계층적 클러스터링 하이브리드 모델의 활용을 통한 지속가능한 안드로이드 악성 앱 탐지 기법

이승민, 안석현, 조성제, 김동재, 황영섭

WDSC 2024

2024.08.20

단국대학교 DANKOOK UNIVERSITY

Computer Security & OS LAB

INDEX

01 서론 & 관련 연구

서론

❖ 안드로이드 악성 앱 시장

- Kaspersky - 2023년도 3,380만건의 악성 앱 차단
- Google – 2023년도 228만건의 악성 앱 차단

❖ 악성 앱 패턴 변화

- Android version upgrade
- API 정책 변화 → 기존 모델의 탐지 성능 저하
- 탐지 회피 기술의 발전

❖ 기존 모델의 성능 저하 이슈

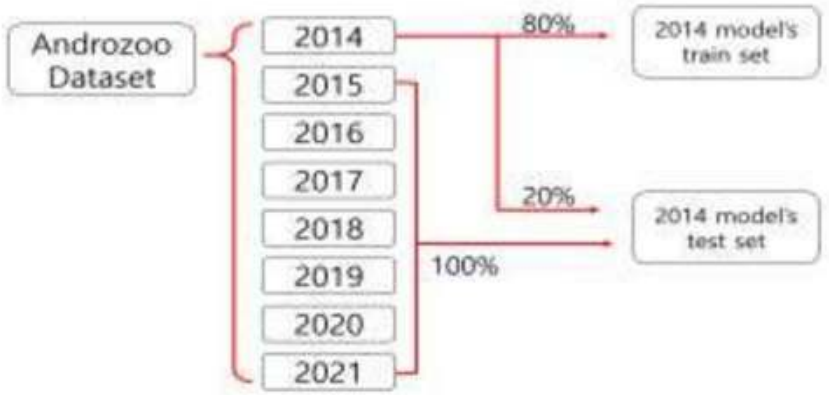
- 악성 앱 변화를 대응할 수 있는 방안이 필요 -> 계층적 클러스터링 기반의 하이브리드 모델

관련 연구

[1] 이호준, 조성제, 한상철, 조우상, 정지헌, 황영섭, 어호웅, “API 콜을 이용한 머신러닝 기반 안드로이드 멀웨어 탐지의 지속가능성 분석 연구”, 2022 한국차세대컴퓨팅학회 학술대회, pp. 298-301, 2022. 05.

❖ 데이터 셋 & 모델

- 2014 ~ 2021 dataset
- 특징정보: API call
- 학습 모델: RF(Random Forest)
- 학습 방법: 각 년도 기반의 모델 생성(총 8개의 모델)



❖ 분석

- 평가 지표: 정확도, F1-score
- 동일 년도의 테스트 셋에서는 97%이상의 정확도, 2014 ~ 2018의 각 모델들이 2019년 이후의 데이터에 정확도가 크게 하락

❖ 한계점

- 성능 하락 원인 분석 -> 각 모델의 feature importance 분석 -> feature importance의 변화가 원인

관련 연구

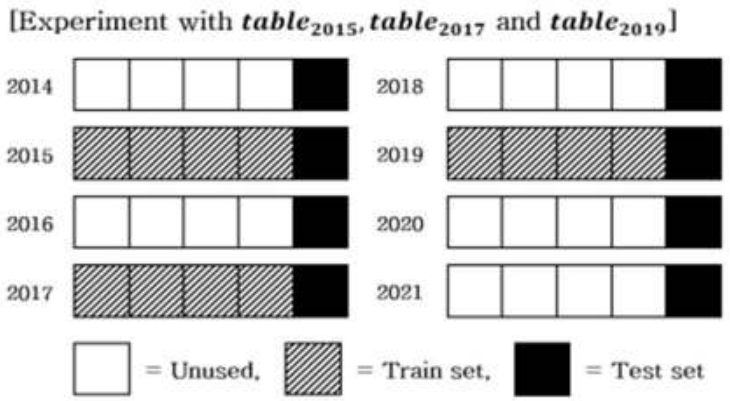
[2] 조우상, 조성제, 황영섭, 한상철, “연도별 데이터 조합 기계학습을 통한 안드로이드 악성 앱 탐지의 지속 가능성 향상 연구”, 한국차세대컴퓨팅학회 논문지, 18, 5, pp. 47-57, 2022.

❖ 데이터 셋 & 모델

- 2014 ~ 2021 dataset
- 특징정보: API call, Permission
- 학습 모델: RF(Random Forest), XGBoost(XG), LightGBM(LGBM), Neural Network(NN)
- 학습 방법: 2~ 3개의 연도별 데이터의 조합으로 학습

❖ 분석

- 평가 지표: 정확도, AUT(Area Under Time)
- 결과: LGBM(2015,2017,2020) 으로 학습시에 정확도 96.8%, AUT 96.9%



관련 연구

[3] H. Lee, S. -j. Cho, H. Han, W. Cho and K. Suh, "Enhancing Sustainability in Machine Learning-based Android Malware Detection using API calls", IEEE Fifth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 131-134, 2022.

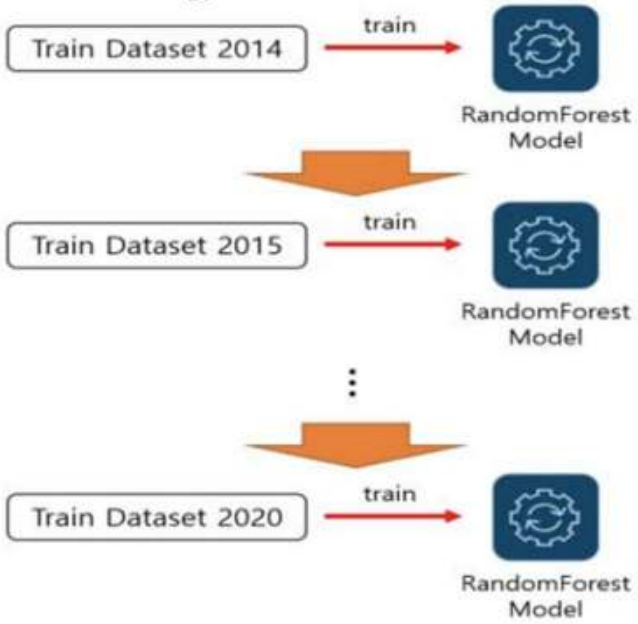
❖ 데이터 셋 & 모델

- 2014 ~ 2020 dataset
- 특징정보: API call
- 학습 모델: RF(2014~2015), TML(Traditional Machine Learning), TKIL(Tree-Keeping Incremental Learning)
- 학습 방법: 전체 연도를 모두 학습시킨 TML과 온라인 학습 기법인 TKIL

❖ 분석

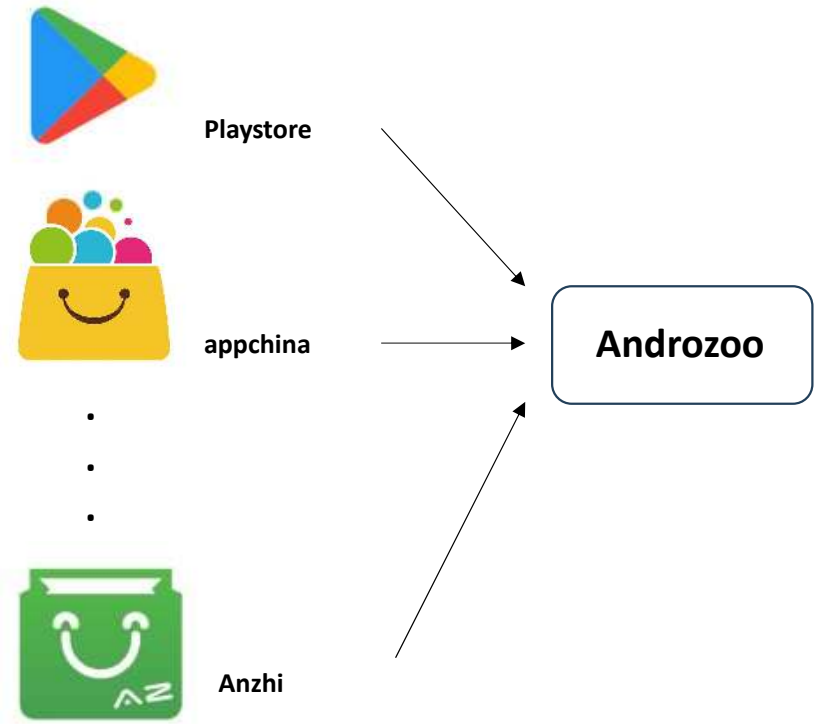
- 평가 지표: 정확도, F1-score, AUT
- 결과:

Metric	RF(2014~2015)	TML	TKIL
정확도	77.8%	97.8%	84.1%
F1-score	80%	98.7%	88.2%
AUT(F1-score)	82.7%	98.7%	89.5%



02 데이터 셋

데이터 셋



Year	Benign	Malware	Total
2014	1,500	1,500	3,000
2015	1,500	1,500	3,000
2016	1,500	1,500	3,000
2017	1,500	1,500	3,000
2018	500	500	1,000
2019	500	500	1,000
2020	500	500	1,000
2021	500	500	1,000
Total	8,000	8,000	16,000

데이터 전처리

❖ 악성, 정상 앱 선정

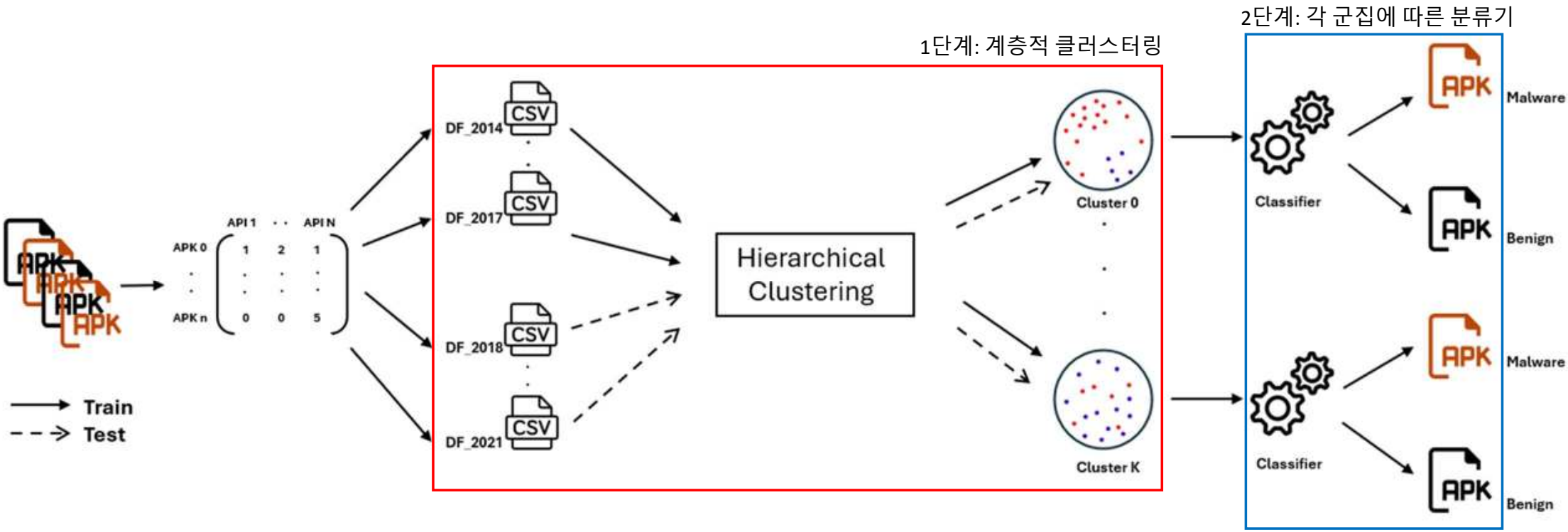
- Androzoo metadata
  - vt_detection ≥ 3 -> 악성
  - vt_detection = 0 -> 정상

❖ 데이터 전처리

- dexdump -> API call 호출 횟수 추출
- API 선정
  - D. Arp, M. Spreitzenbarth, M. Hubner, H. Gascon, K. Rieck, and C. E. R. T. Siemens, "Drebin: Effective and explainable detection of android malware in your pocket." Ndss. Vol. 14, pp. 23-26, 2014
  - Y. Aafer, W. Du, and H. Yin, "Droidapiminer: Mining api-level features for robust malware detection in android." International conference on security and privacy in communication systems. Springer, Cham, pp. 86-103, 2013
  - 1,848개의 API 선정

03 악성 앱 탐지를 위한 하이브리드 모델

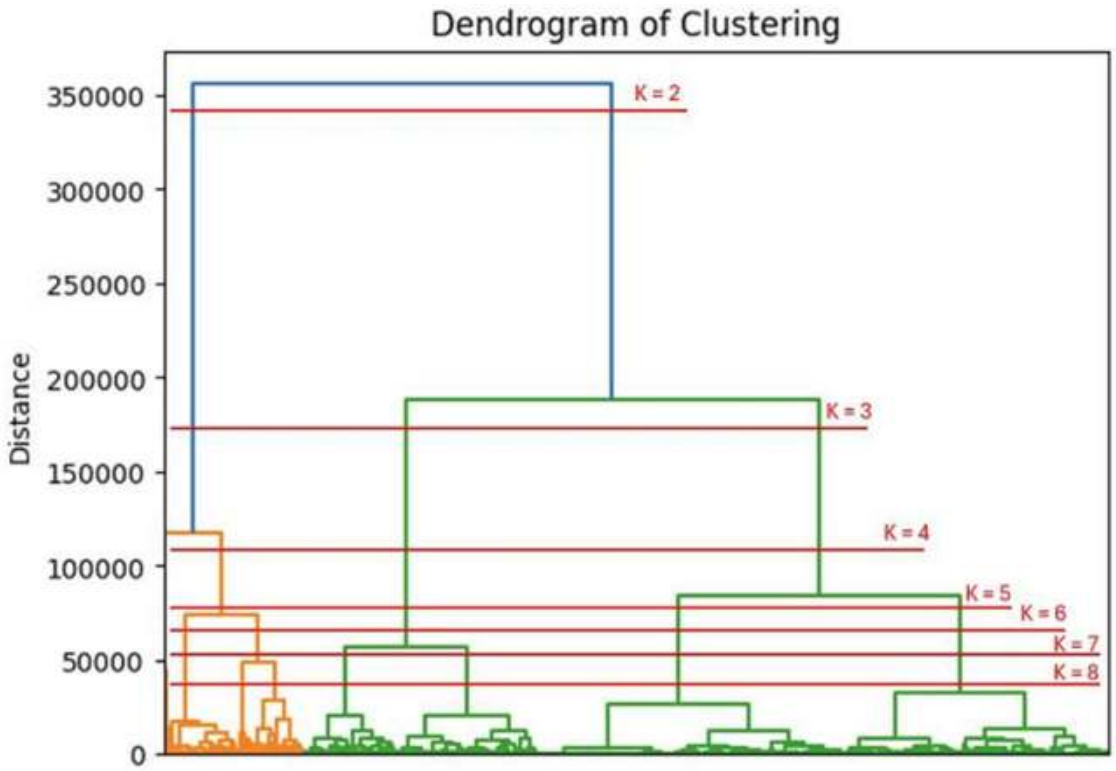
Flowchart


1단계: 계층적 클러스터링
2단계: 각 군집에 따른 분류기
APK
APK
APK
APK
API 1
API N
APK 0
APK n
1 2 1
0 0 5
CSV
DF_2014
CSV
DF_2017
CSV
DF_2018
CSV
DF_2021
Hierarchical Clustering
Cluster 0
Cluster K
Classifier
Classifier
APK
APK
APK
APK
Malware
Benign
Malware
Benign
Train
Test

계층적 클러스터링

❖ Hierarchical clustering

- 사전의 군집의 개수 미 선정
- Hyperparameter
  - 유사도 측정
    - Euclidean distance
    - Manhattan distance
    - Max distance
  - Linkage
    - ward
    - complete
    - single



❖ K(군집의 개수) = 2, 3, 4, 5, 6, 7, 8

Silhouette score

❖ Silhouette score

- 0 ~ 1사이의 값, 1에 가까울수록 밀집화와 다른 군집과 구분되어 있음
- 가장 높았던 순으로 2, 3, 4, 5를 선정

❖ 2단계 모델

- 선정된 K에 따라 RF, LightGBM(LGBM), AdaBoost 분류기에 적용

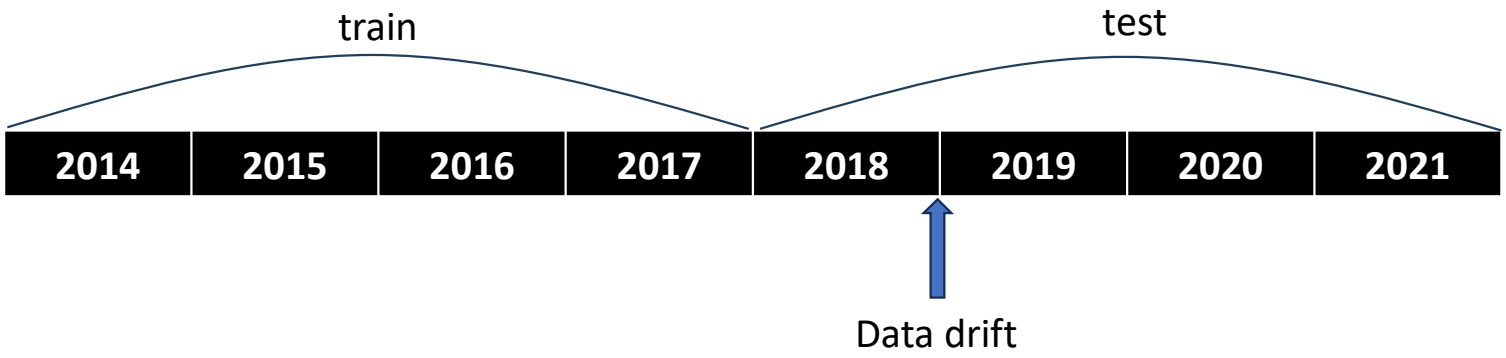
K	2	3	4	5	6	7	8
S	0.663	0.558	0.556	0.496	0.467	0.462	0.464

실험 설계

❖ F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder and L. Cavallaro, "{TESSERACT}: Eliminating experimental bias in malware classification across space and time," 28th USENIX Security Symposium, pp. 729-746, 2019.

- Temporal bias

❖ 선행 연구에서 2019년도에서 data drift가 관측 -> 새로운 유형의 악성 앱의 출현



04 실험 결과 및 논의

데이터 불균형

Cluster	Benign	Malware	Total
1	31	56	87
2	706	988	1,694
3	2,034	902	2,936
4	1,243	2,722	3,965
5	1,986	1,332	3,318

❖ K = 5일때 데이터 불균형 관측

❖ Chicco, D., Jurman, G. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". BMC Genomics, 21, 5, 2020

❖ 데이터 불균형 발생시 정확도는 평가지표로 적합하지 않음 -> F1-score 사용

평가지표

❖ Confusion matrix

- TP: 악성을 악성으로 예측
- TN: 정상을 정상으로 예측
- FP: 정상을 악성으로 예측
- FN: 악성을 정상으로 예측

$$Precision = \frac{TP}{TP + FP} \qquad Recall = \frac{TP}{TP + FN}$$

$$F1 - score = \frac{2 * Precision * recall}{Precision + recall}$$

❖ M. Sokolova, G. Lapalme, “A systematic analysis of performance measures for classification tasks”, Information Processing & Management. pp. 427-437, 2009

$$Recall_{\mu} = \frac{\sum_{i=1}^{i} TP_i}{\sum_{i=1}^{i} (TP_i + FN_i)}$$

❖ 데이터 불균형 발생시 개체 수가 적은 클러스터의 성능이 전체 모델 평가의 과도한 영향 가능성
이에, Micro-Average 사용

$$Precision_{\mu} = \frac{\sum_{i=1}^{i} TP_i}{\sum_{i=1}^{i} (TP_i + FP_i)}$$

$$F1 - score_{\mu} = \frac{(\beta^2 + 1) Precision_{\mu} Recall_{\mu}}{\beta^2 Precision_{\mu} + Recall_{\mu}}$$

평가지표

$$AUT^{-}(f,N)=\frac{1}{N-1}\sum_{k=1}^{N-1}\frac{[f(x_{k+1})+f(x_k)]}{2}$$

❖ 지속가능성 평가 지표

❖ $f(x)=$ 성능지표

❖ N = 테스트 데이터 셋(DF_{year}) 개수 = (2018~2021)

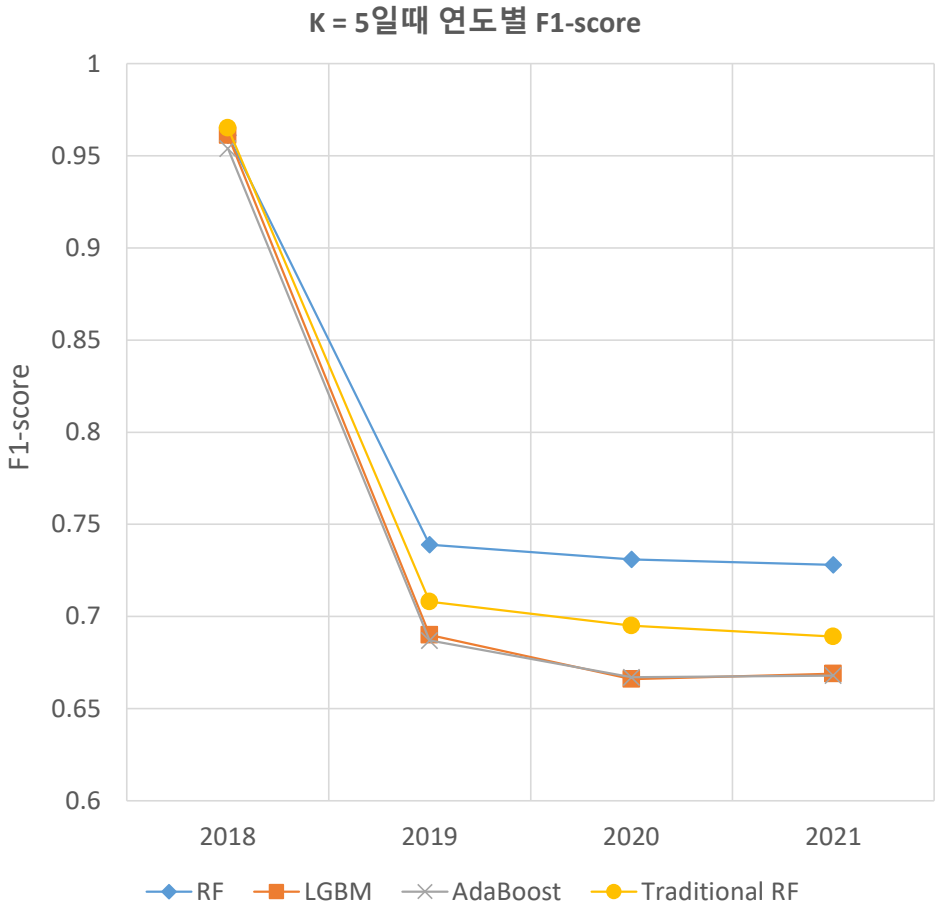
$AUT^{-}(F1-score,4)$

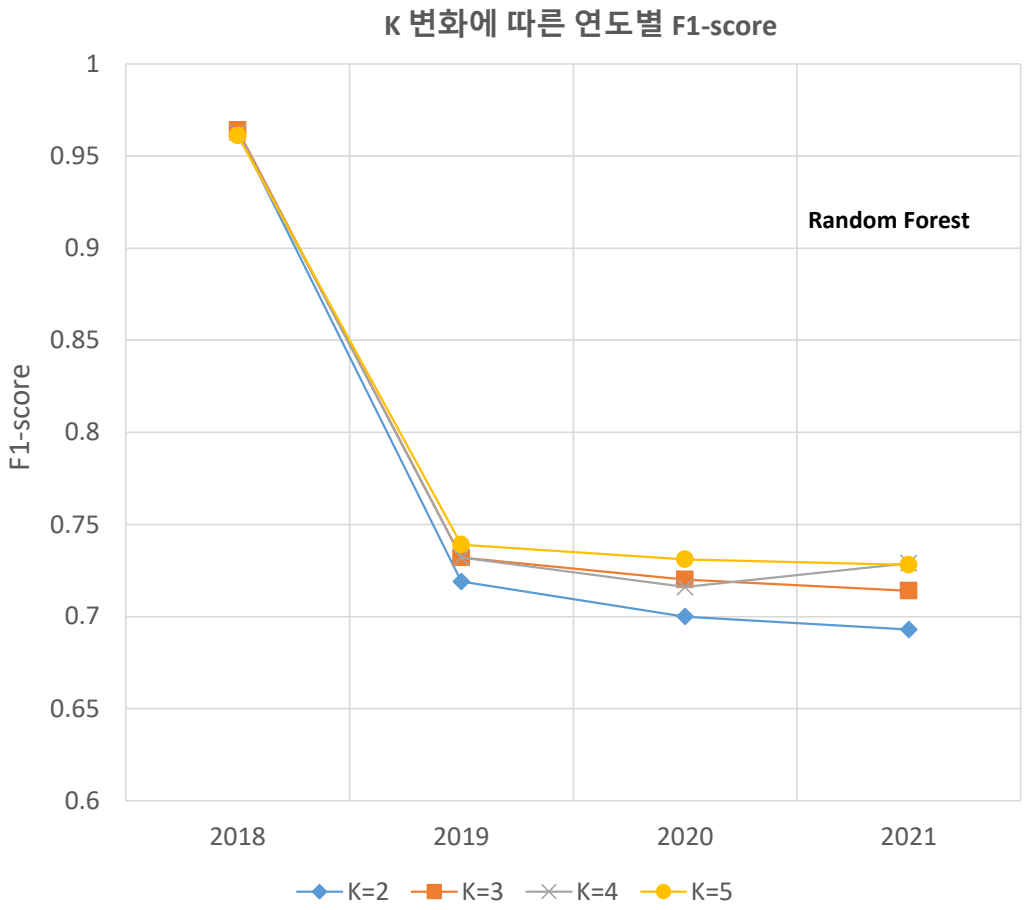
❖ F. Pendlebury, F. Pierazzi, R. Jordaney, J. Kinder and L. Cavallaro, "{TESSERACT}: Eliminating experimental bias in malware classification across space and time," 28th USENIX Security Symposium, pp. 729-746, 2019.

실험결과

클러스터 개수(K)	RF		LGBM		AdaBoost	
	F1-score	AUT^{-}	F1-score	AUT^{-}	F1-score	AUT^{-}
K=2	0.754	0.749	0.727	0.722	**0.725**	**0.72**
K=3	0.77	0.763	0.727	0.722	0.725	0.721
K=4	0.773	0.764	0.727	0.722	0.725	0.721
K=5	**0.779**	**0.771**	0.728	0.723	0.726	0.721
평균	0.769	0.761	0.727	0.722	0.725	0.72

실험 결과 분석


K = 5일때 연도별 F1-score
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
F1-score
2018
2019
2020
2021
RF
LGBM
AdaBoost
Traditional RF


K 변화에 따른 연도별 F1-score
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
F1-score
Random Forest
2018
2019
2020
2021
K=2
K=3
K=4
K=5

실험 결과 분석

Model	RF		Hybrid(K=5)	
	F1-score	AUT^{-}	F1-score	AUT^{-}
Score	0.749	0.743	0.779	0.771

❖ Confusion Matrix 분석 결과

		Predict	
		Positive	Negative
Actual	Positive	1,963	37
	Negative	1,074	926

❖ True Positive Rate(TPR): 98.1%

❖ False Positive Rate(FPR): 53.7%

Discussion

1 각 클러스터 별 데이터 불균형 문제

- Over sampling: SMOTE(Synthetic Minority Over-sampling Technique)
  - 과적합 발생 가능

2 FPR 비율 조정

- 임계값(threshold), 가중치 조정
  - FPR조정 시에 TPR 감소 가능성

05 결론 및 향후 연구

결론 및 향후 연구

❖ 결론

- 성능이 가장 좋은 조합
  - 군집의 개수 5개, 분류기 모델 RF
  - 전통적인 방식 대비 약 3%p의 성능 개선

❖ 향후 연구

- FPR의 비율 감소
  - 임계값, 가중치 설정
- 데이터 불균형
  - over sampling 적용

Acknowledgement

이 논문은 산업통상자원부(MOTIE)와 한국에너지기술평가원(KETEP)의
지원을 받아 수행한 연구과제임.(No. 20212020800120)
또한,
이 연구는 2021년도 정부(과학기술정보통신부)의 재원으로
한국연구재단의 지원을 받아 수행된 기초연구사업임.(no. 2021R1A2C2012574)

Thank you

Q&A

Errata Slip #6

Proceedings of the 28th USENIX Security Symposium

For the paper "TESSERACT: Eliminating Experimental Bias in Malware Classification across Space and Time" by Feargus Pendlebury, Fabio Pierazzi, and Roberto Jordaney, King's College London & Royal Holloway, University of London; Johannes Kinder, Bundeswehr University Munich; Lorenzo Cavallaro, King's College London (Thursday session, "Mobile Security 2," p. 735 of the Proceedings) the authors have provided a fix to a typo in Equation (4).

The published version reported an extra normalization factor of (1/N); the correct version of Equation (4) is:

$$AUT(f, N) = \frac{1}{N-1} \sum_{k=1}^{N-1} \frac{[f(x_{k+1}) + f(x_k)]}{2}$$

The results in the paper and the released implementation of the TESSERACT framework remain valid and correct as they rely on Python's numpy implementation of area under the curve.

The authors would like to thank Xiaohan Zhang from Fudan University for identifying this typo.