학사 학위 논문

안드로이드 앱기반 악성코드의 특징 분류 기술 연구

2020년

한 성 대 학 교

컴 퓨 터 공 학 부

컴 퓨 터 공 학 전 공

장 두 혁

학 사 학 위 논 문
지도교수 허준영

안드로이드 앱기반 악성코드의 특징 분류 기술 연구

Study of the Multi-Class classification technology of Android application-based malware

2020년 1월 7일

한 성 대 학 교

컴 퓨 터 공 학 과

컴 퓨 터 공 학 전 공

장 두 혁

학 사 학 위 논 문
지도교수 허준영

안드로이드 앱기반 악성코드의 특징 분류 기술 연구

Study of the Multi-Class classification technology of Android application-based malware

위 논문을 공학 학사학위 논문으로 제출함

2020년 1월 7일

한 성 대 학 교

컴 퓨 터 공 학 과

컴 퓨 터 공 학 전 공

장 두 혁

장두혁의 공학 학사학위 논문을 인준함

2020년 1월 9일

심 사 위 원
장 _____(인)

심 사 위 원 _____(인)

심 사 위 원 _____(인)

국 문 초 록

안드로이드 앱기반 악성코드의 특징 분류 기술 연구

한 성 대 학 교
컴 퓨 터 공 학 과
컴 퓨 터 공 학 전 공
장 두 혁

본 연구는 Virustotal에서 제공되는악성코드 repository를 통해 정적, 동적분석을 기반으로 추출된 API에서 url에 따라 악성코드 앱을 분석하여, DNN(Deep Neural Network) 과 KNN (K-neighbor network) , SVM(Support Vector machine) 기계학습 모델들을 통해, Malware 12가지 타입으로 타겟으로 분류하여, 분류정확도를 측정한다. 악성코드분류모델은 24372개 데이터를 k-fold 기법k =10으로 정해, 과적합을 줄여, 평균 97.4%, 97%, 97% 으로 분류했다.

키워드: android, malware, SVM, KNN, DNN, machine leaning

목 차

표 목 차

그 림 목 차

제 1 장 서론

제 1 절 안드로이드 스마트폰 보급증가에 따른 악성코드 동향

스마트폰은 빠르게 성장한 시장 중 하나로, 2009년 11월 아이폰이 국내 스마트폰
시장에 진출한 이후 폭발적으로 성장하였으며, 그 후 삼성의 갤럭시 시리즈를 비롯한 많은 국내 스마트폰 제조업체들은 사용자의 요구를 충족시키기 위하여 기술적 혁신을 이루어 다양한 기기를 출시하였으며 이에 따라 매해 스마트폰 사용자도 급격히 증가하고 있다. 조사기업 닐슨에 따르면, 2012년 이후로 안드로이드기반 스마트폰이 아이폰 보급률을 넘어서 선두가 되며, 가장 많이 사용되고 있는 스마트폰 운영체제로, 증가하는 스마트폰 시장에 발맞춰 모바일게임, 어플시장도 급격한 성장을 보이고 있다. 악성 앱이 마켓에 존재하며 사용자에게 위협이 되고 있다. 현대인들은 이동하면서 웹 검색, SNS(Social Network Service), 모바일 뱅킹 등 다양한 서비스를 이용할 수 있게 되었으며 현대인의 생활과 산업의 패러다임까지 변화시키고 있다. 이러한 스마트폰의 발전은 현대인에게 편리함과 유용성을 가져다 주었지만 그 이면에는 보안에는 보급률만큼 강화되지 않고, 취약점이라는 커다란 위협이 도사리고 있다. 스마트폰은 사용자의 위치정보, 연락처, 공인인증서 등 다양한 개인중요정보들이 산재하여 있기 때문에 해커의 공격으로부터 위협적인 피해를 입을 가능성이 크다. 따라서 해커들은 기존 PC 환경의 공격에서 스마트폰 및 모바일 기기로 공격지를 확대하고 있다. 스마트폰에서 발생하는 악성코드의 대부분의 목적은 개인정보를 유출하는데 있다. 특히 안드로이드 플랫폼은 다른 모바일 플랫폼보다 더 많은 보안취약점이 존재하며 그 피해가 증가하고 있다. 한국인터넷진흥원은 악성코드 앱을 통한 피해 확산방지를 위해 통신사, 보안업체 등 협력 을 통해 악성코드 앱 수집, 분석 및 차단을 수행하고 있다. 그러나 악성코드

앱은 날로 진화하고 있어 역부족인 상황이다. 따라서 악성코드 앱을 보다 빠르게 분류가 프로그램이 필요함 관련 연구로는 인공지능 등의 기계학습 기반으로 기존 악성코드 분류 방법에 대한 연구는 이미 활발하게 연구되고있지만, 신종이나 변종된 악성코드에 대한 분류에 대한 연구는 아직은 미비한 결과이다[9][12]. 또 최근 심각한 사이버 위협으로 대두되고 있는 랜섬웨어를 대상으 로 본 논문의 실험 결과를 적용해 봄으로써 본 논 문에서 제시한 방안의 랜섬웨어 탐지에 대한 기계학습의 모델을 짜서 분류의 효율성을 증명하고자 한다[1] [2] .

제 2 절 연구 내용

악성코드(Malware)란, 일반 사용자의 컴퓨터를 감염시켜 악성 행위를 하기 위한 목적으로 만들어진 바이너리 파일을 일컫는다. 아이디나 암호와 같은 개인정보를 유출하는 것에서부터 주요기관에 대한 DDoS공격 까지, 다양한 종류의 악성 행위를 하며, 보통 이들의 행위를 바탕으로 트로이 목마, 바이러스, 웜, 디도스 등으로 구분된다. 이러한 악성코드의 탐지는 여러 가지 많은 분석방법을 통해 만들어진 시그니처에 의해 주로 수행된다. 신종과 변종이 요즘 너무 많이 등장하여[17], 대부분, 기계학습을 통한, 정상코드속에 숨어있는 악성코드Malware에 대해서 KNN모델을 통해서 탐지[15] 또는 DNN를 통해서 탐지[19]하는 연구들이 많다. 아니면은 효과적인 전처리방법을 통해서, 악성코드 탐지 방법에 관한 연구[16] 등으로 정상과 악성 분류하는 연구 또는 신종과 변종을 구별하기 위한 연구의 시작인 유사성 측정이[17] 대부분이다. 그리고 데이터 또한 양이 적어서 정확도가 높다고 말할 수도 없다. 하지만 KNN모델로 악성코드를 분류[18]하는 opcode와 퍼미션, 시스템 정보 등 467개의 특징점을 뽑아서하는 연구는 찾아본 결과 많은 공개 되어있지 않은 것 같다. 또한 본 논문이 신종과 변종 악성코드를 분류하기 위한 도움이 될 수 있는 연구라고 생각한다. 그리고 SVM은 정확도를 높혀주는 특징점만 모아서[20], 분류 정확도와 DNN모델로도 무겁지 않게 은닉층의 깊이와 너비에 따른 정확도 또한 분류 정확도를 측정할 예정이다.

본 논문에서는 24372개의 데이터를 가지고, 악성코드를 분류하는 모델를 설계하여, 정확도를 측정할 것이다. 대부분[15][16]연구들 또한, 데이터를 특징을 추출할 때 API를 통한 정적, 동적 분석을 사용한다.

본 논문은 2장에서 관련 연구인 DNN과 SVM, KNN에 대해서 알아본다. 3장에서는 본 논문에서 제시하는 기계학습 모델에 적합한 모델설계분석 등을 설명한다. 4장에서는 본 논문에서 제시하는 기계학습 모델에 대한 실험과 정확도에 대한 결과를 제시하고, 마지막으로 5장에서는 결론 및 향후 연구를 논의하는 것으로 논문을 마무리 한다.

제 2 장 관련연구

제 1 절 악성코드 수집과 분석

전세계의 다양한 바이러스 백신의 엔진을 이용하여 파일이나 웹사이트 등을 검사 할 수있는 사이트 Virustotal에서 제공하는 API에서 url에 따라 악성코드 앱샘플을 정적 분석, 동적 분석 리포트를 수집하여 분석한다.

1) 정적분석

정적 분석이란 프로그램의 기능을 파악하고 코드나 프로그램의 구조를 분석하는 단계로 악성코드 앱을 실제로 실행해보지 않고 분석하는 방법이다[3][7] . 정적 분석을 통해서 표1과 같이 Basic Properties, 표2와 같이 History, Names , 표3와 같이 Android Info, Certificate Attributes, Subject, 표4와 같이 Permissions, Activites등을 알 수있다.[7]

항목	설명
MD5	업로드 된 파일의 MD5 해쉬값
SHA-1	업로드 된 파일의 SHA-1 해쉬값
SHA-256	업로드 된 파일의 SHA-256 해쉬값
SSDEEP	업로드 된 파일의 SSDEEP 해쉬값

<표1>

항목	설명
First Submission	Virustotal에 업로드 된 처음 날짜
Last Submission	Virustotal에 업로드 된 마지막 날짜
Last Analysis	Virustotal에서 분석 된 마지막 날짜
Earliest Contents Modification	업로드 된 파일이 변경 된 처음 날짜
Latest Contents Modification	업로드 된 파일이 변경 된 마지막 날짜

<표2>

항목	설명
Android Type	안드로이드 Type
	ex. APK
Package Name	패키지 이름
	ex. cn.handpod.handmap
Internel Version	버전 코드, 악성코드 앱을 갱신했을 때 사용됨
	ex. 1
Displayed Version	버전 이름
	ex. 1.0.0

<표3>

번호	권한	설명
1	ACCESS_COARSE_LOCATION	Wi-fi 정보를 이용한 위치 확인(대략적)
2	ACCESS_FINE_LOCATION	GPS 정보를 이용한 위치 확인(정확)
3	ADD_VOICEMAIL	음성 이메일 추가 권한
4	ANSWER_PHONE_CALLS	걸려 오는 전화에 응답하도록 허용하는 권한
5	BODY_SENSORS	사용자가 심장 박동과 같은 신체 내부에서 일어나는 일을 측정하기 위해 사용하는 센서의 데이터에 액세스 할 수 있게 하는 권한
6	CALL_PHONE	전화 통화
7	CAMERA	카메라 권한
8	GET_ACCOUNTS	계정 서비스의 계정 목록에 대한 액세스를 허용하는 권한
9	PROCESS_OUTGOING_CALLS	전화 발신 체크 권한
10	READ_CALENDAR	캘린더 관련 권한
11	READ_CALL_LOG	사용자의 통화 기록을 읽을 수 있도록 하는 권한
12	READ_CONTACTS	주소록 관련 권한

13	READ_EXTERNAL_STORAGE	외부 저장소에서 읽을 수 있도록 하는 권한
14	READ_PHONE_NUMBERS	장치의 전화 번호에 대한 읽기 액세스 허용하는 권한
15	READ_PHONE_STATE	폰 상태 관련 권한
16	READ_SMS	SMS 문자 관련 권한
17	RECEIVE_MMS	MMS 수신 관련 권한
18	RECEIVE_SMS	SMS(문자) 수신 관련 권한
19	RECEIVE_WAP_PUSH	WAP 푸시 권한
20	RECORD_AUDIO	오디오 녹음 권한
21	SEND_SMS	SMS(문자) 보내기 권한
22	USE_SIP	SIP 서비스를 사용할 수 있도록 하는 권한
23	WRITE_CALENDAR	캘린더 쓰기 관련 권한

24	WRITE_CALL_LOG	사용자의 통화 기록 데이터를 쓰지만 읽지는 못하게 하는 권한
25	WRITE_CONTACTS	주소록 쓰기 관련 권한
26	WRITE_EXTERNAL_STORAGE	외부저장장치 관련 권한
27	ACCEPT_HANDOVER	호출 app이 다른 app에서 시작된 호출을 계속할 수 있도록 하는 권한
28	ACCESS_BACKGROUND_LOCATION	app이 백그라운드에서 위치 정보에 액세스 할 수 있도록 허용하는 권한
29	ACCESS_CHECKIN_PROPERTIES	체크인데이터베이스 속성테이블의 읽고 쓰기 권한
30	ACCESS_LOCATION_EXTRA_COMMANDS	추가적인 위치 제공하는 GPS 사용 권한
31	ACCESS_MEDIA_LOCATION	Media에 기록된 위치 정보 접근 권한
32	ACCESS_NETWORK_STATE	네트워크 정보에 접근 권한
33	ACCESS_NOTIFICAT	알람 허용하는 권한

	ION_PO LICY	
34	ACCESS_WIFI_STA TE	Wi-fi 정보에 접근 권한
35	ACCOUNT_MANAG ER	계정 인증 시스템의 사용을 허용하는 권한
36	ACTIVITY_RECOG NITION	신체적 활동 인식 권한
37	BATTERY_STATS	배터리 상태 권한
38	BIND_ACCESSIBILIT Y_SER VICE	AccessibilityService 사용 시 요구되는 권한
39	BIND_APPWIDGET	AppWidget 사용 시 요구되는 권한
40	BIND_AUTOFILL_S ERVICE	AutofillService 사용 시 요구되는 권한
41	BIND_CALL_REDIR ECTION_ SERVICE	CallRedirectionService 사용 시 요구되는 권한
42	BIND_CARRIER_ME SSAGING CLIENT_SERVICE	CarrierMessagingClientService 사용 시 요구되는 권한
43	BIND_CARRIER_SE RVICES	Carrier App에 사용되는 권한
44	BIND_CHOOSER_TA RGET_SE RVICE	ChooserTargetService 사용 시 요구되는 권한
45	BIND_CONDITION_P ROVIDE R_SERVICE	ConditionProviderService 사용 시 요구되는 권한
46	BIND_DEVICE_AD MIN	Device Administration Receiver 사용 시 요구되는 권한

47	BIND_DREAM_S ERVICE	DreamService 사용 시 요구되는 권한
48	BIND_INCALL_S ERVICE	InCallService 사용 시 요구되는 권한
49	BIND_INPUT_M ETHOD	InputMethodService 사용 시 요구되는 권한
50	BIND_MIDI_DEVI CE_SERVI CE	MidiDeviceService 사용 시 요구되는 권한

51	BIND_NFC_SERVICE	HostApduService 또는 OffHostApduService 사용 시 요구되는 권한
52	BIND_NOTIFICATION_LISTENER_SERVICE	NotificationListenerService 사용 시 요구되는 권한
53	BIND_PRINT_SERVICE	PrintService 사용 시 요구되는 권한
54	BIND_QUICK_SETTINGS_TILE	app이 설정 파일에 빠르게 binding 되도록 하는 권한
55	BIND_REMOTEVIEWS	RemoteViewsService 사용 시 요구되는 권한
56	BIND_SCREENING_SERVICE	CallScreeningService 사용 시 요구되는 권한
57	BIND_TELECOM_CONNECTION_SERVICE	ConnectionService 사용 시 요구되는 권한
58	BIND_TEXT_SERVICE	TextService 사용 시 요구되는 권한
59	BIND_TV_INPUT	TvInputService 사용 시 요구되는 권한
60	BIND_VISUAL_VOICEMAIL_SERVICE	VisualVoicemailService 사용 시 요구되는 권한
61	BIND_VOICE_INTERACTION	VoiceInteractionService 사용 시 요구되는 권한
62	BIND_VPN_SERVICE	VpnService 사용 시 요구되는 권한
63	BIND_VR_LISTENER_SERVICE	VrListenerService 사용 시 요구되는 권한
64	BIND_WALLPAPER	WallpaperService 사용 시 요구되는 권한
65	BLUETOOTH	블루투스 연결 허용
66	BLUETOOTH_ADMIN	블루투스 장치를 검색하고 페어링
67	BLUETOOTH_PRIVILEGED	사용자 상호작용 없이 블루투스 장치를 페어링하고 연락처 액세스 또는 메시지 액세스를 허용 또는 허용하지 않도록 하는 권한
68	BROADCAST_PA	app 패키지 제거 알림 권한

	CKAGE_REMOVED	
69	BROADCAST_SMS	SMS 관련 권한

70	BROADCAST_STICKY	sticky intent 허용 권한
71	CALL_COMPANION_APP	InCallService 사용 시 요구되는 권한
72	CALL_PRIVILEGED	전화 통화 긴급 통화 포함
73	CAPTURE_AUDIO_OUTPUT	출력되는 소리를 캡처하도록 허용하는 권한
74	CHANGE_COMPONENT_ENABLED_STATE	app 구성 요소의 사용 여부를 app이 변경할 수 있는 권한
75	CHANGE_CONFIGURATION	Configuration 관련 권한
76	CHANGE_NETWORK_STATE	인터넷(네트워크) 권한
77	CHANGE_WIFI_MULTICAST_STATE	Wi-Fi 멀티 캐스트 모드로 들어갈 수 있게 하는 권한
78	CHANGE_WIFI_STATE	Wi-fi 사용 권한
79	CLEAR_APP_CACHE	설치된 앱 캐쉬 삭제 권한
80	CONTROL_LOCATION_UPDATES	라디오에서 위치 업데이트 알림을 활성화 / 비활성화 할 수 있게 하는 권한
81	DELETE_CACHE FILES	캐쉬 파일 제거 권한
82	DELETE_PACKAGES	패키지 삭제 권한
83	DIAGNOSTIC	app이 리소스를 진단하는 권한
84	DISABLE_KEYGUARD	keyguard를 비활성화 할게 하는 권한
85	DUMP	시스템 서비스에서 dump info를 검색 할 수 있게 하는 권한
86	EXPAND_STATUS_BAR	상태 표시줄 확장 권한
87	FACTORY_TEST	루트 권한으로 app 실행

88	FOREGROUND_SERVICE	Service.startForeground에서 사용되는 권한
89	GET_ACCOUNTS_PRIVILEGED	계정 서비스의 계정 목록에 대한 액세스를 허용하는 권한
90	GET_PACKAGE_SIZE	패키지 사용 공간 관련 권한
91	GET_TASKS	태스크 관련 권한
92	GLOBAL_SEARCH	전역 검색 시스템에서 데이터에 액세스 할 수 있도록 허용
93	INSTALL_LOCATION_PROVIDER	위치 관리자에 위치 제공자를 설치하도록 허용
94	INSTALL_PACKAGES	패키지 설치 권한

95	INSTALL_SHORTCUT	Launcher에 바로 가기를 설치할 수 있도록 허용
96	INSTANT_APP_FOREGROUND SERVICE	foreground service를 만들 수 있게 허용하는 권한
97	INTERNET	네트워크 소켓을 열도록 허용
98	KILL_BACKGROUND_PROCESSES	전화를 할 수 있게 허용
99	LOCATION_HARDWARE	app이 geofencing api같은 hardware에서의 위치 권한
100	MANAGE_DOCUMENTS	문서에 대한 관리 권한
101	MANAGE_OWN_CALLS	자체 호출을 관리하는 app을 허용
102	MASTER_CLEAR	타사 app에서 사용할 수 없게 하는 권한
103	MEDIA_CONTENT_CONTROL	재생중인 콘텐츠를 파악하고 재생을 제어할 수 있게 하는 권한
104	MODIFY_AUDIO SETTINGS	전체 오디오 설정을 수정할 수 있도록 하는 권한
105	MODIFY_PHONE_STATE	전원 켜기, mmi 등의 전화 통신 상태를 수정할 수 있게 하는 권한
106	MOUNT_FORMAT_FILESYSTE	이동식 저장소에 파일 시스템을 포맷 할 수 있게 하는 권한

	MS	
107	MOUNT_UNMOUNT_FILESYSTEMS	이동식 저장소의 파일 시스템 마운트 및 마운트 해제를 허용하는 권한
108	NFC	NFC 권한
109	NFC_TRANSACTION_EVENT	NFC 트랜잭션 이벤트를 수신 할 수 있게 하는 권한
110	PACKAGE_USAGE_STATS	구성 요소의 사용 통계를 수집하도록 허용
111	PERSISTENT_ACTIVITY	API 15에서 제거, 활동을 지속하도록 하는 권한
112	READ_INPUT_STATE	키보드 관련 권한
113	READ_LOGS	로그 관련 권한
114	READ_SYNC_SETTINGS	동기화 설정을 읽을 수 하는 권한
115	READ_SYNC_STATS	동기화 상태를 읽을 수 있게 하는 권한
116	READ_VOICEMAIL	시스템의 음성 메일을 읽을 수 있도록 하는 권한
117	REBOOT	장치를 재부팅 할 수 있게 하는 권한
118	RECEIVE_BOOT_COMPLETED	부팅 완료 관련 권한
119	REORDER_TASKS	작업의 Z- 순서를 변경할 수 있게 하는 권한
120	REQUEST_COMPANION_RUN_IN_BACKGROUND	컴패니언 app을 백그라운드에서 실행할 수 있게 하는 권한
121	REQUEST_COMPANION_USE_DATA_IN_BACKGROUND	컴패니언 app이 백그라운드에서 데이터를 사용할 수 있도록 허용하는 권한
122	REQUEST_DELETE_PACKAGES	패키지 삭제를 요청할 수 있게 하는 권한
123	REQUEST_IGNORE_BATTERY	사용자 승인 없이 권한을 보유하는 권한

	_OPTIMIZATIONS	
124	REQUEST_INSTALL_PACKAGES	패키지 설치를 요청할 수 있게 하는 권한
125	REQUEST_PASSWORD_COMPLEXITY	사용자에게 화면 잠금을 특정 복잡성 수준으로 업데이트하라는 메시지를 표시하도록 허용하는 권한
126	RESTART_PACKAGES	API 15에서 제거
127	SEND_RESPOND_VIA_MESSAGE	수신 통화 중 메시지를 통한 응답 작업을 처리 할 수 있도록 다른 app에 요청을 보낼 수 있게 하는 권한
128	SET_ALARM	알람 관련 권한
129	SET_ALWAYS_FINISH	액티비티 종료 권한
130	SET_ANIMATION_SCALE	전역 애니메이션 크기 조정 요소를 수정하는 권한
131	SET_DEBUG_APP	디버깅을 위해 응용 프로그램을 구성하는 권한
132	SET_PREFERRED_APPLICATIONS	API 15에서 제거
133	SET_PROCESS_LIMIT	실행 프로세스 제한 권한
134	SET_TIME	시스템 시간을 설정할 수 있게 하는 권한
135	SET_TIME_ZONE	시스템 시간대를 설정할 수 있게 하는 권한
136	SET_WALLPAPER	배경 화면을 설정할 수 있게 하는 권한
137	SET_WALLPAPER_HINTS	배경 화면 힌트 관리 권한
138	SIGNAL_PERSISTENT_PROCESSES	모든 영구 프로세스에 신호를 보내도록 요청할 수 있게 하는 권한
139	SMS_FINANCIAL_TRANSACTIONS	금융 app이 필터링 된 SMS 메시지를 읽을 수 있게 하는 권한
140	STATUS_BAR	상태 표시줄 관련 권한

141	SYSTEM_ALERT_WINDOW	다른 모든 app 위에 표시된 유형을 사용하여 창을 만들 수 있도록 허용하는 권한
142	TRANSMIT_IR	적외선으로 작동하는 장치에서 지원
143	UPDATE_DEVICE_STATS	기기 상태를 업데이트하도록 허용하는 권한
144	USE_BIOMETRIC	기기에서 지원되는 생체 인식 정보를 사용하도록 허용하는 권한
145	USE_FINGERPRINT	API 28에서 제거, USE_BIOMETRIC 대신 응용 프로그램이 요청해야 함
146	USE_FULL_SCREEN_INTENT	notification full screen intents 사용 시 필요한 권한
147	VIBRATE	진동 관련 권한
148	WAKE_LOCK	화면 켜기 관련 권한
149	WRITE_APN_SETTINGS	APN 쓰기 관련 권한
150	WRITE_GSERVICES	Google 서비스 맵을 수정할 수 있도록 하는 권한
151	WRITE_SECURE_SETTINGS	보안 시스템 설정을 읽거나 쓸 수 있게 하는 권한
152	WRITE_SETTINGS	시스템 설정 쓰기 권한
153	WRITE_VOICEMAIL	시스템의 기존 음성 메일을 수정하고 제거하도록 허용하는 권한

<표4>

안드로이드에서 앱은 설치되기 전에 사용자에게 자원에 접근할 수 있도록 퍼미션을 요청하고 사용자에게 알린다. 악성 앱은 정상 앱과 비교하여 특정 퍼미션을 더 자주 요청하는 경향이 있다. Felt 등의 연구에서는 실제 모바일 악성 앱을 분석하여 사용하는 퍼미션을 추출하였고, 퍼미션으로 악성 앱을 탐지하는 방법의 효과를 설명하였다[4]. Sarma 등의 연구에서는 정상 앱과 악성 앱의 퍼미션을 비교 분석하여 악성 앱을 탐지하는데 중요한 퍼미션을 제시하였다[5]. 정적 분석 방법의 다른 모델은 API(Application Program Interfaces) 기반의 탐지 모델이다. API 기반의 탐지 모델에서는 앱의 소스 코드로부터 API 사용정보를 추출하여 악성 앱

을 분류하는데 사용한다[6].

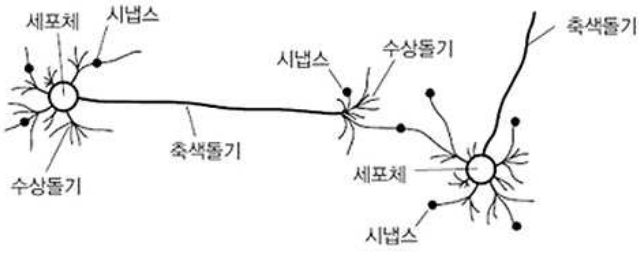
2) 동적분석

동적 분석이란 프로그램의 기능을 파악하기 위햇 악성코드 앱의 실행 전, 후 상태를 조사하여 분석방법으로 악성코드 앱을 직접 실행하여 분석하는 방법이다. Network Communication, File System Actions, Process And Service Actions, Synchronization Mechanisms & Signals, Modules Loaded 등을 알 수있다[7]. Taint analysis 으로 불리는 기법으로 특정 데이터에 마킹을 하고, 어플리케이션 코드 내에서 데이터가 전파되는 과정을 모니터링하여 데이터의 흐름을 추적하는 기법이다. 스마트폰은 가상 머신 상에서 실행되므로 본 기법을 적용하기에 적합하다는 평가를 받았지만 낮은 수준까지 데이터의 흐름을 추적하기 위한 오버헤드로 인해 실제 환경에 적용하기에는 어려움이 있다고한다.

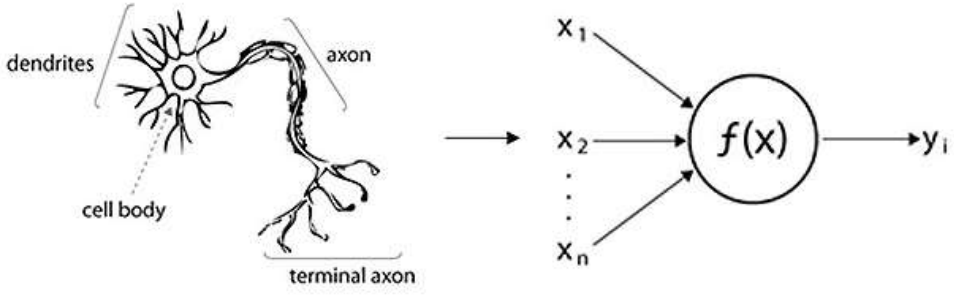
제 2 절 DNN

1) ANN란 ?

신경계는 그림1과 같이, 뉴런들의 결합으로 구성되어있다. 신경세포체는 가지돌기를 통해 자극을 받아들이며, 축삭돌기를 통해 전기자극을 전달한다. 인간을 포함한 많은 생물들은 이렇듯 단순한 뉴런들의 결합을 환경에 적합하도록 진화시켜왔다. 그 결과 신경계는 빛을 감지하거나, 다리를 움직이거나, 기억하거나, 상상하는 등의 복잡한 일을 할 수 있게 되었다. 그림2처럼 인공 신경망은 뉴런에서 신경세포체를 노드로, 축삭돌기를 링크로 모방한 네트워크를 말한다. 각 노드들은 신경세포체와 마찬가지로 정보를 받고 입력된 정보들을, 그것을 전달하는 과정에서 유의미한 결과를 얻을 수 있는 계산하여 출력을 수행한다.



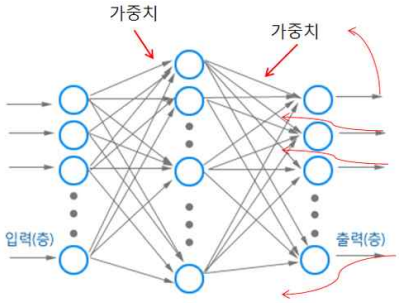
<그림1>



<그림2>

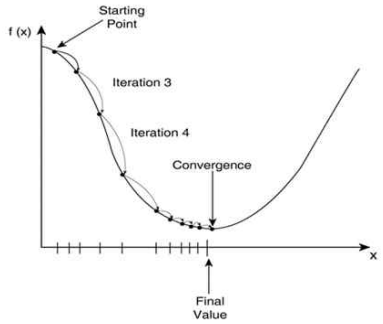
2) DNN란?

2-1에서 설명했듯이, 맥컬룩-피츠 뉴런메델인 로젠블라트의 퍼셉트론[11]을 기반으로, 입력층 input layer 와 출력 output layer 로 이뤄진 것을 인공신경망이라고 한다[2]. 여러개의 단일 인공신경망들을 연결하여, 입력층(input layer) 와 출력층(output layer) 사이에 하나의 은닉층(hidden layer)들로 다층퍼셉트론(MuLti Perception)구성하고, 여러개의 은닉층을 구성하여 심층 인공신경망(Deep artificial Neural Network)을 구성한다[6]. 분류 및 수치예측을 하기 위해서, 주로 사용되고, 이미지 트레이닝이나 문자인식과 같은 분야에서 매우 유용하게 쓰이고 있다. 심층 인공신경망이 표준 오류역전파 알고리즘으로 학습한다. 결과의 오차를 줄이기 위해서 가 노드에서 다음 노드로 이어지는 가중치를 조절하는 학습 방법으로, 이 때 가중치를 조절하기위해서 다시 뒤로 되돌아와 수정을 한다.



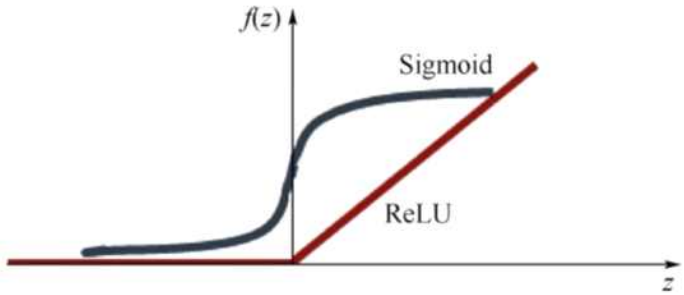
<그림3>

이때, 가중치(weight)들은 아래의 등식을 이용한 확률적 경사 하강법(stochastic gradient descent)[10]을 통하여 갱신될 수 있다.아래 아래 그림과 같이, x는 가중치, $f(x)$는 그에 따른 오차다. 최대로 오차를 줄이는 방향으로 가중치를 변경하기 위한 방법으로, 아래 식에서 W을 수정해 간다. 이로써 신경망의 결과값이 달라진다



$Y = f(X) = \sigma(\sum_i w_i \; g_i(X))$ 그림<4-1>

은닉 계층에 대한 활성화 함수는 네트워크에 비선형성을 적용하는 데 필요하다. 활성화 함수가 적용되고, 결과는 네트워크 내의 다음 뉴런으로 전달된다. 대부분의 비선형 함수들이 사용되는데, 주로 로지스틱 회귀분석에서 쓰이는 시그모이드 함수가 많이 사용된다. 최근에는 ReLU활성화 함수가 자주 쓰인다. 풀어쓰면 Rectified linear unit으로 "선형유닛을 개선" 한다는 의미로써 비선형의 구조를 가진 데이터를 분석하는데 활용되는 활성화 함수다. 또한 기울기를 이용해 가중치를 업데이트하므로 평평한 활성화 함수는 문제가 있다. 가중치의 변화는 활성화 함수의 기울기에 좌우되기 때문이다. 작은 기울기는 곧 학습 능력이 제한된다는 것을 의미하고 이를 일컬어 신경망에 포화(Saturation)가 발생했다고 한다. 때문에 포화가 일어나지 않게 하기 위해서는 입력 값을 작게 유지해야 한다. 수식은 입력 신호에도 영향을 받는다. 때문에 가중치를 너무 작게 만들 수도 없다. 이럴 때 좋은 방법은 입력 값이 0~1사이에 놓이도록 그 크기를 조정하는 것이다.

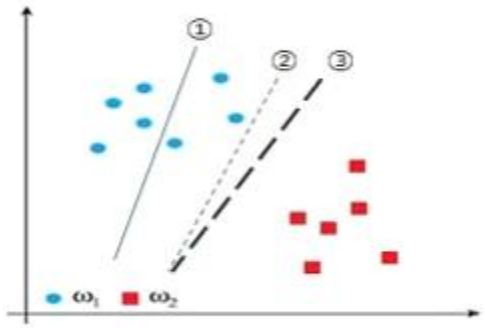


<그림4-2>

제 3절 SVM

1) SVM

SVM(Support Vector Machines)은 이원 패턴 인식 문제를 해결하기 위해 제안된 학습기법으로 두 개의 범주를 구성하는 데이터들을 가장 잘 분리해 낼 수 있는 결정면을 찾아내는 모형이다. 1990년대 말부터 텍스트 범주화 실험에 적용 하기 시작한 SVM은 다른 학습 방법 보다 우수한 성능을 보임에 따라 최근 문자인식, 얼굴인식 뿐만 아니라 텍스트 범주화 연구 분야에서도 주목을 받고 있다. 결정 트리(Decision Tree) 기반 분류기는 기계학습 분야에서 널리 사용되는 규칙 표현 방법으로 객체를 분류하는 규칙들이 트리의 형태로 나타난다[5]. SVM의 가장 큰 장점은 다른 분류기에 비해 일반화 능력이 뛰어나다는 것이다. 아래와같은 그림은 분류기의 일반화 능력을 보여주는 그래프이다. ① 은 두 개의 클래스로 이뤄진 데이터를 분류하는 기준으로 적합하지 않다. 따라서 ②,③ 이 데이터를 분류하는 기준이 될 것이다. 기존 분류기는 초기값 ①에서 시작하여 ② 를 찾고 학습과정을 끝낸다. 하지만 SVM은 ③ 을 찾는 최적화 과정을 거치기 때문에 일반화 측면에서 다른 분류기 이상의 성능을 보여준다. 또한 비선형적인 데이터를 분류할 수 있는 기능과 각 Feature의 가중치를 두기 때문에 데이터의 유연한 분류가 가능하다.

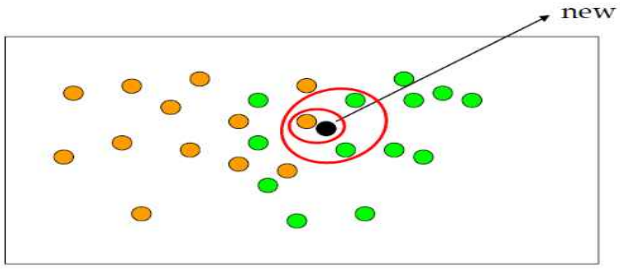


<그림5>

제 4절 KNN

1) KNN

k-nearest neighbor는 데이터를 분류하고 새로운 데이터 포인트의 카테고리를 결정할 때 K 개의 가장 가까운 포인트를 선점하고 그중 가장 많이 선택된 포인트의 카테고리로 이 새로운 데이터를 분류하는 방법이다. k-nearest neighbor에서 고려해야 할 사항은 아래 그림1과 같이, 알고리즘의 핵심 부분이 대상 포인트와의 거리에 대한 측정이고, 이를 계산하는 방법으로 무조건 유클리드 거리 측정 방식과 맨해튼 거리 추적방식을 사용 한다[6].



<그림6>

KNN의 유사도 측정 방법에는 여러 가지 방법이 있다. 데이터의 흐름과 분포에 따라 어떤 측정법을 사용하는지에 따라 결과가 달라진다. 대부분의 경우, 유클리드 거리 측정을 사용하여 유사도를 측정한다. KNN은 비 모수적 방법이기 때문에 어떤 분포이든 상관없이 사용할 수 있고, 알고리즘의 특성상 쉽고 이해하기 직관적이다. K의 값을 크게 줄 경우, 대세의 흐름을 알 수 있지만, 세분하게 분류하지 못한다는 단점이 있다. 반대로 K의 값이 작을 경우 너무 미세하게 구분되어 오류가 생길 확률이 커진다. 데이터의 개수 와 클래스의 개수에 따라 K를 적절히 선택해야 올바른 분류가 가능해진다

제 3 장 모델 설계

제 1 절 전처리 방식 & K-fold방식

1) 전처리방식

DNN모델을 적용시키기 위해서, 아래 그림과 같이, X 독립변수에 대해서 MinMaScaler()로 통한 Normalize화를 시켰다. Y종속변수 타겟에 대한 숫자화하였다. LabelEncoder()함수를 활용하여 0~9까지 정수화시켰다. np_utils.to_categorical()함수를 통해 matrix화 시켜서 10차수로 변환 시켰다. 그리고 나머지 String 타입의 데이터의 컬럼은 제거하였다.

2) 모델 설계

3층의 레이어와 하나의 hidden레이어층으로 구성된 모델을 구성하여, 첫 번째 레이어 입력층의 입력뉴런 개수를 66개 출력 뉴런개수 500개 설정하고, 두 번째 레이어 은닉층의 인력 뉴런 개수를 500개 출력 뉴런개수 100개로, 세 번째 레이어 출력층의 입력뉴런 개수를 100개 출력 뉴런개수 10개로 분류하였다. 사용된 활성함수는 ReLU 함수는 아래그림 와 같이 음수에서는 0, 양수에서는 해당 값을 그대로 사용하는 방식이며, Sigmoid 함수의 단점인 값을 변형시키는 점을 보완한 함수이다. 마지막 세 번째로 3번째 은닉 층에서 출력 층 사이에는 Softmax 확률화시켜서 출력하여 분류했다 [8].

svm모델에 적용하기 위해서, 데이터셋 columns수를 조절했다[20]. 1차적으로 모든 columns에 대해서 모델을 적용시 정확도가 올라가지않음

1) 레이어의 노드의 개수를 증가시켜서 적용

2) 레이어의 깊이를 증가 시켜 적용

2차적으로, 1차의 방식으로 모든 columns을 넣어서 학습을 시킨 정확도는 30프로를 넘지 못해서, 각 columns에서 사용횟수를 파악하여 많이 467개 중에 208개만 뽑아서 학습에 사용하기로 했다.

아래 표7와 같이, 퍼미션은 1/3 사용과 op관련된 columns들은 1/2, 시스템

은 1/3 정도를 사용

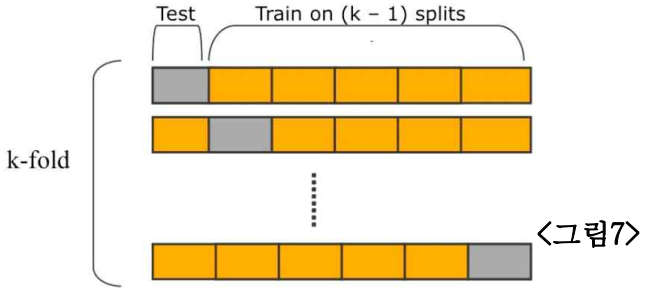
	사용	전체
퍼미션	54	157
op	144	282
시스템	10	27

<표7>

그리고, 나머지는 동일하게 Y종속변수 타겟에 대한 숫자화하였다. LabelEncoder()함수를 활용하여 0~10까지 정수화 시켰다. np_utils.to_categorical()함수를 통해 matrix화 시켜서 11차수로 변환 시켰다. 그리고 나머지 String 타입의 데이터의 컬럼은 제거하였다. knn모델을 적용할때도 그대로 사용했으며, knn모델에서는 K값을 1부터 10까지 대입하여서 정확도가 높은 홀수인 K값을 확인되며, alorithm 값을 auto로 하여, 여러 가지 alorithm를 넣어 최적의 값이 나올 때 로 설정하였다. inkowski의 P값을 기본값 2로 설정하여 유클리드 거리를 사용하여 거리척도를 계산했다.

2) K-fold방식(K겹 교차방식)

아래 그림와 같이 데이터셋을 여러 개로 나누어 하나씩 테스트셋으로 사용하고 나머지를 모두 합하여 학습셋으로 사용하여 성능값에 모델의 평균을 계산하는 방법이다. 이렇게 하면, 가지고 있는 데이터의 100%를 테스트셋으로 사용할 수있고, 학습 알고리즘이 더 정확하고 안정적인 모델을 만들수있다. K겹 교차검증은 중복을 허락하지않은 리샘플링 기법이기 때문에 모델 성능의 추정에 분산이 낮다[6].



<그림7>

제 4 장 실험

제 1 절 실험환경 &데이터셋과 분류타겟 1)실험 환경

본 논문은 표5와 같은 기준으로 평가실험을 진행한다. GPU구동을 사용하기 위해서cuda10.1를 설치한다. <표5>

Hardware dependecies	NVIDIA GPU 1060
	RAM 16GB memory
Software depencecies	Python 3.6
	Pycharm
GPU components	NVIDIA GPU driver
	CUDA 10.1
Modules	pandas
	numpy
	sklearn - LabelEncoder
	sklearn - MinMaxScaler
	np_utils
	Sequential,Dense
	StratifiedKFlod
	tf

2) 악성코드 데이터셋

2.1와 2.2의 정적분석과 동적분석으로 추출하여 얻어진 데이터셋을 사용하기전에, 전처리 과정이 중요하다. **24372개**의 데이터와 **467개**의 columns인 정적특징의 정보와 op code 와 permission들로 구성되어 있다. opcode 는 282개와 permission 157개로 구성되어있다.

3) 분류타겟

분류하고자 하는 타겟은 아래 표6 과같이 ADM_type으로 적혀있는 Malware 10가지 타입에 대해 분류하다. 항목은 아래 그림과같다.

KIND	개수
Adware	**14080**
backdoor	**1047**
HackerTool	**332**
Ransom	**2080**
Trojan	**511**
Trojan-Banker	**929**
Trojan-Clicker	**30**
Trojan-Dropper	**297**
Trojan-SMS	**3215**
Trojan-Spy	**1851**

<표6>

제 2 절 모델 검증평가

1) 실험결과

다중분류하기 위해서, 데이터 전처리한 가공된 데이터를 train/ test data으로 나누어, 모델을 과적합을 피하기 위해서 k-fold방식으로 학습과 검증을 통해 정확도 측정하여 평균를 낸다. DNN모델에 대한 결과는 10개의 정확도의 평균값인, 아래 그림과같이 97.4%정도의 정확도를 보여준다. 비록 깊이 보다는 너비를 넓게 하여, 간단한 모델을 설계했지만, 높은 분류결과를 보여주었다.

10 fold accuracy 결과
0.974615384664291
0.975025641086774
0.9703076923688253
0.9758461538339272
0.975025641086774
0.9762564103175433
0.973589743577517
0.9752307692919022
0.975641025689932
0.977641025628799
평균 :0.973918

<표7>

아래 그림4와 같이, confusion matrix의 결과를 확률로, 모델에 대한결과는 97프로의 정확도를 보여줬다.

<Classification report>

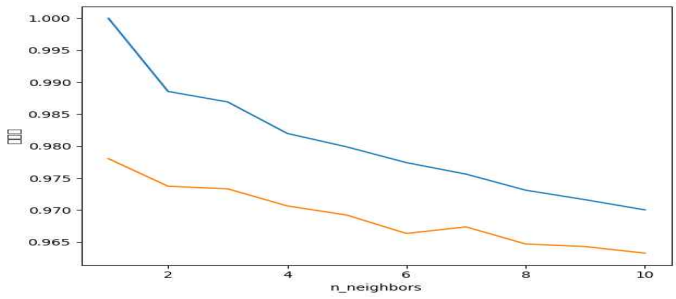
```
Classification_report
              precision    recall  f1-score   support

           0       0.98      0.99      0.98      2801
           1       0.92      0.88      0.90       199
           2       0.89      0.79      0.84        75
           3       0.98      0.99      0.99       377
           4       0.88      0.78      0.83        96
           5       1.00      0.99      0.99       183
           6       0.75      1.00      0.86         6
           7       0.77      0.67      0.72        61
           8       0.98      0.99      0.98       659
           9       1.00      0.99      1.00       379
          10       0.94      0.85      0.89        39

    accuracy                           0.97      4875
   macro avg       0.92      0.90      0.91      4875
weighted avg       0.97      0.97      0.97      4875

총 테스트 갯수:4875
정확도 : 0.97
```

<그림8>

정확도 0.97

<그림4>

KNN모델에 대한 결과는 아래그림과같이 KNN때 97.3%의 정확도를 보여줬다. 아래 그림5와 같이 k = 3 테스트 세트의 정확도 : 0.973 이고, k = 5 테스트 세트의 정확도 : 0.969로 k =3일 때 k가 홀수일 때 97.3%의 정확도를 가진다고 볼 수있다.



<그림9>

2)오답률에 대한 분석

전체 데이터 24372개중에 틀린 DNN모델의 틀린 값은 1270개로, 나머지 대략적인 2.5프로에 해당되는 틀린 값들 중에, 어느 것이 많이 틀렸나 확인을 하자면, 아래 그림을 보면, Trojan-clicker 타입의 악성코드를 분류하는 게 0.5프로로 제일 많이 틀렸거, 성능이 좋아서 이 타입을 잘맞춘다고 말은 못하지만, 수치상으로 봤을 때 Trojan-Spy을 많이 맞추는 것으로 확인이 된다. <표8>

	KIND	Y타겟	구성	틀린개수	비율
총개수			24372	1270	0.052109
	Adware	0	14080	368	0.026136
	backdoor	1	1047	142	0.135626
	HackerTool	2	332	52	0.156627
	Ransom	3	2080	58	0.030382
	Trojan	4	511	238	0.465753
	Trojan-Banker	5	929	56	0.06028
	Trojan-Clicker	6	30	15	0.5
	Trojan-Dropper	7	297	165	0.555556

	Trojan-SMS	8	3215	125	0.03888
	Trojan-Spy	9	1851	27	0.014587

KNN모델 같은 경우에는, k=3일 경우 24372개중에 256개를 k=5경우 284개를 틀렸다. k=3일 때와 k=5일 때 Trogan-spy의 오답률은 보이지 않았고, k =5일 때는 Trogan, Trogan-Clicker, Trogan- Dropper가 그나마 1퍼센트 조금넘는 오답률을 보였고, k =3일때는 Trogan, Trogan-Clicker, Trogan -Dropper로 동일하게 수치는 다르지만, 1퍼센트의 오답률을 보였다. KNN으로는 Trojan-Spy를 정확히 맞추는 걸로 확인 또한 되었다.<표9>

	KIND	Y타겟	구성	k=5	k=3	비율(k=5)	비율(k=3)
총개수			24372	284	256		
	Adware	0	14080	35	43	0.002486	0.003054
	backdoor	1	1047	46	39	0.043935	0.037249
	HackerTool	2	332	20	13	0.060241	0.039157
	Ransom	3	1909	2	2	0.001048	0.001048
	Trojan	4	511	78	68	0.152642	0.133072
	Trojan-Banker	5	929	8	6	0.008611	0.006459
	Trojan-Clicker	6	30	4	4	0.133333	0.133333
	Trojan-Dropper	7	297	47	43	0.158249	0.144781
	Trojan-SMS	8	3215	44	38	0.013686	0.01182
	Trojan-Spy	9	1851	0	0	0	0

제 5 장 결론

본 논문은 기계학습방식으로, 다중분류 DNN, KNN, SVM 모델를 구현하였다. 기존에 악성코드 탐지가 아닌, 분류를 위한 모델을 간단한 구조를 활용하여, 만드므로써, 정확도가 높은 모델의 결과를 보여주었다.

DNN모델을 짤 때는, 과연 은닉층의 개수가 많을수록 정확도가 높아질 줄 알았으며, 그래서 최대 10층까지 증가 시켜서 했지만, 10~12프로정도 올라는 수치만 보이고, 40프로를 넘지 못했지만, 기존 columns수에서 3~4배 되는 수치 노드 수를 증가 시키고 감소시키는 형태로 잡으니 전반적으로, 분류를 잘하는 규칙같지 않은 규칙을 찾은 것같다. 그래도 은닉층 노드개수를 정하는 규칙을 찾아내지는 못했다.

기계학습 DNN모델은 초반에 너비보다 깊이를 생각하여, 많은 은닉층을 넣고, 계속 실험을 한 결과, 낮은 분류의 정확도와 또한 낮은 학습률을 보였다. 모델의 깊이보다는, 너비로 늘려 노드 수의 증가 시키므로써, 정확도를 높였다.

KNN모델과 DNN모델 둘다 비슷한 분류 정확도값이 나왔지만, 악성코드 오답률를 보았을 때 KNN보다는 DNN모델이 오답률이 적은 것으로 확인되었다. 그래도 400개가 넘는 특징을 추출하여, 신종이나 변종 악성코드 또한, 여러 은닉층 또는 이를 활용하여, 그뒤에 패턴을 분석하여[14], 신종과 변종을 향후 분류 가능할 예정으로 보인다. 안드로이드 앱 파일들에 대한 정보를 수집하여 타임라인 데이터를 구현하고, 그 파일 안에서도 신종 악성코드 탐지 기법을 통해 의심되는 파일을 탐지할 수 있도록 탐지도구를 개발하여, 의심 파일들에 정보와 인터넷 접속 이벤트 정보들의 시간을 정렬함으로써, 좀 더 빠르게 은닉된 악성코드를 발견견 할 수 있다고 생각이 든다[13]. 그렇다고, 모든 특징점으로 악성코드에 분류에 사용되는 것이 아니라, SVM모델에서와 같이, 주로 많이 퍼미션과 시스템 함수 사용횟수등을 주로 사용되는 특정 columns에 대해서 악성코드 분류에 도움을 준다는 것 또한 본 연구를 통해 알게 되었다. 그래서 앞으로 신종과 변종 악성코드를 탐지하거나 분류하는 데있어서 퍼미션과 시스템 함수 사용횟수등,

다양한 코드의 흔적들이 매우 중요하다는 것을 본 연구를 통해서 알게 되었다.

참 고 문 헌

[1] Hiran V. Nath, Babu M. Mehtre, “Static Malware Analysis Using /인트로
Machine Learning Methods”, G. Martinez Perez et al. (Eds.): SNDS 2014, CCIS 420, (2014).

[2] A권혜윤. “딥러닝을 이용한 악성코드의 분류“ VOL.- NO.- (2018)

[3] 정재민. “API 콜 및 Permission 기반 기계학습을 사용하는 안드로이드 악성코드 탐지 시스템의 성능 분석“ VOL.- NO.- (2019)

[4] Soo-tai Nam, Seong-yoon Shin, Chan-yong Jin. (2019). Data Mining for the k-Nearest Neighbors Classification Algorithm Based on a Machine Learning. 한국정보통신학회 종합학술대회 논문집, 23(2), 279-282. knn

[5] 박부영. “잠재의미색인(LSI) 기법을 이용한 kNN 분류기의 자질 선정에 관한 연구“ VOL.- NO.- (2004)

[6] 머신러닝교과서with 파이썬, 사이킷런, 텐서플로 11, 12장

[7] 악성코드 repository 기반 악성코드 앱 수집 및 분류 기술 연구 에 관한연구 산학협력기관 숭실대학교

[8] 신진호. “SVM, DNN 분류기를 이용한 다중생체신호 기반 인간감정인식“ VOL.- NO.- (2018)

[9] 박성빈. “공통속성과 가중치를 이용한 변종 악성코드 탐지 방법“ VOL.- NO.- (2013)

[10] 조경우, 정용진, 강철규, & 오창헌. (2019). 미세먼지 예측을 위한 기계 학습 알고리즘의 적합성 평가. 한국정보통신학회논문지, 23(1), 20-26.

[11] 김대식. (2016). 김대식의 인간 vs 기계: 인공지능이란 무엇인가. 동아시아.

[12] 박남열, 김용민, 노봉남. (2006). 우회기법을 이용하는 악성코드 행위

기반 탐지 방법. 정보보호학회논문지, 16(3), 17-28.

[13] 송재훈. “파일구조 기반 신종 윈도우 악성코드 탐지 방법에 관한 연구“ VOL.- NO.- (2015)

[14] 송인수. “악성코드 패턴 분석 및 시각화 기법 연구“ VOL.- NO.- (2011)

[15] 유진현. “Study on DNN based Android malware detection method for mobile environment“ VOL.- NO.- (2017)

[16] 배성재,조재익,손태식,문종섭. “Native API 의 효과적인 전처리 방법을 이용한 악성 코드 탐지 방법에 관한 연구“ 정보보호학회논문지 VOL.22 NO.4 (2012):785-796

[17] 권희준, 김선우, & 임을규. (2012). Multi N-gram 을 이용한 악성코드 분류 시스템. 보안공학연구논문지 (Journal of Security Engineering), 9(6).

[18] Choi Jiyeon,Kim HeeSeok,Choi Jangwon,Song Jungsuk. “A Malware Classification Method Based on Generic Malware Information“ Lecture Notes in Computer Science VOL.2015 NO.9490 (2015):329-336

[19] 유진현 (Jinhyun Yu), 서인혁 (In Hyuk Seo), 김승주 (Seungjoo Kim). “모바일 환경에 적합한 DNN 기반의 악성 앱 탐지 방법에 관한 연구“ 정보처리학회논문지. 컴퓨터 및 통신시스템 VOL.6 NO.3 (2017):159-168

[20] 김기현(Ki-Hyun Kim),최미정(Mi-Jung Choi). “선형 SVM을 사용한 안드로이드 기반의 악성코드 탐지 및 성능 향상을 위한 Feature 선정“ 韓國通信學會論文誌 VOL.39 NO.8 (2014):738-745

ABSTRACT

Study of the Multi-Class classification technology of Android application-based malware

Chang Duhyeuk

Major in Computer Engineering

Dept. of Computer Engineering

Hansung University

This study analyzes malware app according to url in API extracted based on static and dynamic analysis through malicious code repository provided by Virustotal, categorizes and measures DNN (Deep Neural Network) and KNN (K-neighbor network) and SVM (Support Vorachine Machine) machine learning models as 12 types of target hardware models. The malicious code classification model designated 2,4372 data as k-fold technique k = 10 and reduced overconformity, and classified them as 97.4%, 97%, and 97% on average.

Keywords: android, malware, SVM, KNN, DNN, machine leaning