함 총 358,525개의 데이터가 포함되어 있다. 악성코드 데이터는 실제 환경에서 수집되었으며, 정상 파일은 Androzoo 데이터세트를 통해 수집되었다. 정상파일 데이터의 세부 현황은 Table 1에서 보는 바와 같다.

Table 1. Details of Benign Dataset

Category	Amount of sample
Benign0	32,804
Benign1	47,861
Benign2	42,635
Benign3	7,847
Benign4	31,754
Total	162,901

악성코드는 VirusTotal을 활용하여 14개 클래스 및 191개 패밀리로 분류하였으며, 악성코드 클래스 분포는 Table 2에서 보는 바와 같다.

Table 2. Details of Malware in CICAndMal2020 Dataset

Category	Amount of sample
Adware	47,210
Backdoor	1,538
Banker	887
Dropper	2,302
Fileinfector	669
PUA	2,051
Ransomware	6,202
Riskware	97,349
Scareware	1,556
SMS	3,125
Spy	3,540
Trojan	13,559
No_Category	2,296
Zero_day	13,340
Total	195,624

분석을 위해 제공되는 특성은 정적 특성 및 동적 특성 두 가지로 구분된다. 정적 특성은 앱 권한, 의도(intent), 행위(activity) 등 AndroidManifest.xml 파일에서 추출된 총 9,503개의 고차원 정보로 구성되며, 대부분의 값이 0 또는 1인 희소행렬 구조를 가진다. 반면, 동적 특성은 가상환경에서 앱을 실행하여 수집된 메모리 사용량(23개), API 호출(105개), 배터리 소모량(2개), 네트워크 트래픽(4개) 등 총 141개의 행위 정보를 포함하고 있다. 이처럼 CCCS-CIC-AndMal-2020 데이터세트는 균형 잡힌 구성에 더해 상세한 특성집합을 포함하고 있어 고도화된 악성코드 탐지 모델의 개발과 검증을 위한 포괄적인 데이터세트로 활용되고 있다.

2. Related works

안드로이드 악성코드 탐지 및 분류와 관련된 최근 연구들은 악성 행위를 정확하게 식별하기 위해 애플리케이션의 정적·동적 특성, 특히 API Call 정보를 적극적으로 활용하고 있다. 그러나 API Call 정보를 기반으로 생성된 데이터세트는 일반적으로 수천 개에 달하는 고차원 특성을 포함하고 있어 분석 과정에서 계산 복잡성과 자원 소모 문제를 야기한다.

이러한 비효율성을 해결하고 모델의 성능을 최적화하기 위해 최신 연구들은 차원 축소를 핵심적인 전처리 단계로 채택하는 경향을 보인다. 데이터의 차원을 축소하는 방법은 크게 두 가지로 구분된다. 첫 번째는 특성 선택(feature selection) 방식으로 통계적 기법이나 트리 기반 앙상블 모델 등을 활용하여 전체 특성 중 악성코드 분류에 가장 유의미한 영향을 미치는 핵심 특성만을 선별한다. 두 번째는 특성 추출(feature extraction) 방식으로 PCA (Principal Component Analysis)와 같이 기존 특성들을 수학적으로 조합하여 원본 데이터의 분산을 최대한 유지하는 새로운 저차원 특성 공간을 생성한다.

Rahali 등[5]은 본 연구의 실험에서 사용하는 CCCS-CIC-AndMal-2020 데이터세트를 직접 생성하였다. 그리고 Extra-Tree Classifier를 이용하여 API Call을 포함한 9,503개의 특성을 2,200여 개로 줄인 후 2D 이미지로 변환하여 CNN 모델을 적용해 악성코드 분류 성능을 분석하였다. 이진분류는 실시하지 않았으며, 다중분류 정확도는 83.22%로 나타났다. 그러나 일반적인 컴퓨팅 환경이 아닌 50개의 CPU와 500GB의 RAM이 장착된 고성능 서버에서 실험이 이루어져 현실적인 환경에서의 적용 가능성을 확인하기는 어렵다는 한계가 있다.

Hwang 등[6]은 CCCS-CIC-AndMal-2020 데이터세트의 방대한 API Call 정보의 차원을 축소하여 핵심 특성집합을 추출하는 방안을 제시하였다. 실험 환경의 한계로 인해 전체 데이터세트를 활용하지는 않았으며, 7,000개의 정상 파일과 7,200개의 악성 파일을 무작위로 추출해 서브 데이터세트를 구성한 후 실험을 진행하였다. 다양한 특성 선택 기법들을 적용해 9,503개의 특성을 이진분류에서는 약 15%, 다중분류에서는 약 8% 수준으로 축소한 후 이미지로 변환하고 CNN 모델을 기반으로 분류 성능을 측정하였다. 그 결과 이진분류에서는 97.09%, 다중분류에서는 83.4%의 정확도를 달성하여 제한된 데이터와 특성만으로도 효율적인 분류가 가능함을 입증하였다.

Jeon 등[7]은 고차원 API Call 정보로 인한 분석 과정에서의 비효율성 문제를 해결하기 위해 PCA를 이용한 차원 축소 기법을 제안하였다. PCA를 통해 9,503개 특성을 1% 미만인 70~100개의 주성분으로 축소한 후, LightGBM, Random Forest, KNN 모델에 적용하였다. 실험 결과 이진분류는 96.8%, 다중분류는 약 87%의 정확도를 달성했으며, 데이터 샘플링과 차원 축소를 통해서도 기존 연구보

다 높은 성능을 확보할 수 있음을 입증하였다.

Musikawan 등[8]은 각 은닉층의 예측 결과를 결합하는 앙상블 구조를 적용하여 분류 성능을 높인 심층 신경망 모델을 제안하였다. 별도의 특성 선택 알고리즘은 사용하지 않았으며, 다수의 서브 네트워크를 통해 9,503개의 전체 특성으로부터 자동으로 의미 있는 특성 표현을 학습하도록 설계하였다. 실험 결과 이진분류 정확도는 97.72%로 나타났다.

Chopra 등[9]은 에너지 효율적이고 계산 비용이 낮은 악성코드 탐지 기법을 제안하였다. 앱 설치 시 생성되는 OAT(Of Ahead Time) 파일을 힐버트공간 채움곡선을 이용해 2D 이미지로 변환한 후 CNN, LeNet, AlexNet 및 전이학습모델을 적용하여 이진분류를 수행하였다. 실험 결과 전이학습모델이 97.19%로 가장 높은 정확도를 보였다.

Polatidis 등[10]은 안드로이드 악성코드 탐지에서 알고리즘의 복잡성과 고차원 특성 사용 문제를 해결하기 위해 효과적인 부분 특성집합 선택 방법론을 제안하였다. 데이터 이진화를 통한 전처리와 통계적 기준을 적용하여 특성을 필터링한 결과 9,503개의 특성이 9 ~ 27개의 특성으로 감소되었다. 이후 다중 분류의 복잡성을 해결하고자 One-vs-One 전략을 채택하여 Decision Tree와 Random Forest 모델을 기반으로 한 이진분류를 수행한 결과, 특성 수가 크게 줄었음에도 95% 이상의 높은 정확도를 유지하였다. 그러나 통계적 기준(60%, 70%)은 특정 데이터세트에 최적화된 값으로 다른 데이터세트에서는 동일한 성능을 보장하기 어렵다는 한계가 존재한다.

Ghourabi[11]은 어텐션(attention) 메커니즘을 특성 선택에 활용하는 새로운 접근법을 제안하였다. 신경망의 어텐션 계층을 통해 5,911개 특성에 대한 중요도를 가중치로 산출한 뒤, 가중치가 높은 상위 특성집합을 선택하였다. LightGBM 모델로 분류를 수행한 결과, 이진분류에서는 300개의 특성으로 98.71%의 정확도를, 다중분류에서는 500개의 특성으로 F1-Score 86.33%를 달성하였다.

Jung 등[12]은 CCCS-CIC-AndMal-2020 데이터세트에 포함된 APK 파일을 디컴파일하여 획득한 Smali 코드를 자연어 텍스트로 간주하고, 이를 BERT, RoBERTa 및 BART 모델로 직접 학습시키는 접근법을 제안하였다. 실험 결과, 이진분류에서 97.81%의 높은 정확도를 달성하여 트랜스포머 기반 모델이 Smali 코드 분석에 유효함을 입증하였다.

고차원 데이터세트에 적용하는 특성 선택 단계를 고도화하여 기계학습 모델의 분류 성능 향상을 시도했던 연구도 있다. Mahindru 등[13]은 필터(filter) 방식과 래퍼(wrapper) 방식을 결합하는 2단계 하이브리드 특성 선택 프레임워크를 제안하였다. 1단계에서는 필터 방식(t-test 및 ULR)으로 특성을 선별하고, 2단계에서는 래퍼 방식(상관관계 분석 및 다변량 회귀)으로 최적의 특성집합을 확정하였다. 확정된 특성집합을 기반으로 DNN 모델에서 분류를 시도한 결과 98.8%의 높은 정확도를 달성하였다.

Ele 등[14]은 'FilWrap'이라는 하이브리드 방식의 접근법을 제안하였다. [13]과 유사하게 필터 방식(Chi-Square)과 래퍼 방식(RFE)을 결합한 후, 유전 알고리즘(Genetic Algorithm)으로 최종 최적화를 수행하였다. 그 결과 215개의 원본 특성을 20개까지 축소했음에도 불구하고 KNN 분류기에서 100%의 분류 성능을 달성하였다.

동적 분석을 기반으로 한 연구도 활발하게 진행되었다. Keyes 등[15]은 데이터세트 제작 연구의 후속으로 CCCS-CIC-AndMal-2020의 동적 특성을 분석하는 EntropLyzer를 제안했다. 가상환경에서 악성코드를 실행해 141개의 동적 특성을 추출하였으며, 재부팅 전후 행위 변화를 포착하기 위해 섀넌 엔트로피를 계산하여 분석에 활용하였다. 추출된 141개 동적 특성 전체를 Decision Tree 모델에 적용하여 12개의 악성코드 카테고리를 분류한 결과, 98.4%의 Precision 및 98.3%의 Recall을 달성하였다. 정적 분석 기반 연구와의 직접적인 성능 비교는 어렵지만 동적 특성 분석만으로도 높은 수준의 분류가 가능하다는 점을 확인하였다. 그러나 에뮬레이터 환경을 인지하고 실행되지 않는 악성코드가 있어 실제보다 적은 수의 샘플만 분석되었다는 점은 연구의 한계로 지적되고 있다.

Ababneh 등[16]은 CCCS-CIC-AndMal-2020 데이터세트의 동적 특성을 기반으로 4개의 특정 악성코드 카테고리에 대한 분류 프레임워크를 제시하였다. 정보 이득과 CFS(Critical Success Factor) 평가기를 결합한 후 수동 분석을 통해 141개의 동적 특성을 24개까지 줄였다. Random Forest 모델을 적용하여 4개 카테고리를 분류한 결과, 재부팅 후 데이터에서 98.96%의 매우 높은 정확도를 달성하였다. 이는 소수의 핵심 동적 특성만으로도 특정 유형의 악성코드를 정밀하게 탐지할 수 있음을 보여주지만 4개의 카테고리에만 국한된 실험으로 전체 악성코드 유형에 대한 일반화 성능을 확인하기는 어렵다.

Sraratee 등[17]은 동적 특성 분석의 효율성을 높이기 위해 특성 선택과 차원 축소를 통합한 2단계 파이프라인을 제안하였다. 상호 정보량(Mutual Information)을 이용해 141개 동적 특성 중 50개를 선택하고 이후 PCA를 통해 최종적으로 33개의 주성분으로 축소하였다. Random Forest 모델 이용 다중분류 결과 98%의 높은 정확도를 달성하였다. 이를 통해 원본 특성의 76% 이상을 제거하면서도 높은 성능을 유지하여 경량화와 효율성을 입증하였다.

III. The Proposed Scheme

본 연구에서 제안하는 접근방법은 특성이 고차원이면서 희소행렬 구조인 CCCS-CIC-AndMal-2020 데이터세트에서 핵심 특성만을 효율적으로 선택하여 경량화된 분류 모델을 구축하는 것을 목표로 한다. 이를 위해 단순한 단일 기법이 아닌 3단계로 구성된 체계적인 특성 선택 파이프라인을 설계했으며, 전체 흐름은 Fig. 1에서 보는 바와 같다. 이 파이프라인은 광범위한 데이터 정제부터 모델별 최적화까지 점진적으로 특성 공간을 정제해 나간다.

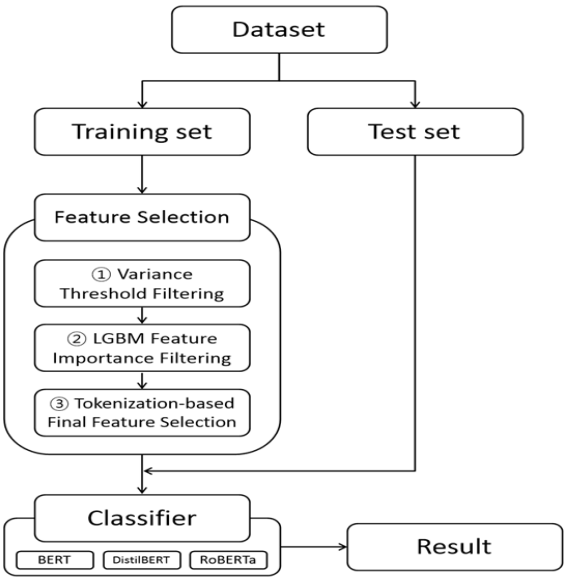


Fig. 1. Overall Architecture of the Proposed Model

먼저, API Call 정보를 포함하는 CCCS-CIC-AndMal-2020 데이터세트와 정상 파일인 Androzoo를 대상으로 분산 임계값 필터링을 수행한다(①). 이 단계는 본격적인 특성 선택에 앞서 모든 샘플에서 동일한 값을 가져 분류에 기여하지 못하는 무의미한 특성(분산=0)을 제거하는 데이터 정제 과정이다. 이를 통해 불필요한 노이즈를 사전에 제거하고 후속 단계의 계산 효율성을 높인다. 1차 정제된 데이터에 LightGBM 기반 특성 중요도 필터링을 적용한다(②). 이 단계는 모델 학습 과정에서 특성의 중요도를 평가하는 임베디드 기법 중 트리 기반 앙상블 모델인 LightGBM이 각 특성의 악성코드 분류 기여도를 정보 이득을 기반으로 정량화한다. 이 과정에서 모든 유효 특성의 기여도 기준으로 내림차순 정렬한 목록을 생성하며, 이는 다음 단계에서 최종 특성을 선택하는 기준이 된다.

마지막으로 토큰화 기반 최종 특성 선택 단계를 수행한다(③). 이 과정은 2단계에서 생성된 중요도 순위 목록을 바탕으로 최종 분류기로 사용할 각 트랜스포머 모델의 아키텍처 제약조건을 충족시키는 최적의 특성 부분집합을 결정한다. 중요도가 높은 특성부터 순서대로 'FEAT_0 : 1'과 같은 문자열 형태로 변환하여 하나의 긴 시퀀스를 구성한 뒤, 각 모델의 고유한 토크나이저로 토큰화를 진행한다. 이때, 모델의 최대 입력 길이인 512 토큰을 초과하지 않는 범위까지 가장 중요한 특성들을 순차적으로 포함시킨다. 각 모델이 사용하는 토크나이저의 차이로 인해, 동일한 512 토큰 제약 하에서 포함할 수 있는 최대 특성의 개수가 모델별로 상이하게 결정된다. 이렇게 최종적으로 선별된 핵심 특성집합을 기반으로 이진분류 및 다중분류를 실시한다.

1. Experimental Dataset

실험에 사용된 악성코드 데이터는 원본 데이터세트 제작자들이 실험과정에서 배제할 것을 권고했던 No_Category와 Zero_day 클래스 데이터를 제외한 12개 클래스의 데이터만 사용하였다. No_Category 클래스는 특성 정보를 명확하게 분류하기 힘든 악성코드로 구성되어 있어 특성 학습을 통한 분류가 불가능하다. Zero_day는 하위 패밀리가 다양하여 고유한 특성을 파악하기 어려워 모델 학습 후 정확한 성능 검증을 방해할 수 있다.

실험은 정상파일과 악성코드를 구분하는 이진분류와 12개 악성코드 클래스를 분류하는 다중분류로 구분하여 진행하였다. 두 실험 모두 8:2 비율로 학습 및 테스트 데이터를 분리하였으며, 이진분류 및 다중분류 실험을 위한 데이터세트 구성은 Table 3 및 Table 4에서 보는 바와 같다.

Table 3. Experimental Dataset for Binary classification

Class	Label	Train data	Test data
Benign	0	129,745	32,436
Malware	1	143,991	35,997
Total	-	273,736	68,433

Table 4. Experimental Dataset for Multi-class classification

Class	Label	Train	Test
Adware	0	37,768	9,442
Backdoor	1	1,230	308
Banker	2	709	178
Dropper	3	1,841	461
Fileinfector	4	535	134
PUA	5	1,640	411
Ransomware	6	4,961	1,241
Riskware	7	77,879	19,470
Scareware	8	1,244	312
SMS	9	2,500	625
Spy	10	2,832	708
Trojan	11	10,847	2,712
Total	-	143,986	36,002

2. Multi-stage Feature Selection Pipeline

본 절에서는 고차원의 희소행렬 구조를 가지는 CCCS-CIC-AndMal-2020 데이터세트에서 노이즈를 제거하고 모델의 예측 성능에 직접적으로 기여할 핵심 특성만을 효율적으로 선별하기 위해 3단계로 구성된 다단계 특성 선택 파이프라인에 대해 보다 상세하게 설명한다.

2.1 Variance Threshold Filtering

데이터 전처리의 일환으로 모든 샘플에서 동일한 값을 가져 분류에 어떠한 정보도 제공하지 못하는 무의미한 특성을 제거하는 과정이다. 분산이 0인 특성은 모든 데이터에 걸쳐 변화가 없기 때문에 모델 학습에 노이즈로 작용할 수 있다. 따라서 분산 임계값을 0으로 설정하여 이러한 단일 값 특성들을 일괄적으로 제거함으로써 데이터의 정합성을 높이고 다음 단계의 분석에 필요한 계산 효율성을 확보하였다.

2.2 LightGBM-based Feature Importance Extraction

임베디드 기법을 활용하여 모델의 예측 성능에 직접적으로 기여하는 특성을 식별하고 순위를 매긴다. 본 연구에서는 결정 트리 기반의 앙상블 모델인 LightGBM을 특성 선택 도구로 사용하였다.

LightGBM은 학습 과정에서 각 결정 트리의 노드를 분기할 때 정보 이득을 최대로 하는 특성을 우선적으로 선택한다. 모델 학습이 완료된 후 앙상블을 구성하는 모든 트리에서 각 특성이 분기에 사용된 총 횟수나 이를 통해 얻은 총 정보 이득을 기반으로 특성 중요도를 산출한다. 이러한 과정을 통해 2.1의분산 임계값 필터링에서 제거되지 않은 모든 유효 특성들에 대해 악성코드 분류 기여도를 기준으로 내림차순 순위를 부여한 목록을 생성한다.

2.3 Tokenization-based Final Feature Selection

마지막 단계에서는 2단계에서 생성된 특성 중요도 순위 목록을 바탕으로 최종 분류기로 사용할 트랜스포머 모델의 아키텍처 제약조건을 충족하는 최적의 특성 부분집합을 결정한다. 이 과정은 다음과 같은 순서로 수행된다.

① 중요도 기반 특성 시퀀스 생성

각 데이터 샘플의 특성들을 2단계에서 산출된 특성 중요도가 높은 순서대로 나열한다. 나열된 특성들은 'FEAT_0 : 1'과 같은 문자열 형태로 변환되어 하나의 긴 시퀀스를 구성한다.

② 모델별 토큰화 및 특성 수 결정

과정 ①을 통해 구성된 문자열 시퀀스를 각 트랜스포머 모델의 토크나이저(tokenizer)로 입력한다. 토큰의 총 개수가 모델의 최대 입력 길이인 512 토큰을 초과하지 않을 때까지 중요도가 높은 특성 문자열을 순차적으로 시퀀스에 포함시킨다.

BERT 및 DistilBERT 모델의 토크나이저(WordPiece)와 RoBERTa 모델의 토크나이저(BPE)는 동일하지 않아 입력 가능한 최대 특성의 수는 달라진다. 최종적으로 BERT 및 DistilBERT 모델을 위해서는 95개, RoBERTa 모델을 위해서는 80개의 특성이 선택되었다. 이 과정을 통해 모델별로 최적화된 가장 정보 밀도가 높은 최종 특성집합이 완성된다.

3. Transformer-based Classifier

선별된 핵심 특성집합을 기반으로 악성코드를 분류하기 위해 본 연구에서는 트랜스포머 기반의 언어 모델을 분류기로 활용한다. 트랜스포머는 셀프 어텐션 메커니즘을 통해 입력 시퀀스 내 요소들 간의 복잡한 상호 연관성을 효과적으로 학습할 수 있어 API Call 특성들 간의 잠재적인 행위 패턴을 파악하는 데 적합하다.

3.1 Model Input Configuration

2.3의 과정을 통해 최종적으로 선택된 95개 또는 80개의 특성 시퀀스는 트랜스포머 모델의 입력으로 사용되기 위해 다음과 같이 최종 형태로 변환된다.

먼저, 모든 특성 시퀀스의 맨 앞에는 분류 작업을 위한 특수 토큰인 [CLS]가 추가된다. 이 [CLS] 토큰은 트랜스포머 인코더를 통과한 후, 전체 시퀀스의 의미를 함축하는 대표 벡터로 사용되어 최종 분류 계층에 전달된다. 이후, 모든 시퀀스의 길이를 통일하기 위해 각 시퀀스의 끝에는 [PAD] 토큰을 추가하는 패딩(Padding) 과정이 수행된다.

[CLS]와 [PAD]가 추가된 특성 시퀀스는 모델의 임베딩 계층으로 전달되어 고차원 벡터 시퀀스로 변환된다. 이 과정에서 각 토큰은 고유한 의미를 나타내는 토큰 임베딩과 시퀀스 내에서의 위치 정보를 나타내는 위치 임베딩으로 변환된 후, 두 임베딩이 합산된다. 이렇게 완성된 최종 임베딩 벡터 시퀀스가 트랜스포머 인코더의 실제 입력으로 사용된다.

3.2 Model Architecture

본 연구에서는 제안하는 특성 선택 파이프라인의 효과를 검증하기 위해 자연어 처리 분야에서 성능이 입증된 3종의 대표적인 트랜스포머 기반 사전 학습 모델을 분류기로 채택하였다. 각 모델은 아키텍처와 학습 방식에서 명확하게 구별되는 특징을 가지므로 성능과 효율성 측면에서 다각적인 비교 분석을 수행하였다.

BERT[18]는 2018년 구글에서 발표한 모델로서 양방향 컨텍스트 학습을 통해 자연어 이해의 새로운 지평을 열었다는 평가를 받고 있다. 기존 언어 모델들과 달리 MLM 기법을 기반으로 문장 내에서 단어의 좌우 문맥을 동시에 고려하는 BERT의 양방향 학습 방식이 API Call 특성 시퀀스 내의 복잡한 상호 관계를 파악하는 데 효과적일 것으로 판단하였다. 이에 본 연구에서는 12개의 인코더 레이어로 구성된 BERT-Base 모델을 실험에 사용하였다.

DistilBERT[19]는 Hugging Face가 개발한 BERT의 경량화 버전이다. 지식 증류(knowledge distillation) 기법을 사용하여 BERT-Base의 성능을 약 97% 수준으로 유지하면서도 레이어 수를 절반(6개)으로 줄여 모델의 전체 파라미터 수를 40% 가량 감소시켰다. 그 결과 추론 속도가 60% 이상 향상되는 높은 효율성을 보인다. 본 연구에서는 경량 모델이 고차원의 악성코드 데이터를 얼마나 효율적으로 처리할 수 있는지 탐색하기 위해 DistilBERT를 실험에 포함하였다.

RoBERTa[20]는 Meta AI가 BERT의 사전 학습 방식을 개선하여 제안한 모델로서 학습 과정을 견고하게 최적화하는 데 초점을 맞췄다. 주요 개선점으로는 더 큰 배치 크기와 더 많은 데이터 사용, NSP 손실 함수 제거, 마스킹 패턴을 동적으로 변경하는 기법 적용 등이 있다. 이러한 최적화를 통해 RoBERTa는 BERT보다 더 안정적이고 우수한 다운스트림 태스크 성능을 보이는 것으로 알려져 있다.

IV. Experimental Result and Analysis

1. Experimental Environment and Evaluation Metrics

실험은 구글에서 제공하는 Colab pro+ 환경(python 3.12.11 버전, A100 GPU, 80GB RAM)에서 진행되었다.

모델의 성능은 분류 모델 평가에 널리 사용되는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 및 F1-Score 네 가지 지표를 사용하여 종합적으로 평가하였다. 본 논문에서 제시하는 모든 성능 지표는 무작위성에 의한 결과 왜곡을 최소화하고 모델의 안정적인 성능을 검증하기 위해 10회 반복 실험을 통해 산출된 결과의 평균값이다.

다중분류 성능 평가에서는 클래스 간 데이터 불균형을 고려하여 각 클래스를 동등하게 평가하기 위해 매크로 평균 방식을 적용하였다. 매크로 평균은 각 클래스에 대해 개별적으로 성능 지표를 계산한 후 이들의 산술 평균을 취하는 방식이다. 이를 통해 샘플 수가 적은 소수 클래스에서의 성능 저하가 전체 점수에 명확하게 반영되도록 하였다.

2. Binary Classification

본 연구에서 제안하는 다단계 특성 선택 파이프라인과 트랜스포머 모델의 성능을 검증하기 위해 악성코드와 정상 파일을 판별하는 이진분류 실험을 수행하였다. RoBERTa, BERT 및 DistilBERT 세 가지 모델에 대한 분류 성능 평가 결과는 Table 5에서 보는 바와 같다.

Table 5. Results of Binary Classification

Model	Accuracy	Precision	Recall	F1-Score
BERT	98.22	98.21	98.22	98.21
DistilBERT	98.25	98.22	98.27	98.24
RoBERTa	98.28	98.26	98.29	98.27

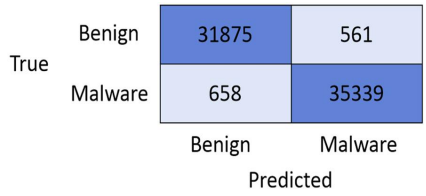
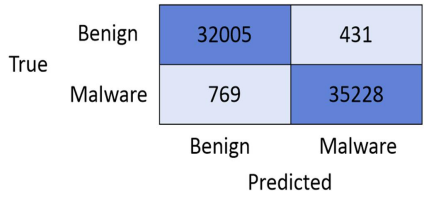
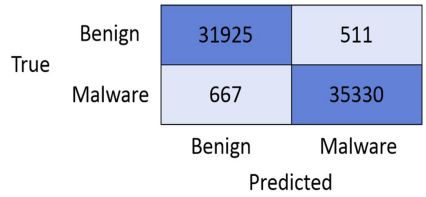
세 가지 모델 모두 98.2% 이상의 높은 정확도와 F1-Score를 기록하여 제안하는 특성 선택 기법이 악성코드 분류에 효과적임을 확인하였다. 특히, RoBERTa 모델이 98.28%의 정확도와 98.27%의 F1-Score를 달성하여 미세한 차이로 가장 우수한 종합 성능을 보였다.

모델별 예측 오류를 보다 상세하게 분석하기 위해 혼동행렬을 확인한 결과 각 모델별로 강점이 달라지는 것을 식별할 수 있었다. 각 모델의 혼동행렬은 Table. 6에서 보는 바와 같다.

BERT 모델은 실제 악성코드를 정상으로 잘못 판단한 미탐지(False Negative) 건수가 658개로 세 모델 중 가장 적었다. 이는 실제 위협을 놓치지 않는 탐지 능력이 가장 뛰어남을 의미한다. 따라서 BERT 모델은 보안성을 최우선으로 고려해야 하는 환경에 적합한 모델이 될 수 있다. 반면, 경량화 모델인 DistilBERT는 정상 파일을 악성코드로 잘못 판단한 오탐지(False Positive) 건수가 431개로 가장 적었다. 이는 사용자에게 불필요한 경고를 최소화하여 사용자 경험을 해치지 않는 것이 중요한 환경에서 가장 큰 강점을 가진다. 그러나 미탐지 건수는 769개로 가장 많아 탐지율 측면에서 가장 낮은 성능을 보였다. RoBERTa는 종합적인 성능 지표는 가장 높았으나 미탐지(667개)와 오탐지(511개) 측면에서는 우수한 성능을 달성하지 못했다.

결론적으로 제안된 특성 선택 파이프라인은 모든 트랜스포머 모델에서 높은 성능을 이끌어냈다. 특히, 최대 탐지율을 목표로 할 경우 BERT가 적합하며, 사용자 경험을 중시하고 오탐을 최소화해야 할 경우 경량 모델인 DistilBERT가 효과적인 대안이 될 수 있음을 확인하였다.

Table 6. Confusion Matrix of Binary Classification

Model	Confusion Matrix
BERT	
DistilBERT	
RoBERTa	

본 연구에서 제안하는 특성 선택 파이프라인이 적용된 모델의 이진분류 성능을 객관적으로 평가하기 위해 동일한 데이터세트를 사용한 연구들과 성능을 비교한 결과는 Table 7에서 보는 바와 같다.

Table 7. Performance comparison of Binary Classification

	Approach	Feature	Accuracy	F1-Score
[6]	ANOVA+ CNN	1,496	97.09	96.74
[7]	PCA+ LightGBM	100	96.80	96.79
[8]	Enhaced DNN	9,503	97.72	97.72
[11]	Attention+ LightGBM	300	98.71	98.56
Proposed	LightGBM+ RoBERTa	80	98.28	98.27

본 연구에서 제안하는 RoBERTa 기반 모델은 단 80개의 특성만으로 98.28%라는 높은 정확도를 기록하였다. 한편, [11]에서는 300개의 특성으로 98.71%라는 가장 높은 이진분류 정확도를 달성했지만, 본 연구에서 제안하는 모델은 훨씬 적은 수의 특성으로 거의 대등한 수준의 성능을 기록했다는 점에서 큰 의의를 가진다. 그리고 이러한 결과를 통해 본 연구에서 제안하는 다단계 특성 선택 파이프라인이 불필요한 노이즈를 효과적으로 제거하면서 모델의 경량화와 효율성 측면에서도 강점을 가짐을 확인할 수 있다.

3. Multi-Class Classification

12개의 악성코드 클래스를 식별하는 다중분류 실험을 통해 제안된 방법론의 세분화된 탐지 성능을 평가하였다. 이진분류에서와 동일하게 RoBERTa, BERT 및 DistilBERT 세 가지 모델을 사용하였으며, 클래스 불균형 문제를 고려해 Macro F1-Score를 핵심 성능 지표로 삼았다. 전체적인 실험 결과는 Table 8에서 보는 바와 같다.

Table 8. Results of Multi-Class Classification

Model	Accuracy	Precision	Recall	F1-Score
BERT	93.59	85.61	82.22	83.66
DistilBERT	93.41	85.18	80.57	82.56
RoBERTa	93.69	86.14	81.37	83.45

세 가지 모델 모두 약 93.5% 수준으로 정확도가 매우 높게 나타났으나, 이는 데이터 수가 압도적으로 많은 특정 클래스(Class 7, Riskware)의 영향이 크게 반영된 결과이다. 따라서 각 클래스를 동등하게 평가하는 Macro F1-Score를 기준으로 보면 BERT가 83.66%로 가장 우수한 성능을 기록했으며 이진분류에서 가장 우수한 성능을 보인 RoBERTa가 근소한 차이로 그 뒤를 이었다.

악성코드 클래스별 분류 성능을 세부적으로 분석한 결과, 모델의 탐지 능력은 악성코드 유형에 따라 상당한 편차를 보였다. 다른 클래스에 비해 데이터 수가 많은 Riskware(Class 7)와 Adware(Class 0)의 경우 정탐지 건수가 압도적으로 높아 매우 안정적으로 분류됨을 확인하였다. 예를 들어, RoBERTa 모델은 실제 Riskware 데이터 19,470개 중 18,832개를, Adware 데이터 9,442개 중 8,905개를 정확하게 분류하였다. 반면, PUA(Class 5)는 세 모델 모두에서 가장 분류하기 어려운 유형으로 나타났다.

DistilBERT 모델의 경우 PUA(Class 5) 데이터 411개 중 단 224개만을 올바르게 분류하여 정탐지율이 약 54.5%에 그쳤다. 또한, Backdoor(Class 1) 역시 RoBERTa 모델 기준 정탐지율이 약 70.1%로 상대적으로 낮은 성능을 보였다.

이러한 오분류 경향은 혼동 행렬을 통해서 더욱 명확하게 확인할 수 있다. 다중분류에 대한 혼동 행렬은 Table. 9에서 보는 바와 같다.

분류 성능이 가장 저조한 PUA(Class 5) 데이터의 상당수가 Adware(Class 0) 또는 다수 클래스인 Riskware(Class 7)로 잘못 예측되는 패턴이 공통적으로 관찰되었다. 이러한 결과는 이들 악성코드의 API Call 기반 행위 특성이 서로 유사하여 모델이 명확하게 구분하는 데 어려움을 겪었음을 의미한다.

결론적으로 제안된 특성 선택 파이프라인과 트랜스포머 모델들은 다중분류에서도 전반적으로 높은 성능을 보였다. 특히, BERT 모델이 가장 균형 잡힌 성능을 제공하였으나 데이터 수가 적고 행위 패턴이 모호한 특정 클래스에 대해서는 여전히 분류 성능 개선이 필요한 과제로 남았다.

다중분류 성능을 객관적으로 평가하기 위해 동일 데이터 세트를 사용했던 선행 연구들과 Macro F1- Score를 중심으로 성능을 비교한 결과는 Table 10에서 확인할 수 있다.

Table 9. Confusion Matrix of Multi-Class Classification

True \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11
0	8856	30	3	13	1	61	37	354	4	6	4	73
1	29	226	0	0	0	2	11	27	4	3	1	5
2	4	0	150	4	0	2	6	3	0	1	1	7
3	11	2	5	322	0	2	78	12	0	0	3	26
4	15	0	10	0	99	1	0	0	0	0	0	3
5	73	1	0	2	0	255	1	60	3	0	0	16
6	3	0	2	16	0	2	1170	9	3	1	30	5
7	410	21	1	14	5	56	55	18807	9	23	9	60
8	31	2	0	0	0	3	4	6	233	0	0	33
9	5	6	1	1	0	2	6	13	0	563	15	13
10	3	0	0	5	0	6	64	14	1	0	601	14
11	103	6	11	20	0	22	16	89	1	18	14	2412

(a) BERT

True \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11
0	8876	29	1	11	3	55	35	347	3	7	3	72
1	38	211	0	0	0	1	11	31	4	7	1	4
2	8	0	149	5	0	0	5	4	0	3	0	4
3	8	3	5	324	0	3	79	18	1	1	3	16
4	16	0	9	0	98	0	0	8	0	0	0	3
5	99	1	0	2	0	224	1	62	2	0	1	19
6	5	2	2	21	1	1	1163	7	2	1	29	7
7	402	24	1	14	5	34	53	18815	5	28	7	82
8	33	4	0	0	0	1	2	21	218	1	0	32
9	9	3	2	1	0	2	7	12	1	560	10	18
10	2	0	0	3	1	6	63	15	1	4	599	14
11	107	6	9	20	0	18	18	94	5	24	18	2393

(b) DistilBERT

True \ Predicted	0	1	2	3	4	5	6	7	8	9	10	11
0	8905	21	0	6	2	71	36	331	5	7	3	55
1	30	216	0	0	1	0	11	35	4	2	1	8
2	4	0	147	8	0	0	6	5	0	1	1	6
3	9	0	7	323	0	3	74	14	1	1	4	25
4	15	0	9	0	97	1	0	9	0	1	0	2
5	79	2	0	0	1	253	2	57	1	1	2	13
6	0	0	2	18	0	0	1165	5	0	0	38	5
7	396	20	0	17	5	46	52	18832	2	18	6	76
8	29	5	0	0	0	1	2	21	221	0	0	33
9	12	8	0	0	0	3	8	11	0	561	9	13
10	5	0	1	3	0	4	60	18	0	2	605	10
11	117	5	6	27	0	16	19	80	4	16	15	2407

(c) RoBERTa

⓪ Adware	① Backdoor	② Banker
③ Dropper	④ FileInfector	⑤ PUA
⑥ Ransomware	⑦ Riskware	⑧ Scareware
⑨ SMS	⑩ Spy	⑪ Trojan

Table 10. Performance comparison of Multi-Class Classification

	Approach	Feature	F1-Score
[5]	Extra-Tree+ CNN	2,237	83
[6]	MI + CNN	593	82.26
[7]	PCA + RF	70	86.94
[11]	Attention+ LightGBM	500	86.83
Proposed	LightGBM+ BERT	95	83.66

제안하는 모델의 성능이 모든 선행 연구의 성능을 능가하지는 못했지만, 사용한 특성 개수 대비 매우 효율적이고 경쟁력 있는 성능을 달성하였음을 확인할 수 있다.

2,237개의 특성으로 83%의 F1-Score를 달성한 선행연구[5]와 비교하여 BERT 기반 모델은 특성의 수를 약 4% 수준인 95개만 사용하면서도 더 높은 83.66%의 F1-Score를 달성하였다. 이는 제안하는 특성 선택 파이프라인이 원본 데이터의 성능을 유지하면서도 모델을 극도로 경량화하는 데 성공했음을 보여준다. 500개의 특성으로 86.83%의 F1-Score를 달성한 연구[11]와 비교하면 성능은 다소 낮지만 사용된 특성의 수는 5분의 1 미만에 불과하다. 이는 약간의 성능 차이를 감수하더라도 계산 비용과 메모리 사용량 측면에서 훨씬 효율적인 모델을 구축하였음을 의미한다.

가장 높은 성능을 보인 [7]의 연구는 PCA를 통해 70개의 새로운 특성을 추출하여 86.94%의 F1- Score를 달성하였다. 제안 모델의 성능이 이에 미치지는 못하지만 PCA가 원본 특성을 조합하여 해석이 어려운 새로운 특성을 만드는 반면, 본 연구는 해석 가능한 원본 API Call 특성 95개를 직접 선택하여 유사한 성능에 도달했다는 점에서 차별점을 가진다.

결론적으로 제안하는 방법론은 극소수 핵심 특성만으로도 수백, 수천 개의 특성을 사용한 선행 연구들과 대등하거나 더 나은 성능을 보이는 것을 확인하였으며, 이는 모델의 경량화와 성능 간의 균형을 효과적으로 달성하였음을 입증한다.

V. Conclusions

본 연구는 고차원의 희소 행렬 구조를 가지는 API Call 정보 기반의 CCCS-CIC-AndMal-2020 데이터세트를 효율적으로 분석하여 경량화와 높은 탐지 성능을 동시에 달성하는 안드로이드 악성코드 분류 모델을 제안하는 것을

목표로 하였다. 이를 위해 분산 임계값 필터링, LightGBM 기반 특성 중요도 필터링, 그리고 트랜스포머 모델의 토큰화 방식을 고려한 최종 특성 선택으로 구성된 다단계 특성 선택 파이프라인을 설계하고, 그 유효성을 BERT, DistilBERT 및 RoBERTa 모델을 통해 검증하였다.

실험 결과, 제안하는 특성 선택 파이프라인은 9,503개의 원본 특성을 약 1% 수준인 80~95개의 핵심 특성으로 대폭 축소하는 데 성공하였다. 이진분류 실험에서는 RoBERTa 모델이 98.28%의 높은 정확도를 기록하였으며, 다중분류 실험에서도 BERT 모델이 83.66%의 경쟁력 있는 Macro F1-Score를 달성하였다. 이는 극소수의 특성만으로도 수천 개의 특성을 사용한 선행 연구들과 대등하거나 더 우수한 성능을 보인 결과이며, 제안하는 방법론이 모델의 경량화와 효율성 측면에서 우위를 가짐을 확인시켜 준다. 또한, 혼동 행렬 분석을 통해 최대 탐지율을 목표로 할 경우에는 BERT 모델이 적합하며 사용자 경험을 중시하고 오탐을 최소화해야 할 경우에는 DistilBERT 모델이 효과적인 대안이 될 수 있다는 모델별 특성을 파악하였다.

다만, 본 연구는 몇 가지 한계점을 가지며 이는 다음과 같은 향후 연구 방향으로 이어질 수 있다. 첫째, 다중분류 실험 결과 원본 데이터세트의 클래스 불균형으로 인해 소수 클래스의 탐지 성능이 상대적으로 낮게 나타났다. 향후 연구에서는 SMOTE와 같은 오버샘플링 기법이나 Tomek Links와 같은 언더샘플링 기법을 적용하여 데이터 분포를 개선함으로써 소수 클래스에 대한 탐지 성능을 향상시키는 연구를 진행할 수 있다. 둘째, 본 연구는 정적 분석 기반의 API Call 특성에만 국한되었다. 향후에는 동적 분석을 통해 얻을 수 있는 행위 정보를 추가하여 정적-동적 특징을 결합한 하이브리드 분석 모델을 구축함으로써 더욱 정교하고 회피하기 어려운 탐지 시스템을 개발할 수 있을 것이다.

결론적으로 본 연구는 고차원 악성코드 데이터 분석에서 효과적인 특성 선택의 중요성을 입증하였고, 제안하는 파이프라인이 경량화와 고성능을 동시에 달성할 수 있는 실용적인 해결책임을 제시하였다는 점에서 그 의의를 찾을 수 있다.

REFERENCES

[1] Statcounter, “Mobile Operating System Market Share Worldwide”, https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-202408-202508

[2] McAfee, “Goldoson: Privacy-invasive and Clicker Android Adware found in popular apps in South Korea”, https://www.mcafee.com/blogs/other-blogs/mcafee-labs/goldoson-privacy-invasive-and-clicker-android-adware-found-in-popular-apps-in-south-korea/

[3] Zimperium, “2024 Global Mobile Threat Report”, https://lp.zimperium.com/hubfs/MAPS_MTD/REPORT/GEN/Global Mobile Threat Report 2024 FINAL (1).pdf

[4] UNB, “CCCS-CIC-AndMal-2020”, https://www.unb.ca/cic/datasets/andmal2020.html

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

[6] 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. 2021, DOI: 10.9708/jksci.2021.26.11.041

[7] Dong-Ha Jeon and Soojin Lee, “Light-weight Classification Model for Android Malware through the Dimensional Reduction of API Call Sequence using PCA”, Journal of The Korea Society of Computer and Information, Vol.27, No. 11, pp.123-130, Nov. 2022, DOI: 10.9708/jksci.2022.27.11.123

[8] P. Musikawan, Y. Kongsorot, I. You, and C. So-In, “An Enhanced Deep Learning Neural Network for the Detection and Identification of Android Malware”, IEEE Internet of Things Journal, Vol. 10, No. 9, pp. 7838-7851, May 2023, DOI: 10.1109/JIOT.2022.3194881

[9] R. Chopra, S. Acharya, U. Rawat, and R. Bhatnagar, “An Energy Efficient, Robust, Sustainable, and Low Computational Cost Method for Mobile Malware Detection”, Applied Computational Intelligence and Soft Computing, Vol. 2023, Article ID 2029064, 12 pages, Feb. 2023, DOI: 10.1155/2023/2029064

[10] N. Polatidis, S. Kapetanakis, M. Trovati, I. Korkontzelos, and Y. Manolopoulos, “FSSDroid: Feature subset selection for Android malware detection”, World Wide Web, Vol. 27, Article No. 50, July 2024, DOI: 10.1007/s11280-024-01287-y

[11] A. Ghourabi, “An Attention-Based Approach to Enhance the Detection and Classification of Android Malware”, Computers, Materials & Continua, Vol. 80, No. 2, pp. 2743-2760, Aug. 2024, DOI: 10.32604/cmc.2024.053163

[12] Won Sik Jung and Jae-Pyo Park, “Android Malware Detection Method Based on Smali Code Learning Using Natural Language Processing (Transformer) Model”, Journal of the Korea Academia-Industrial cooperation Society, Vol. 25, No. 10, pp. 922-928, Oct. 2024, DOI: 10.5762/ KAIS.2024.25.10.922

[13] A. Mahindru, H. Arora, A. Kumar, S. K. Gupta, S. Mahajan, S. Kadry and Jungeun Kim, “PermDroid a framework developed

using proposed feature selection approach and machine learning techniques for Android malware detection", Scientific Reports, Vol. 14, Article 10724, May 2024, DOI: 10.1038/s41598-024-60982-y

[14] S. Igbo, B. I. Ele, M. S. Julius, D. O. Egete, M. S. Ele and O. H. Anayo, "Optimized Hybrid Feature Selection Model for Automatic Malware Detection and Classification Using Network Traffic Android Data" , NIPES - Journal of Science and Technology Research, Vol. 7, No. 3, pp. 127-139, July 2025, DOI: 10.37933/nipes/7.3.2025.09

[15] 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", 2021 Reconciling Data Analytics, Automation, Privacy, and Security: A Big Data Challenge (RDAAPS), pp. 1-8, June 2021, DOI: 10.1109/RDAAPS48126.2021.9452002

[16] M. Ababneh, A. Al-Droos and A. El-Hassan, "Modern Mobile Malware Detection Framework Using Machine Learning and Random Forest Algorithm", Computer Systems Science and Engineering, Vol. 48, No. 5, pp. 1171-1191, Sep. 2024, DOI: 10.32604/csse.2024.052875

[17] A. Al-Sraratee and A. Al-Azawei, "CLASSIFYING ANDROID MALWARE CATEGORIES BASED ON DYNAMIC FEATURES: AN INTEGRATION OF FEATURE REDUCTION AND SELECTION TECHNIQUES", Kufa Journal of Engineering, Vol. 16, No. 2, pp. 96-118, Apr. 2025, DOI: 10.30572/2018/KJE/160206

[18] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Minneapolis, Minnesota, Vol. 1, pp. 4171-4186, Jun. 2019, DOI: 10.18653/v1%2FN19-1423

[19] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter", arXiv preprint arXiv:1910.01108, Oct. 2019.

[20] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "RoBERTa: A Robustly Optimized BERT Pretraining Approach", arXiv:1907.11692, Jul. 2019, DOI: 10.48550/arXiv.1907.11692

Authors

Gwang-Ho Kim received the B.S. degrees in Foreign Languages from the Korea Naval Academy. He is currently a graduate student in the Department of Cyber and Computer Science, Korea National Defense University. His research interests include Machine learning, the security of Artificial Intelligence 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.