KIPS Trans. Softw. and Data Eng.
Vol.8, No.11 pp.427~432 pISSN: 2287-5905

https://doi.org/10.3745/KTSDE.2019.8.11.427

Modeling and Selecting Optimal Features for Machine Learning Based Detections of Android Malwares

Kye Woong Lee† · Seung Taek Oh†† · Young Yoon†††

ABSTRACT

In this paper, we propose three approaches to modeling Android malware. The first method involves human security experts for meticulously selecting feature sets. With the second approach, we choose 300 features with the highest importance among the top 99% features in terms of occurrence rate. The third approach is to combine multiple models and identify malware through weighted voting. In addition, we applied a novel method of eliminating permission information which used to be regarded as a critical factor for distinguishing malware. With our carefully generated feature sets and the weighted voting by the ensemble algorithm, we were able to reach the highest malware detection accuracy of 97.8%. We also verified that discarding the permission information lead to the improvement in terms of false positive and false negative rates.

Keywords : Android, Malware, APK, AI, Ensemble Algorithm

머신러닝 기반 안드로이드 모바일 악성 앱의 최적 특징점 선정 및 모델링 방안 제안

이 계 웅† · 오 승 택†† · 윤 영†††

요 약

모바일 운영체제 중 안드로이드의 점유율이 높아지면서 모바일 악성코드 위협은 대부분 안드로이드에서 발생하고 있다. 그러나 정상앱이나 악성앱이 진화하면서 권한 등의 단일 특징점으로 악성여부를 연구하는 방법은 유효성 문제가 발생하여 다양한 특징점 추출 및 기계학습을 통해 이를 극복하고자 한다. 본 논문에서는 APK 파일에서 구동에 필요한 다섯 종류의 특징점들을 안드로가드라는 정적분석 툴을 사용하여 학습 데이터의 특성을 추출한다. 또한 추출된 중요 특징점을 기반으로 모델링을 하는 세 가지 방법을 제시한다. 첫 번째 방법은 보안 전문가에 의해 엄선된 132가지의 특징점 조합을 바탕으로 모델링하는 것이다. 두 번째는 학습 데이터 7,000개의 앱에서 발생 빈도수가 높은 상위 99%인 8,004가지의 특징점들 중 랜덤포레스트 분류기를 이용하여 특성중요도가 가장 높은 300가지를 선정 후 모델링 하는 방법이다. 마지막 방법은 300가지의 특징점을 학습한 다수의 모델을 통합하여 하나의 가중치 투표 모델을 구성하는 방법이다. 추가적으로 오탐률 및 미탐률을 개선하기 위해 권한 정보를 모두 제외하여 특징점을 재구성하고 위와 같은 환경으로 모델링하였다. 최종적으로 가중치 투표 모델인 앙상블 알고리즘 모델을 사용하여 97.8%로 정확도가 개선되었고 오탐률은 1.9%로 성능이 개선된 것이 확인되었다.

키워드 : 안드로이드, 악성코드, APK, 인공지능, 앙상블 알고리즘

1. 서 론

글로벌 시장조사 업체 Statcounter[1)]에 따르면 2019년 5월 스마트폰 운영체제 점유율은 안드로이드가 75.34%, iOS가 22.66%, KaiOS가 0.77%로 나타났다. 또한 글로벌 보안제품 성능 평가기관 AV-TEST[2)]의 조사에 따르면 수집된 안드로이드 악성코드 샘플들이 매년 상당한 규모로 증가하고 있다. 2017년과 2018년에만 각각 총 620만개, 546만개 발생하였고 이는 2015년에 수집된 악성코드보다 2배 이상 큰 수치이다. 이렇게 사이버 공격자들은 특정 OS의 악성코드를 만드는데 집중하고 있는 것은 당연하다.

본 논문에서는 위와 같이 기하급수적으로 증가하는 안드로이드 모바일 악성코드 공격에 대비하기 위해서 2018 KISA 정보보호 R&D 데이터 챌린지 대회에서 사용된 안드로이드 악

※ 이 논문은 2016년 대한민국 교육부와 한국연구재단의 지원(NRF-2016R1D1A1B03931324), 산업통상자원부와 한국산업진흥원의 지원(P0004602)과 2019학년도 홍익대학교 학술연구진흥비에 의하여 지원을 받아 수행되었음.
※ 이 논문은 2019년도 한국정보처리학회 춘계학술발표대회에서 '머신러닝 기반 악성 안드로이드 모바일 앱의 최적특징점 선정 및 모델링 방안 제안'의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 홍익대학교 컴퓨터공학전공 학사과정
†† 정 회 원 : ㈜넷코아테크 수석연구원
††† 비 회 원 : 홍익대학교 컴퓨터공학과 조교수
Manuscript Received : July 8, 2019
Accepted : September 4, 2019
* Corresponding Author : Young Yoon(young.yoon@hongik.ac.kr)

1) http://gs.statcounter.com/os-market-share/mobile/
2) https://www.av-test.org/en/statistics/malware/

성·정상 앱[3] 13,000개를 활용하여 7,000개, 2,000개, 4,000개의 3개 그룹으로 나눈 후, 그 중 7,000개는 훈련 데이터로 사용하고, 나머지 2,000개, 4,000개를 각각 테스트 데이터로 사용한다. 이를 통해 최적의 특징점에 대한 선정과 모델링 방안에 대해서 제안한다. 특히 선행 연구[1]의 미탐률 및 오탐률을 개선하기 위하여 특징점을 재조합하는 과정을 보이고 새롭게 모델링하여 성능을 평가한다. 2장에서는 관련 연구에 대해서 살펴본다. 3장에서는 특성 추출 방법, 4장에서는 선정된 특징점들을 훈련 데이터로 만들어 모델링한다. 5장에서는 실험 결과를 분석하고 마지막으로 6장에서는 결론 및 향후 연구방향에 대해서 기술한다.

2. 관련 연구

안드로이드 시장 점유율과 개발 기술의 다양성과 복잡성이 급격하게 증가함에 따라 머신러닝 기반 악성앱의 특징점을 분석하는 연구가 진행되고 있다. Classes.dex의 실행 코드로 추출 가능한 API만으로 악성코드 탐지를 한 연구에서는 82%의 낮은 탐지율을 보였다[2]. 권한 하나만을 특징점으로 하여 정적분석을 진행한 연구는 빠른 분석 속도가 장점이나 277개라는 적은 데이터로 87%라는 다소 낮은 정확도를 보인 단점이 있었다[3]. API와 권한 간 상세한 매칭 방식을 취한 연구에서는 모델링 했던 데이터의 정상 대 악성 비율과 테스트 데이터의 정상 대 악성 비율이 3대 1로 일치하여 모델의 테스트 결과에 영향을 주었고, 미탐률이 5.2%로 측정되었다[4].

정확도를 높이기 위해 AndroidManifest.xml 파일에서 권한정보, API 호출, 인증서, 인텐트[4] 등 다양한 형태의 정보를 추출하여 99.45%라는 정확도를 얻은 기존 연구는 표본 내 검증만 이루어졌다[5]. 400개의 APK파일로 특징점을 추출하고 주성분분석(PCA)을 통해 98%의 정확도를 얻은 기존 연구 또한 표본 내 검증만 이루어져서 과적합 여부가 불투명하다[6]. 한편, 안드로이드 앱의 과도한 권한과 부족한 권한에 대해 검증을 한 연구에서는 정상앱과 악성앱 모두 무분별한 권한 정보를 사용함을 밝혔다[7]. 정상앱이지만 위험한 권한을 다수 사용하는 경우 악성앱으로 탐지되는 오탐을 야기할 수 있다.

본 논문에서는 APK파일 내의 AndroidManifest.xml과 Classes.dex 헤더 파일에서 얻을 수 있는 총 다섯 종류의 특징점을 결합하는 최적 특징점 선정하는 방법을 제안한다. 또한 권한을 특징점으로 사용한 연구[3-6]와는 달리 정상 및 악성앱의 권한 특징점을 제거한 나머지 네 종류의 특징점만을 결합하여 정확도와 오탐률, 미탐률을 개선하는 모델링 방안을 제안한다.

3. 특성 추출 방법

Fig. 1과 같이 Androguard[5] 정적 분석 도구를 이용하여 다섯 가지 유형으로 분류되는 정상앱과 악성앱의 특징점들을 다음 Table 1과 같이 두 가지 방법으로 선정하고 특징 값을 CSV 파일에 정리하였다.

3) https://www.kisis.or.kr/kisis/subIndex/382.do
4) https://en.wikipedia.org/wiki/Intent_(Android)
5) https://androguard.readthedocs.io/en/latest/#

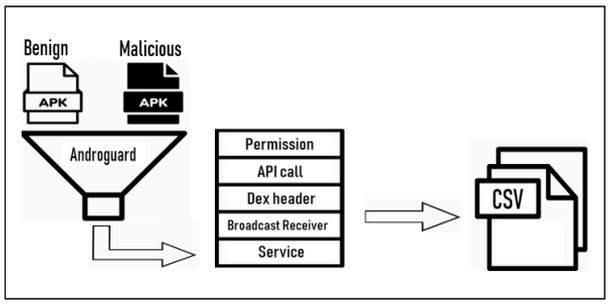

Fig. 1. Feature Extraction Based on Static Analysis Tools

Table 1. Feature Categories and Top 99% Frequent Features

ID	Features	Extracted by Human Experts	Top 99% Features in Terms of Occurrences
F1	Permission	34	295
F2	API	67	7652
F3	Dex header	21	21
F4	Broadcast Receiver	7	20
F5	Service	3	16
Total		132	8004

Permission은 앱에서 사용자 기기에 접근하여 사용자의 정보를 얻기 위한 것으로 AndroidManifest.xml에서 등록된 권한들을 특징점으로 삼았다. 예를 들어, SEND_SMS, READ_SMS, RECEIVE_SMS는 사용자 기기의 문자메세지와 관련된 권한이며, READ_EXTERNAL_STORAGE와 WRITE_EXTERNAL_STORAGE는 외부 저장소에 데이터를 읽거나 쓰는 권한이다. API Call은 Classes.dex파일을 디컴파일하여 얻어낸 자바 소스코드를 참조하였다. 또한 Dex Header는 Classes.dex 파일의 헤더영역에 존재하는 23가지 필드 값 중에서 숫자로 표시되지 않는 magic값과 해쉬값을 제외한 나머지 21가지를 특징점으로 삼았다. 21가지의 필드값으로 악성코드를 분류하였을 때, 모델의 검증과정에서 84%라는 정확도가 나왔고 이는 다른 특징점들과 결합했을 때 악성코드 분류에 도움이 될 수 있다. 또한 이 21가지의 특징점은 기존 연구들과의 가장 큰 차별점이다. 한편, 안드로이드 앱 구조는 Activity, Service, Broadcast Receiver (BR) 및 Content Provider의 네 가지 컴포넌트로 이루어져 있다. 이들 정보 역시 .xml 파일에서 얻을 수 있다. Service는 앱 UI를 닫은 후 파일 다운로드 또는 음악 재생과 같은 백그라운드 프로세스를 정의하고 Broadcast Receiver는 배터리 부족, 메일 알람과 같이 시스템이나 다른 앱에서의 수신하고 응답하기 위해 사용되는 컴포넌트이다. Content Provider는 안드로이드 앱이 공유하는 공간으로 데이터가 파일 시스템이나 다른 장소에 있더라도 앱은 Content Provider를 통해서 데이터를 접근한다.

3.1 보안 전문가에 의한 특징점 선정

보안 전문가에 의해 선정된 132가지의 특징점 조합은 총 네 가지의 논문을[2, 5, 8, 9] 취합하여 선정하였고 그 내용은 다음

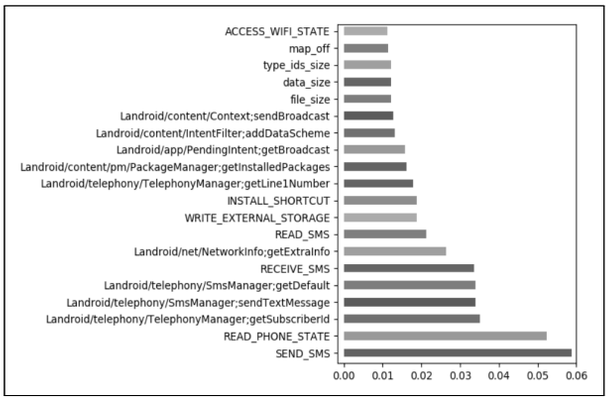

Fig 2. Top 20 Features by Importance

과 같다. 개인정보와 디바이스 동작에 잠재적으로 위험을 가지고 있는 권한들 가운데 SEND_SMS, ACCESS_WIFI_STATE, WAKE_LOCK등 총 34가지를 특징점으로 삼았다. 또한 학습데이터에서 실질적으로 위험 가능한 행위를 하는 API에 대해 빈도수가 높은 것들을 위주로 하여 총 67가지의 특징점을 엄선하였다. Dex Header는 고정적으로 21가지를 특징점으로 삼았다. Service와 BR은 악성앱 3,000개 중에서 빈도수가 상위 98%인 것을 각각 3가지, 7가지를 특징점으로 삼았다.

3.2 랜덤포레스트 분류기에 의한 특징점 선정

훈련데이터 7,000개의 앱에서 얻을 수 있는 다섯 종류의 특징점의 평균 빈도수가 상위 99%인 것을 1차 선별한다. 각 특성 집단에 따라 추출하는 기준을 달리하였는데, Permission은 7천개의 앱에서 추출되는 모든 권한 정보를 특징점으로 하였다. API는 악성앱 3,000개에서 빈도수가 상위 99%인 API와 정상앱 4,000개에서 빈도수가 상위 99%인 API를 합한 것을 특징점으로 하였다. Dex header의 특징점은 21가지이며 BR과 Service는 빈도수가 상위 99%인 것들 중 악성앱에서의 빈도수가 정상앱에서의 빈도수보다 3배 이상 큰 것들을 특징점으로 하였다. 이렇게 추출한 8,004가지 특징점들의 조합으로 Random Forest 분류기를 구성한다. 이 모델에서 특성 중요도가 높은 300가지만 선정할 수 있고, 이 300가지 특징점으로 다시 모델링을 진행한다. 아래 (Fig 2)는 300가지의 특징점 중 특성 중요도가 높은 상위 20가지를 그래프로 나타낸 것이다.

Fig. 2의 file_size, data_size, type_ids_size, map_off는 Dex header에서 얻은 특징점이다. 특성 중요도 상위 20가지는 Permission이 7가지, API는 9가지, Dex header가 4가지로 구성되었다. 300가지의 특징점을 모두 확인했을 때, Permission이 22, API는 266, Dex header가 7, Broadcast Receiver는 2, Service는 3가지로 구성된 것을 알 수 있었다.

Random Forest Classifier[11]와 같은 결정 트리 기반 모델은 특성 중요도 기능을 제공한다. 특징점의 엔트로피가 낮을수록 악성앱과 정상앱의 분류에서 특성 중요도는 증가한다. 즉, 특성 중요도 값이 높아질수록 해당 특징점은 분류에 더 많은 영향을 끼친다. 중요한 특징점들을 원하는 개수만큼 반복적으로 재조합 및 재선정하는 방식으로 최적의 학습 모델 구성을 꾀할 수 있다.

3.3 권한 특징점을 제외한 특징점 선정

특성 중요도가 높은 상위 20가지 특성에서, 사용자를 위협할 수 있는 WRITE_EXTERNAL_STORAGE, READ_PHONE_STATE와 같은 위험한 권한들은 악성 행위에 관여하기도 하지만, 정상앱에서도 높은 빈도수로 사용자에게 권한을 요청한다. 아래 Table 2는 정상앱 4,000개와 악성앱 3,000개를 합한 학습데이터 7,000개 중에서 주요 권한들에 대한 빈도수를 나타낸 것이다. INTERNET과 READ_PHONE_STATE와 같은 권한은 하나의 앱에서 AndroidManifest.xml에 동일한 권한이 다중등록 되어있는 경우가 있어 총 악성앱의 수 3,000개보다 높은 빈도수를 기록하였다. 악성앱과 정상앱 두 집단에서 모두 빈도수가 높은 권한들을 특징점들을 사용하였을 때, 미탐률과 오탐률을 높이는 원인이 될 수 있다. 이를 최소화하기 위해, 탐지에 혼동을 줄 수 있는 특징점인 권한을 모두 제거하여 API, Dex header, Broadcast Receiver, Service 총 네 가지의 특징점만을 결합하였다. 보안전문가가 선별한 132가지에서는 권한 정보 34가지를 제외하였고, 랜덤포레스트 분류기에서 선정한 300가지의 경우에서는 권한 정보 22가지를 제외하였다. 아래 Fig. 3은 권한 특징점을 제외한 특성 중요도가 높은 상위 20가지를 그래프로 나타낸 것이다. 상위 20가지는 API 17, Dex header 3가지로 구성되었고, 문자 메시지를 보내는 행위와 디바이스의 가입자 아이디를 반환하는 행위의 중요도가 가장 높았다.

Table 2. Frequency of Different Permission Settings

Permission	Frequency	
	Malicious (3,000)	Benign (4,000)
INTERNET	3381	2902
READ_PHONE_STATE	3220	1163
WRITE_EXTERNAL_STORAGE	2921	1330
ACCESS_NETWORK_STATE	2656	1908
ACCESS_WIFI_STATE	1802	501
WAKE_LOCK	1729	705

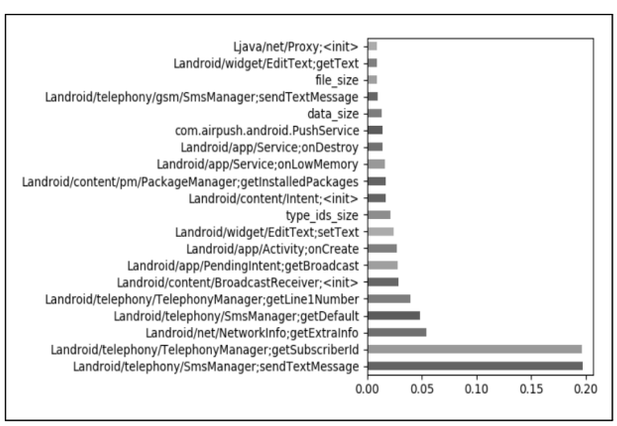

Fig. 3. Top 20 Features Excluding Permissions

4. 모델 선정

4.1 단일 알고리즘에 의한 모델링

앞서 추출한 특징점들을 바탕으로 Decision Tree Classifier [10], Random Forest Classifier [11], Gradient Boosting Classifier [12], Ada Boost Classifier (Adaptive Boosting) [13], XGBoost Classifier (Extreme Gradient Boosting) [14] 등 5가지 주요 기계학습 알고리즘들을 활용하여 모델을 학습하고 단일 알고리즘 모델별 표본 내 검증 정확도를 측정하여, 가장 정확도가 높은 단일 알고리즘 모델을 선정하였다. 모델의 하이퍼파라미터 (hyper-parameter)의 경우, n_estimators를 제외한 나머지는 기본으로 설정된 값 에도 안정적인 성능을 보였기에 3가지 모델의 n_estimators의 값에만 변화를 주었다. Random Forest Classifier의 경우, n_estimators를 실험적으로 최적인 1000으로 설정하였고 Gradient Boosting Classifier와 Ada Boost Classifier 경우에는 실험적으로 최적인 500으로 설정하였다.

Table 3. Accuracy Measurement

Algorithms	Feature Extraction	By a Human Expert	Additional Extraction of Top 99% in Terms of Occurrences			
	# of Features	132	300	600	1000	8004
DecisionTree		93.3	94.8	94.3	93.9	93.6
RandomForest		96.2	97.7	97.6	96.7	96.5
GradientBoosting		95.1	96.4	96.3	95.9	96.6
AdaBoost		94.0	95.4	95.3	94.9	94.7
XGBoost		94.2	95.7	96.5	95.9	95.1

훈련데이터 7,000개에 대해 단일 알고리즘 모델을 이용하고, 검증 비율을 30%로 했을 때 위 Table 3과 같이 Bagging 기법을 이용하여 다양한 학습데이터 집합별로 학습된 결정트리를 통합하는 Random Forest 알고리즘의 정확도(ACC)가 96.2%로 확인되었다. 보안 전문가가 엄선한 특징점 132가지 조합보다 발생 빈도수가 높은 상위 99%의 8,004가지의 특징점들 중 Random Forest의 특성 중요도로 최종 300가지 특징점을 조합하여 정상앱과 악성앱을 분류하는 방법이 상대적으로 우수한 것이 관찰되었다. 이는 보안 전문가가 자칫 간과할 수 있는 특성 중요도를 자동으로 선정함으로써 유효한 특징점들이 충분히 반영되어 정확한 분류를 위해 주효했음을 알 수 있다.

4.2 Voting Classifier[6] 앙상블 모델링

앞 장의 단일 알고리즘 활용과는 달리 다수의 모델 간 가중치 투표 방식으로 이종 학습 알고리즘을 혼용하는 앙상블 기법을 활용하였다. 특히 검증 정확도 상위 두 개의 모델을 선택하여 Voting Classifier를 구성하였다. 위의 Table 4와 같이 모델을 많이 결합할수록 정확도는 변함이 없었지만 테스트 시간은 더욱 늘어났기 때문에 학습 효율 차원에서 가중치 투표 방식 모델은 상위 두 모델만을 택하였다. 두 모델의 가중치 투표 방식은 다음과 같이 진행된다. 만약 1번 모델이 악성이라고 판단할 확률은 80%이고, 2번 모델이 악성이라고 판단할 확률은 40%라면 평균은 임계치를 상회하는 60%이므로 가중치 투표 모델은 결국 악성이라고 판단하게 된다.

Table 4. Different Combinations of Voting Classifiers

Voting Classifiers	Test Cases	Weighted Voting (300 Features)				
		Acc	Precision	Recall	FNR	FPR
Combine 2	2,000	97.5	96.2	96.9	3.1	2.2
	4,000	96.9	91.5	98.4	1.6	3.6
Combine 3	2,000	97.7	95.9	97.9	2.0	2.4
	4,000	96.5	89.7	98.5	1.5	4.4
Combine 4	2,000	97.8	96.0	97.9	2.0	2.4
	4,000	96.2	89.4	98.2	1.8	4.5
Combine 5	2,000	96.9	94.1	97.7	2.3	3.5
	4,000	95.3	86.9	97.8	2.2	5.7

4.3 권한을 제외한 특징점 모델링

Table 5. Validation Accuracy with the Exclusion of Permission Information

Algorithms	Feature Extraction	By a Human Expert	Additional Extraction of Top 99% in Terms of Occurrences
	# of Features of Excluding Permissions	98	278
DecisionTree		93.1	93.7
RandomForest		95.2	96.8
GradientBoosting		94.9	97.0
AdaBoost		93.9	95.4
XGBoost		93.8	95.2

위 Table 5와 같이 보안 전문가에 의해 선정된 132가지의 특징점 조합에서 권한 34가지를 제외한 98가지로 5가지 주요 기계학습 알고리즘들을 활용하여 모델을 학습하고 단일 알고리즘 모델별 표본 내 검증 정확도를 측정하였다. 또한, 특성 중요도가 높은 상위 300가지의 특징점에서 권한 22가지를 제외한 278가지에서도 모델별 표본 내 검증 정확도를 측정하였다. 훈련데이터 7,000개에 대해 단일 알고리즘 모델을 이용하고, 검증 비율을 30%로 했을 때 Gradient Boosting 알고리즘의 정확도(ACC)가 97.0%로 확인되었다. 추가적으로, 검증 정확도 상위 두 개의 모델인 Random Forest, Gradient Boosting를 결합하여 Voting Classifier를 구성하였고, 표본 외 테스트 데이터를 2,000개와 4,000개로 나누어 두 차례 정확도를 측정하였다.

5. 실험 결과

5.1 테스트 환경 구성

본 논문의 모델링 효과 검증은 Intel i7-7500 CPU @ 2.70GHz CPU와 8GB RAM을 갖춘 Windows 10 컴퓨터에서 진행하였다. Python 버전 3.6.6의 scikit-learn 라이브러리를

6) https://scikit-learn.org/stable/modules/ensemble.html

활용하고, Androguard는 버전 3.3.1을 사용하였다.

Table 6. Confusion Matrix

		Truth	
		Malware	Benign
Test	Malware	True Positive(TP)	False Positive(FP)
	Benign	False Negative(FN)	True Negative(TN)

Table 7. Definition of Model Performance

Accuracy	(TP+TN / TP+FP+TN+FN) x 100
Precision	(TP / TP+FP) x 100
Recall	(TP / TP+FN) x 100
False Negative Ratio(FNR)	(FN / TP+FN) x 100
False Positive Ratio(FPR)	(FP / FP+TN) x 100

5.2 검증 방법

7,000개의 훈련데이터를 가지고 학습된 모델의 성능을 검증하기 위하여 표본 외 테스트 데이터를 2,000개와 4,000개로 나누어 두 차례 정확도를 측정하였다. 테스트 데이터 6,000개중 첫 2,000개에는 정상앱 1,261개, 악성앱 739개로 구성되어있고, 나머지 4,000개에는 정상앱 2,880개, 악성앱 1,120개로 구성되었다. 두 차례 각 학습모델의 성능은 위의 Table 7과 같이 Accuracy, Precision, Recall, False Negative Ratio(FNR, 미탐률), False Positive Ratio(FPR, 오탐률) 값을 Table 6에 근거하여 측정하였다.

5.3 보안 전문가에 의한 학습 모델링 검증 결과

보안 전문가에 의해 선정된 132가지의 특징점 조합으로 만들어진 학습모델을 두 차례에 나누어 새로운 데이터로 테스트 한 결과는 Table 8과 같이 96.7%의 정확도, 4.3%의 미탐률, 2.7%의 오탐률을 기록 했다.

반면, 권한 정보를 제외한 98가지로 모델링하여 동일한 테스트를 진행한 결과는 권한 정보를 포함한 테스트 결과보다 정확도가 0.2%p만큼 낮아졌다.

Table 8. The Results of Model Testing with Test Data

Including Permission Information						
Experiment	Test	Acc	Precision	Recall	FNR	FPR
Human Expert	2000	96.7	95.4	95.7	4.3	2.7
	4000	96.0	89.9	96.7	3.3	4.2
Random Forest	2000	97.3	96.5	96.2	3.8	2.0
	4000	97.0	92.0	97.8	2.2	3.3
Voting Classifier	2000	97.5	96.2	96.9	3.1	2.2
	4000	96.9	91.5	98.4	1.6	3.6
No Permission Information						
Human Expert	2000	96.5	95.6	94.8	5.1	2.6
	4000	95.8	89.9	95.5	4.5	4.1
Gradient Boosting	2000	97.3	96.1	96.6	3.4	2.3
	4000	96.4	90.4	97.3	2.6	4.0
Voting Classifier	2000	97.8	96.8	97.3	2.7	1.9
	4000	96.9	91.9	97.7	2.3	3.3

5.4 단일 모델 분류기에 의한 학습 모델링 검증

전체 추출한 특징점 8,004가지 조합 중에 특성 중요도를 기반으로 상위 300가지를 추출하였고 이 학습모델을 두 차례에 나누어 표본 외 새 데이터로 테스트한 결과, 132가지의 특징점 조합보다 정탐률이 1.0%p 상승하였고 미탐률과 오탐률이 최대 1.1%p, 0.9%p 개선된 것을 확인하였다. 학습데이터의 표본 내 모델 검증 결과와 견주어도 큰 차이가 없는 점을 미루어 과적합의 발생을 최소화했다고 볼 수 있다.

권한 정보 22가지를 제외한 278가지로 학습을 진행한 Gradient Boosting Classifier로 동일한 테스트를 진행한 결과, 98가지의 특징점 조합보다 정확도가 0.8%p 개선되었고, 미탐률 및 오탐률이 최대 1.9%p, 0.3%p 개선된 것을 확인하였다. 그러나 권한을 포함한 300가지의 특징점으로 학습한 Random Forest Classifier에 의한 테스트 결과보다는 개선되지 않았다.

5.5 Voting Classifier에 의한 학습 모델링 검증

특성 중요도를 기반으로 상위 300가지의 특징점과 권한 정보 22가지를 제외하여 278가지의 특징점을 학습한 Voting Classifier 모두 Random Forest Classifier와 Gradient Booting Classifier로 구성되어 있다. 권한을 포함하여 학습한 모델은 테스트 결과 단일 모델과는 정확도가 최대 0.2%p 차이가 났으나, 미탐률은 2,000개와 4,000개에서 각각 0.7%p, 0.6%p 개선되었다.

권한을 제외하여 학습한 모델은 단일 모델인 Gradient Boosting Classifier보다 모든 성능 지표에서 우수한 성능을 보였고, 나아가 권한을 포함하여 학습한 Voting Classifier의 결과보다도 정확도 및 오탐률이 개선되었다.

6. 결론 및 향후 연구 방향

안드로이드 앱에서 추출 가능한 총 다섯 종류의 8,004가지 특징점들을 결합하고 결정 트리 기반 모델의 특성 중요도에 기반하여 특징점을 간소화하였다. 한편, 악성 및 정상앱에서 공통적으로 요구하고 빈도수가 높게 측정되는 권한들을 특징점으로 사용한다면, 모델의 탐지 능력 저하를 야기할 수 있다. 따라서 권한 정보를 포함한 총 다섯 종류의 특징점과 권한 정보를 제외한 네 종류의 특징점 집합으로 나누어 모델링을 진행하였다. 학습 데이터 7,000개에서의 모델 검증 시 가장 우수한 단일 알고리즘으로 Random Forest Classifier와 Gradient Boosting Classifier가 선정되어 가중치를 이용한 투표모델은 이들의 조합으로 이루어졌다.

학습 데이터로 사용하지 않은 새로운 테스트 데이터를 2,000개와 4,000개로 나누어 모델을 검증하였다. 최종적으로 특징점들의 결합과 무관하게 단일 모델보다는 가중치 투표 모델인 앙상블 알고리즘 모델을 사용하는 것이 가장 효과적이었다. 세부적으로 정확도와 오탐률 측면에서 가장 우수한 성능을 보인 모델은 권한 정보를 포함하지 않고 특징점 278가지를 학습한 가중치 투표 모델이다. 또한 미탐률 기준으로는, 권한 정보를 포함하여 특징점 300가지를 학습한 가중치 투표 모델이다.

최근 각광받는 심층인공신경망 기반의 딥러닝의 활용도 고려하였으나, 2018 KISA 보안 챌린지 PE 파일 악성 판별

부문 상위 입상을 하면서 직접 경험한 바에 의하면, 특징점 수 대비 신경망 파라미터의 수가 과도한 경우 오히려 정확도가 낮아지는 현상을 발견하여, 딥러닝이 반드시 만능 해결책은 아님을 확인하였다. 오히려 앙상블 기법에 기반한 경량화된 기계학습 알고리즘이 효과적이었고, 이 효과는 안드로이드 악성 앱 판별 문제에서도 일관적이었다. 그러나, 타 분야 딥러닝의 성공 사례를 봤을 때, 모바일 악성코드 판별에 최적화된 새로운 형태의 신경망과 딥러닝 알고리즘의 개발도 흥미로운 향후 연구 방향이 될 수 있다고 생각한다.

References

[1] K. W. Lee, S. T. Oh, and Y. Yoon, "Modeling and Selecting Optimal Features for Machine Learning Based Detections of Android Malwares," *The KIPS Spring Conference*, Vol.26, No.1, pp.164-167, 2019.

[2] S. W. Min, H. J. Cho, J. S. Shin, and J. C. Ryou, "Android Malware Analysis and Detection Method Using Machine Learning," *Journal of KIISE : Computing Practices and Letters*, Vol.19, No.2, pp.95-99, 2013.

[3] H. L. Lee, S. H. Jang, and J. W. Yoon, "Efficient Malware Detector for Android Devices," *Journal of The Korea Institute of Information Security & Cryptology*, Vol.24, No.4, pp.617-624, 2014.

[4] S. E. Kang, N. V. Long, and S. H. Jung, "Android Malware Detection Using Permission-Based Machine Learning Approach," *Journal of the Korea Institute of Information Security & Cryptology*, Vol.28, No.3, pp.617-623, 2018.

[5] J. W. Jang, H. J. Kang, J. Y. Woo, A. Mohaisen, and H. K. Kim, "Andro-AutoPsy: Anti-malware System Based on Similarity Matching of Malware and Malware Creator-centric Information," *Digital Investigation*, Vol.14, pp.17-35, 2015.

[6] D. W. Kim, K. G. Na, M. M. Han, M. J. Kim, W. Go, and J. H. Park, "Malware Application Classification based on Feature Extraction and Machine Learning for Malicious Behavior Analysis in Android Platform," *Journal of Internet Computing and Services*, Vol.19, No.1, pp.27-35, 2018.

[7] S. W. Shin, "A Static Analysis of Android Application Permission Requirement," *Masters thesis, Korea Aerospace University*, 2016.

[8] H. A. Alatwi, "Android Malware Detection using Category-based Machine Learning Classifiers," *Masters Thesis*, Rochester Institute of Technology, NY, USA, 2016.

[9] A. Pekta, M. Cavdar, and T. Acarman, "Android Malware Classification by Applying Online Machine Learning," Computer and Information Sciences 31st International Symposium, ISCIS 2016, Kraków, Poland, pp.72-80, 2016.

[10] Dr. Nancy and Dr. Deepak Sharma, "Android Malware Detection using Decision Trees and Network Traffic," *International Journal of Computer Science and Information Technologies*, Vol.7, No.4, pp.1970-1974, 2016.

[11] L. J. Dong, L. I. Xi-Bing, and K. Peng, "Prediction of Rockburst Classification using Random Forest," *Transactions of Nonferrous Metals Society of China*, Vol.23, No.2, pp.472-477, 2013.

[12] Gadyatskaya, Olga, A. L. Lezza, and Y. Zhauniarovich, "Evaluation of Resource-based App Repackaging Detection in Android," *In Nordic Conference on Secure IT Systems*, pp.135-151. Springer, Cham, 2016.

[13] M. Qin, J. Qiu, Z. Lu, L. Chen, and W. Zhao, "AdaBoost-based Class Imbalance Learning Algorithm," *Application Research of Computers*, Vol.34, No.11, pp.1-8, 2017.

[14] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp.785-794, 2016.

이 계 웅

https://orcid.org/0000-0002-3395-5546

e-mail : 0108kw@naver.com

2013년~현 재 홍익대학교 컴퓨터공학전공 학사과정

관심분야 : 빅데이터, 인공지능, 보안

오 승 택

https://orcid.org/0000-0003-4529-2388

e-mail : eentost@gmail.com

2008년 단국대학교 경영학부 경영정보학전공(학사)

2013년 한국과학기술원 위촉연구원

2014년~2018년 ㈜빛스캔 R&D 팀장

2018년~2019년 ㈜너울리 수석연구원

2019년~현 재 ㈜넷코아테크 수석연구원

관심분야 : 악성코드 분석, 인공지능

윤 영

https://orcid.org/0000-0002-5249-2823

e-mail : young.yoon@hongik.ac.kr

2003년 University of Texas at Austin, 컴퓨터과학과(학사)

2006년 University of Texas at Austin, 컴퓨터과학과(석사)

2013년 University of Toronto, 컴퓨터공학과(박사)

2013년~2015년 삼성전자 소프트웨어센터 책임연구원

2015년~현 재 홍익대학교 컴퓨터공학과 조교수

관심분야 : 분산 시스템, 미들웨어, 인공지능, 빅데이터, 보안