정보보호 R&D 데이터 챌린지

고려대 이재융, 연세대 박해주

모바일 악성앱 탐지

문제

제공된 대용량의 안드로이드 앱 데이터셋을 기반으로 아래 사항을 탐지 및 분류할 수 있는 알고리즘 및 프로그램을 제시하시기 바랍니다

- 정상 앱과 악성 앱 구분
- 악성 앱에 한하여 유형(패밀리) 분류

데이터

준비된 데이터 셋은 다음과 같습니다:

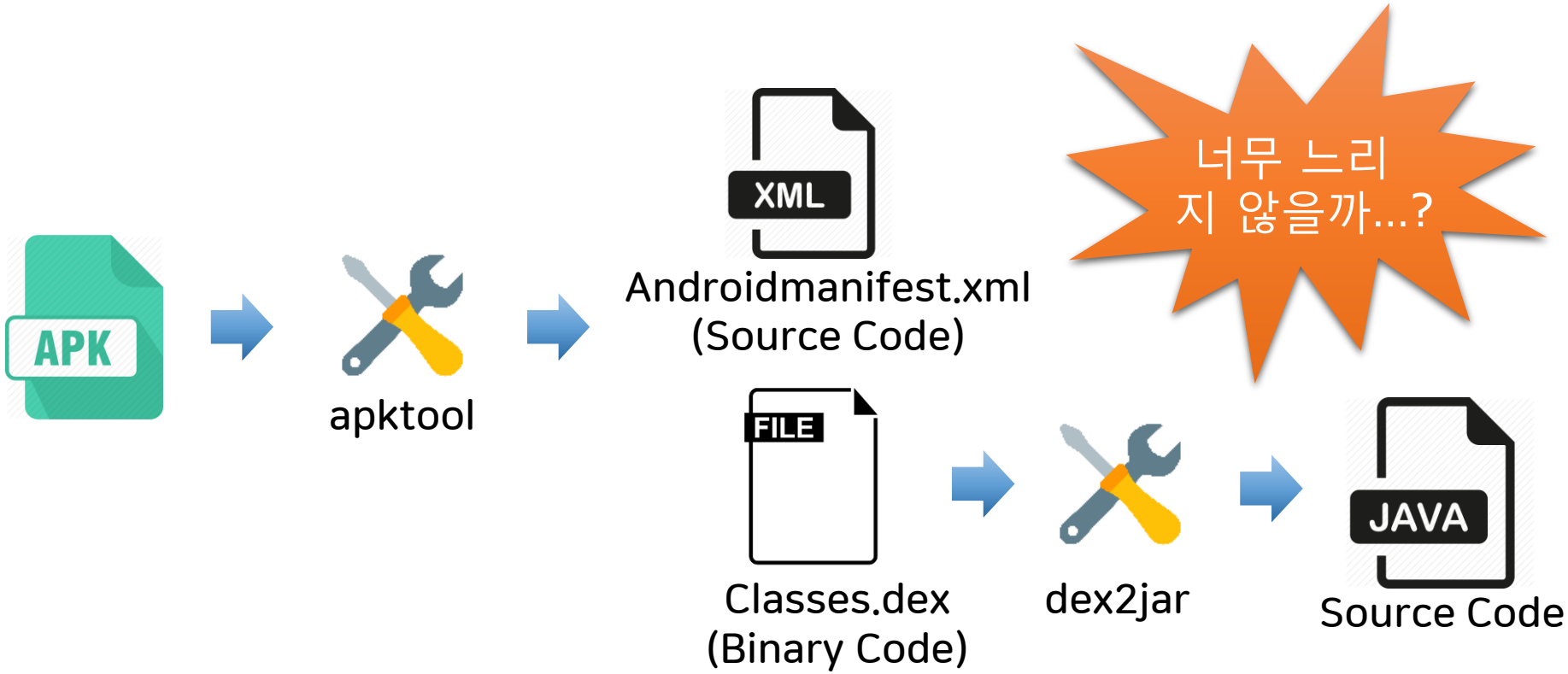
- KU-CISC2017-AutoPsy-1st(학습용, 테스트용)
  - 정상 앱 1500개, 악성 앱 500개 + 정답지 (악성 앱 여부 및 패밀리명)
- KU-CISC2017-AutoPsy-2nd(제출용)
  - 정상 앱 1500개, 악성 앱 500개
- KU-CISC2017-AutoPsy-3rd,4th(본선용)
  - 악성 앱 60개 + 90개

패밀리
AdWo
Boxer
Dowgin
AirPush
Gappusin
SMStado
Counterclank
Wapsx
OpFake
SMSAgent
본선1차(+2종)
본선2차(+3종)

특징선택

Androidmanifest.xml	Source Code
Permission	API

분석?



데이터파싱

```
20db 1e76 2066 1976 8059 1976 b04f 1976
2034 1d76 801e 1976 404f 1976 405a 1976
205a 1976 706b 1976 a05a 1976 3043 1976
605a 1976 a04a 1976 80db 1e76 407d 1d76
```

PE 파일과 비슷한 구조

```
hdr['string_ids_size'] = string_ids_size
hdr['string_ids_off' ] = string_ids_off
hdr['type_ids_size' ] = type_ids_size
hdr['type_ids_off' ] = type_ids_off
hdr['method_ids_size'] = method_ids_size
hdr['method_ids_off' ] = method_ids_off
```

자체제작한 자동화 프로그램

APK

Unzip

XML

Androidmanifest.xml
(Binary)

FILE

Classes.de
x
(Binary)

APK_Parse Module

Parse_dex.py

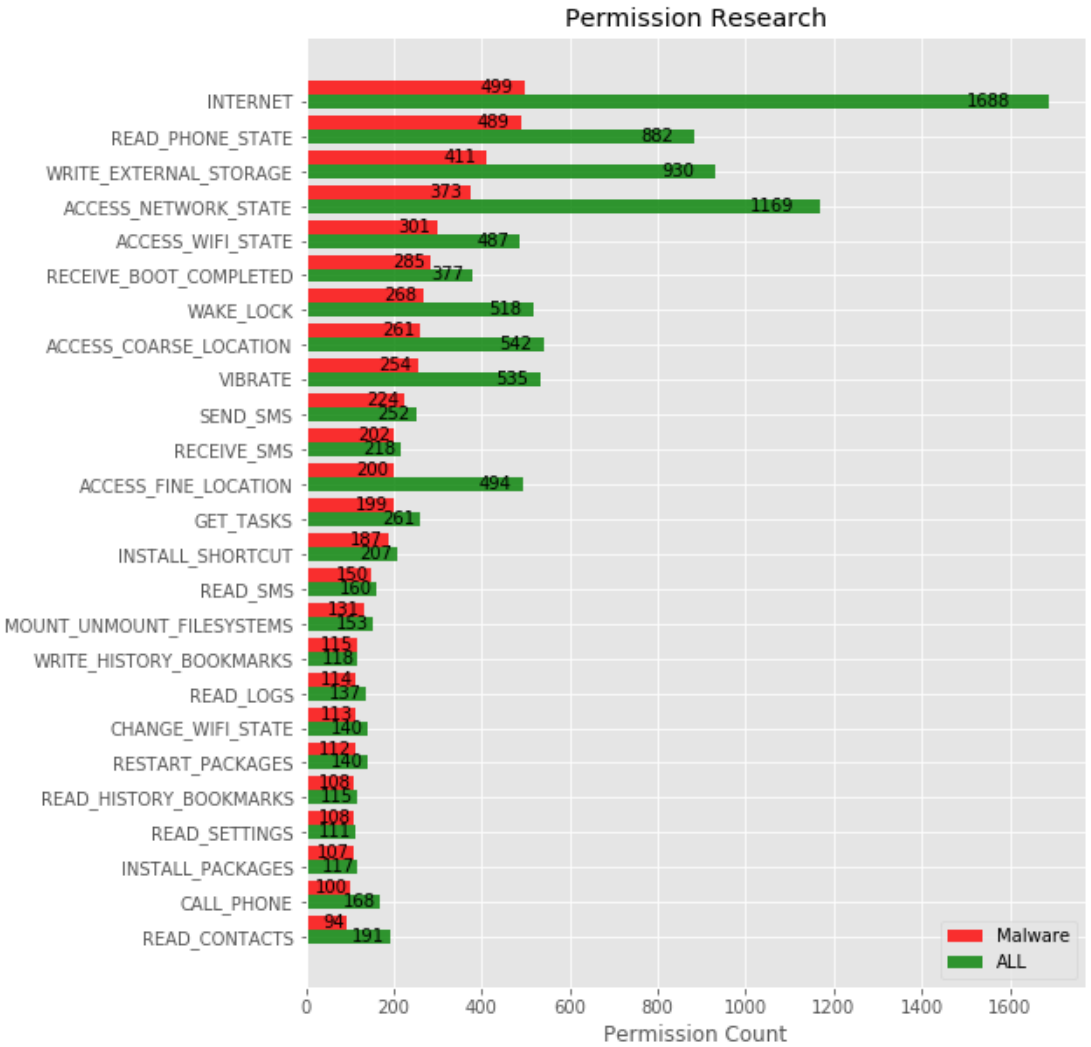
Permission 정보

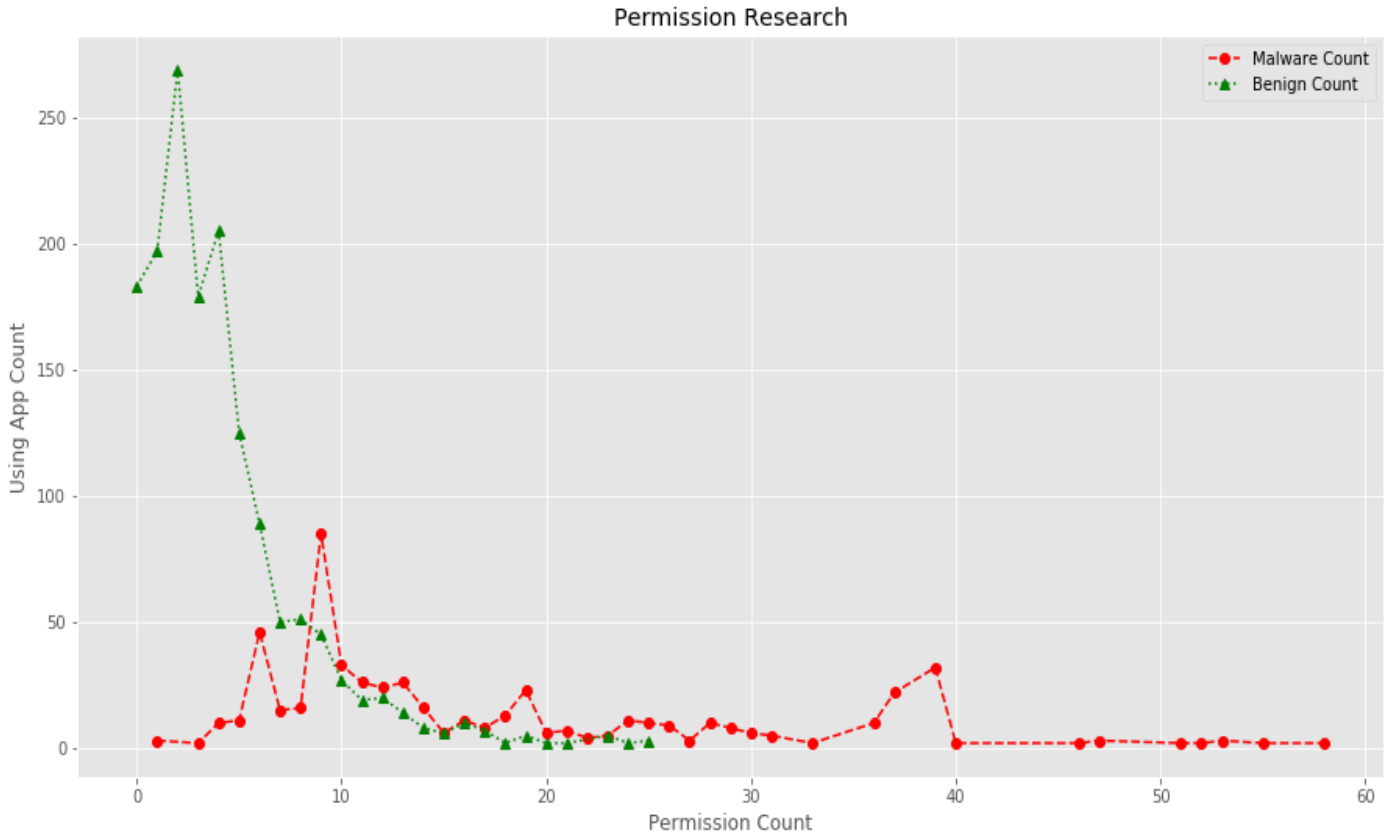
API

API 정보

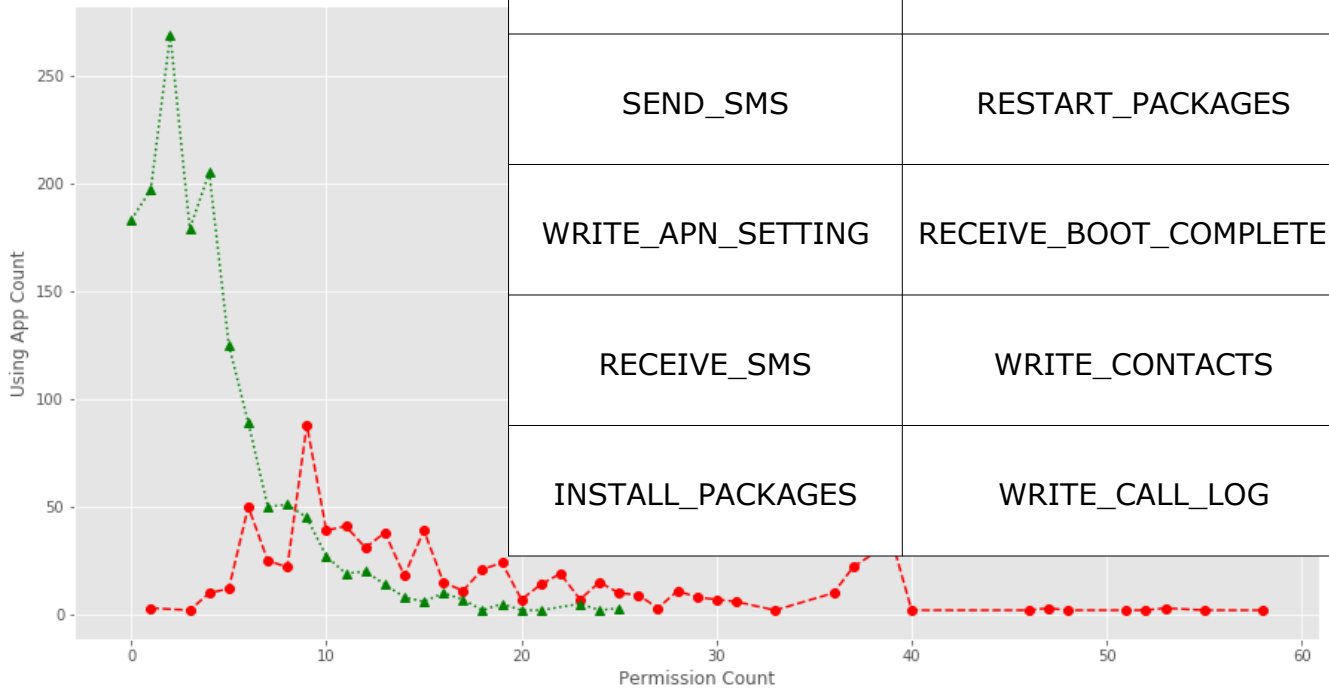
분석

- API분석은 다음 논문을 참조하였습니다
  - Andro-AutoPsy Anti-malware system based on similarity matching of malware and malware creator-centric information
  - Detecting and classifying method based
  - Recognizing API Features for Malware Detecting Using Static Analysis
  - Human Detection And Recognition

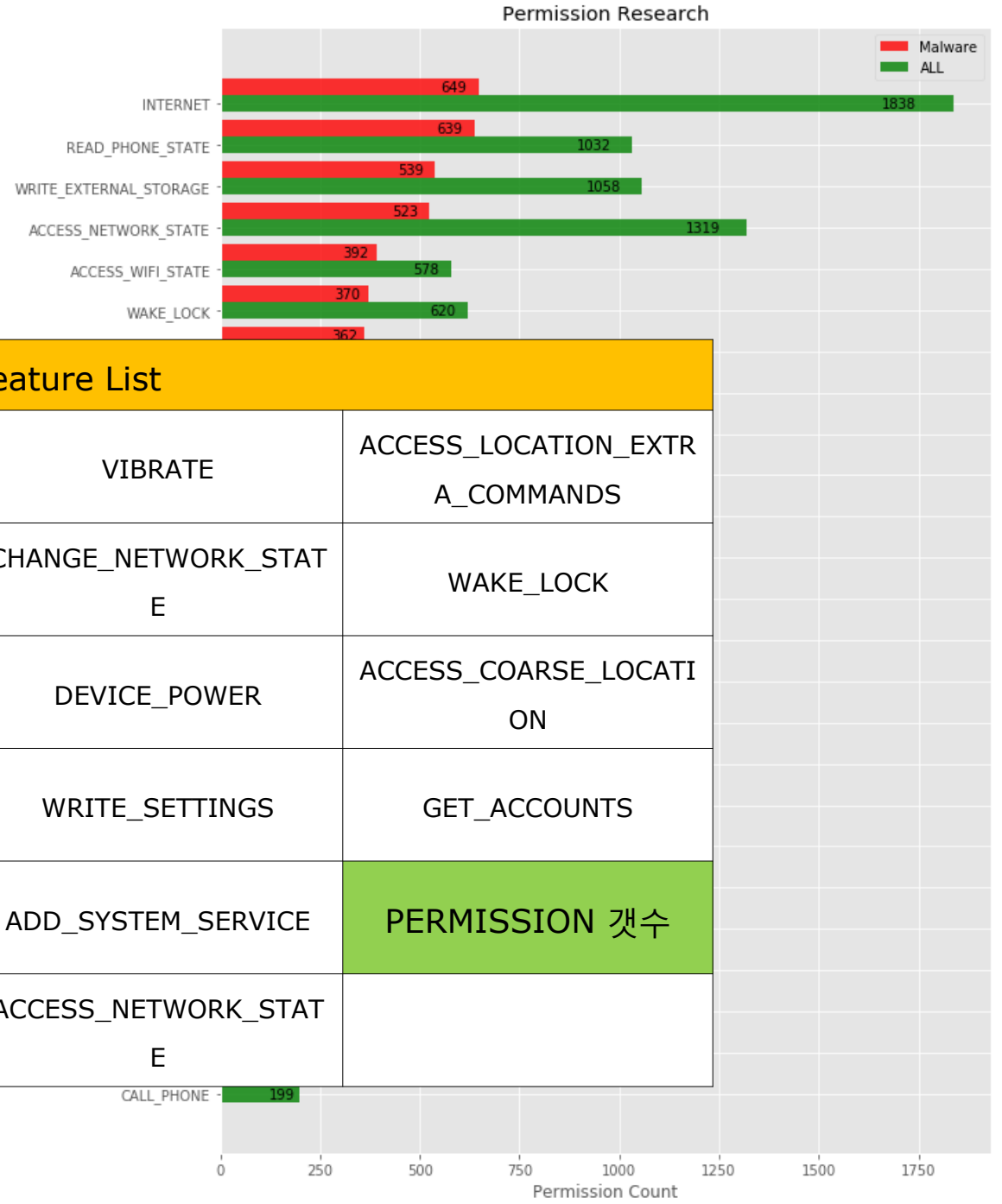




분석

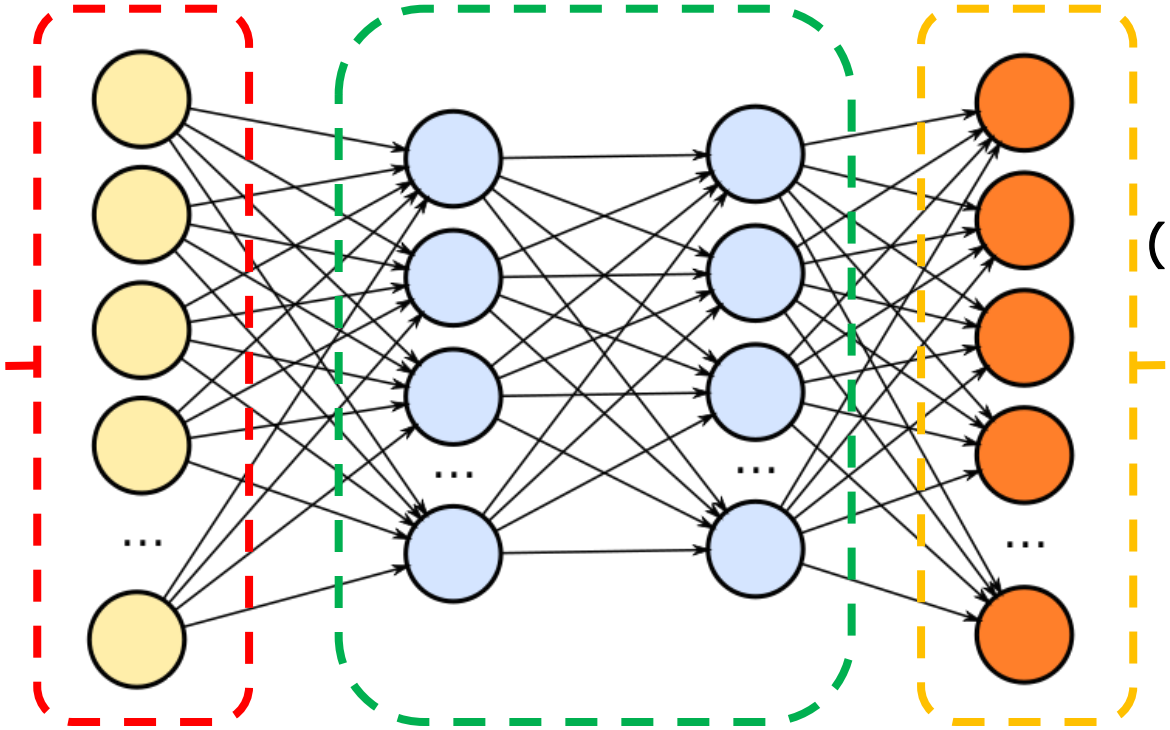


Permission Feature List			
READ_SMS	READ_PHONE_STATE	VIBRATE	ACCESS_LOCATION_EXTRA_COMMANDS
WRITE_SMS	READ_EXTERNAL_STORAGE	CHANGE_NETWORK_STATE	WAKE_LOCK
SEND_SMS	RESTART_PACKAGES	DEVICE_POWER	ACCESS_COARSE_LOCATION
WRITE_APN_SETTING	RECEIVE_BOOT_COMPLETED	WRITE_SETTINGS	GET_ACCOUNTS
RECEIVE_SMS	WRITE_CONTACTS	ADD_SYSTEM_SERVICE	PERMISSION 갯수
INSTALL_PACKAGES	WRITE_CALL_LOG	ACCESS_NETWORK_STATE	



DNN

Deep Neural Network



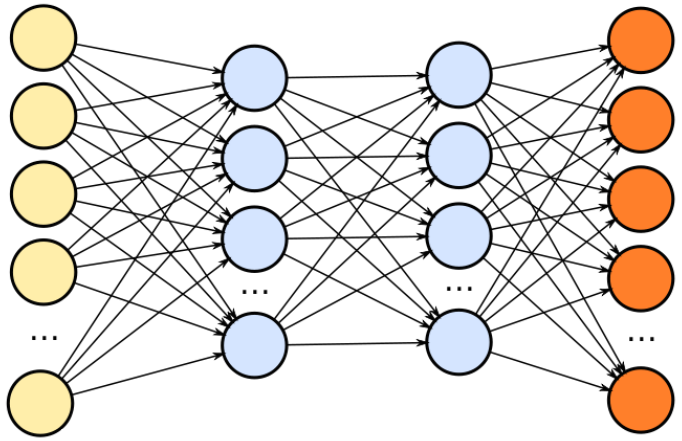
입력층 노드 Feature수
101개

출력층 노드 11+n개
(1개 정상, 10+n개 악성패밀리

2계층 Hidden Layer(ReLU)
매계층에 200개씩 노드배치
ReLU: Rectified Linear Unit
Overfitting을 막기 위해 L2 Regularization적용

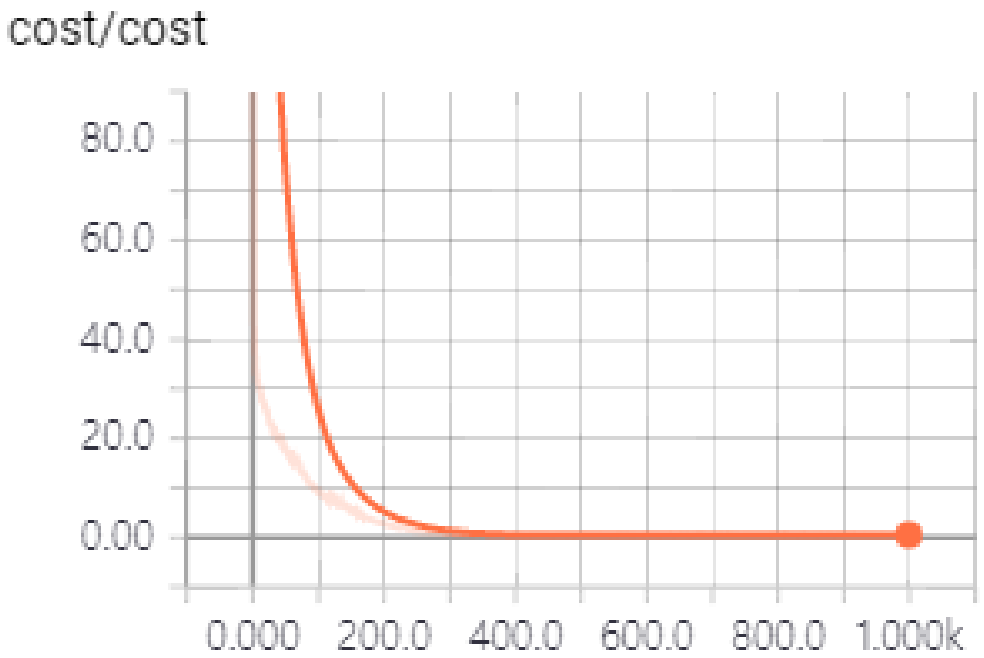
DNN

Deep Neural Network
TensorBoard

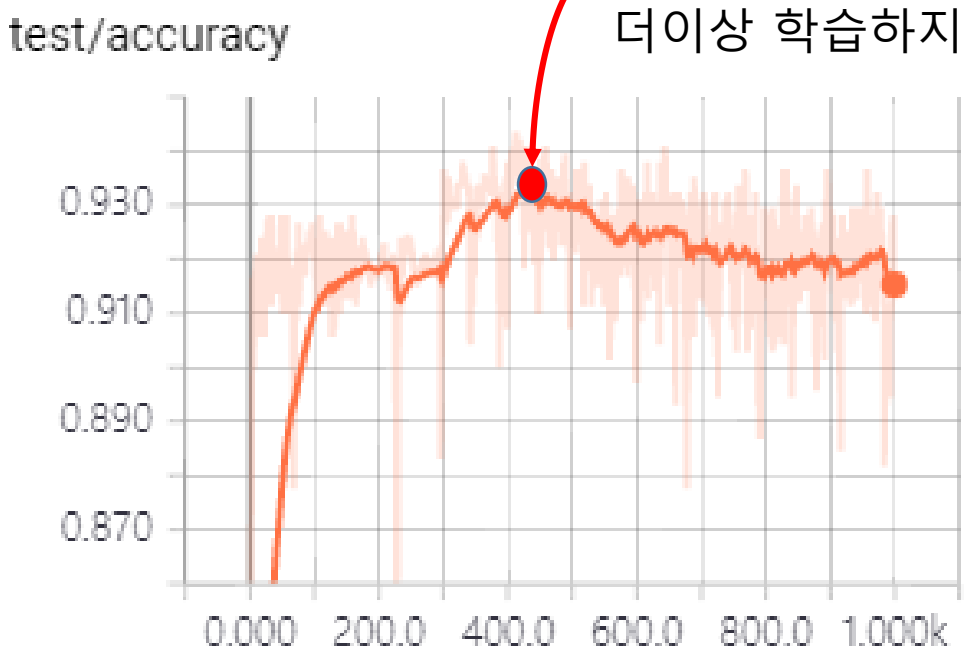


손실함수는 계속 떨어져도 정확도는 어느정도 상승하다가 다시 떨어짐
➔ Overfitting 가능성있음

얼리스탑(Early Stop):
정확도가 최고점에 이르면 더이상 학습하지 않는다

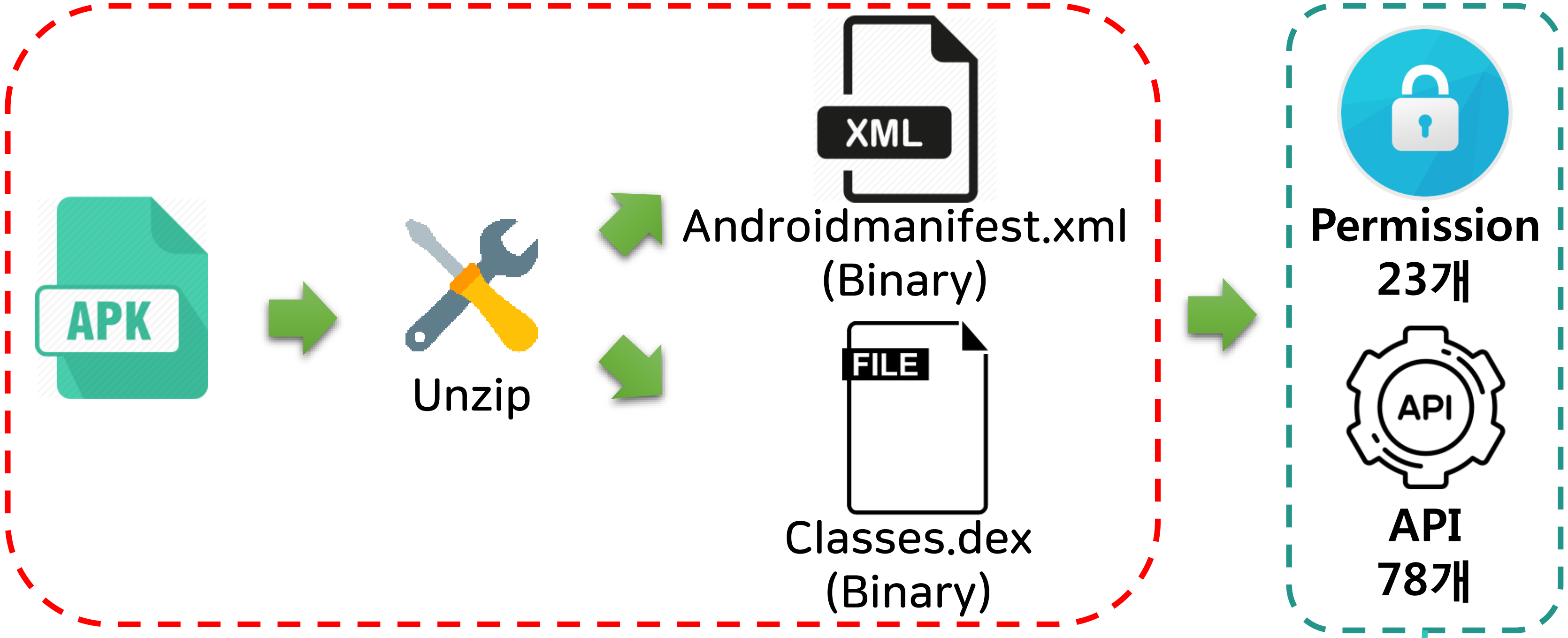


손실 함수 그래프(==정확도에 반비례)

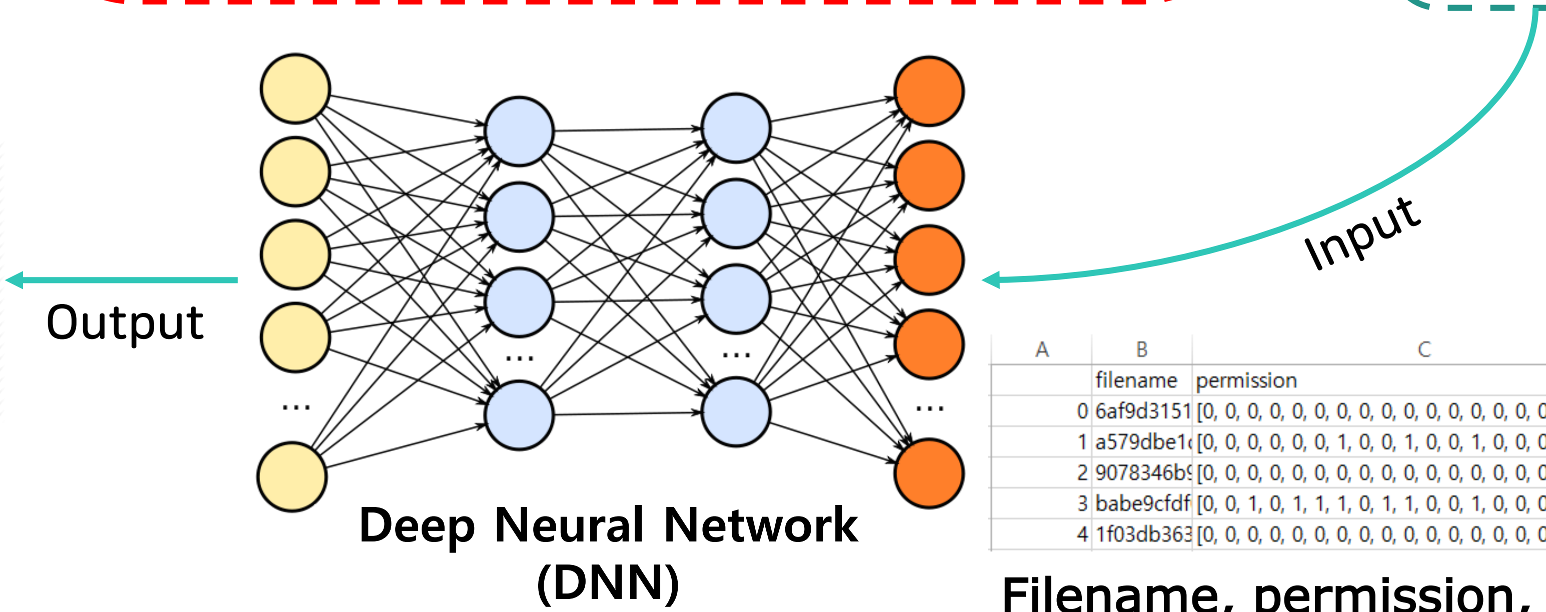


정확도 그래프

과정



A	B	C
filename	class	family
6af9d3151	0	na
a579dbe1(	1	AirPush
9078346b9	0	na
babe9cfdf	1	SMStado
1f03db363	0	na
0178c9d42	0	na
87a4b230a	0	na
2593dae4(	0	na
efd22e96d	0	na
0c2f7fae23	1	AirPush
ac33ed097	1	SMSAgent



A	B	C	D
	filename	permission	sus_b_c
0	6af9d3151	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, (	[0, 0, 0, 0, 0, 0, 0, 0, (
1	a579dbe1(	[0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1	[0, 0, 0, 0, 0, 0, 0, 0, (
2	9078346b9	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1	[0, 0, 0, 0, 0, 0, 0, 0, (
3	babe9cfdf	[0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, (	[0, 0, 0, 0, 0, 0, 0, 0, (
4	1f03db363	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, (	[0, 0, 0, 0, 0, 0, 0, 0, (

Filename, permission, API, Family

정확도

예선

사용한 알고리즘	사용한 Feature 수	결과
SVM	30개	89%~91%(정상 OR 악성)
DNN	30개	93%
DNN	101개	97~98%

본선 1차

사용한 알고리즘	사용한 Feature 수	결과
DNN	101개	92% (학습셋 0.6)
DNN	101개	90%(학습셋 0.9)
DNN	101개	91%(학습셋 0.8)

본선 2차

사용한 알고리즘	사용한 Feature 수	결과
DNN	101개	91% (학습셋 0.6)

링크

https://github.com/jaeyung1001/Malware-detection
https://github.com/haejupark/Malware-detection

THAN

K

Y O

U