정 보 보 호 학 회 지
제 33 권 제1호, 2023. 2

악성코드 유포사이트 탐지 기술 동향 조사

오 성 택*, 신 삼 신**

요 약

인터넷 사용매체 및 네트워크 접속방법이 다양해지면서 인터넷 사용량은 매우 빠르게 증가하고 있다. 이러한 인터넷은 현대사회에서 꼭 필요한 자원이지만 악성코드, 스팸, 개인정보 유출 등 이를 악용한 범죄도 증가하고 있다. 또한 전 세계적으로 유행중인 코로나로 인해 관련된 접종정보, 동선, 재난문자 등으로 위장한 피싱 공격도 증가하고 있다. 대다수의 공격자들은 사이버 범죄를 저지르기 위해 악성코드 유포사이트를 통해 악성코드를 유포한다. 이러한 범죄를 예방하기 위해선 악성코드 유포사이트에 대한 초기 대응이 필수이며, 사용자가 악성코드 유포사이트에 접근하기 전에 차단할 수 있는 실시간 탐지 기술이 필요하다. 본 논문에서는 이러한 탐지 기술 중 URLDeep, POSTER, Random-Forest, XGBoost와 같은 기계학습을 이용한 탐지 기술의 연구동향을 조사하였다.

Ⅰ. 서 론

공격자는 사용자가 쉽게 접근할 수 있는 웹사이트를 통해 사용자의 시스템에 악성코드를 설치하고 금융사기 및 개인 정보 유출 등의 공격을 수행한다. 이러한 공격을 수행하는 방법은 여러 가지가 있으며, 대부분 악성코드를 사용하여 특정 악성 행위를 일으켜 피해를 입힌다. 이러한 범죄에 악용되는 악성코드로 인한 피해 예방 및 대응을 위한 기존의 전통적인 사이버 보안 방식들이 있으나, 현재의 사이버 범죄는 점차 교묘해지고 고도화되고 있어 최근의 공격들은 탐지할 수 없게 되었다. 특히 매우 방대한 양의 데이터 분석 및 탐지가 필요한 현재에는 기존의 방식보다 발전된 머신러닝 기술 등을 활용한 탐지 기술이 필요하다. 본 논문에서는 악성코드 대응 기술 중에서 악성코드 유포사이트 탐지를 위한 다양한 기술들을 정리하고, 정리된 기술을 바탕으로 악성코드 유포사이트 탐지를 위하여 사용할 수 있는 기술을 정리하였다.

Ⅱ. 관련연구

기존 악성코드 유포를 탐지하는 기술은 정적분석 방식과 동적분석 방식으로 분류할 수 있다. 정적분석은 점검대상 홈페이지를 웹 크롤러를 통해 수집하여, 해당 홈페이지 내에 악성링크가 있는지를 검색하는 방식이다. 이 때 탐지패턴으로 사용하는 악성링크는 국내외 다양한 수집 및 공유 경로를 통해 분석한 악성코드 유포지 또는 악성 스크립트 문자열로 탐지가 가능하다. 탐지 패턴을 사용하므로 빠르지만, 신·변종 악성코드는 탐지하기 어려운 단점이 존재한다. 동적 분석은 악성코드 유포 가능성이 있는 주요 홈페이지(기존 악성코드 유포사이트, 웹하드 서비스, MS/구글 제공, 병원/쇼핑몰 등)를 가상화된 사용자 PC환경에서 홈페이지를 직접 접속한다. 이 때 비정상적인 레지스트리 변경, 악성코드 다운로드, 명령제어 서버 접속 등과 같은 악성행위가 발생하는지 분석하는 행위기반 방식이다. 신종 악성코드에도 대응할 수 있는 장점이 있으나, 가상화 환경으로 인해 분석 속도가 느리고 안티 가상화 기법에는 대응하기 어렵다는 단점이 있다.

2.1. 非-머신러닝 악성코드 유포 사이트 탐지 기술

휴리스틱 방식은 알려진 공격의 시그니처를 기반으로 탐지하고, 정적분석 방식은 URL과 웹 페이지의 콘텐츠의 관계를 기반으로 탐지한다. 여기서 특징의 종류는 URL 정보, 호스트 정보(IP, WHOIS,

본 연구는 대한민국 정부(산업통상자원부 및 방위사업청) 재원으로 민군협력진흥원에서 수행하는 민군기술협력사업의 연구비 지원으로 수행되었습니다.(협약번호 UM21306RD3)
* 한국인터넷진흥원 보안기술단 침해대응기술팀 (선임연구원, angelrick@kisa.or.kr)
** 한국인터넷진흥원 보안기술단 침해대응기술팀 (수석연구원, sss@kisa.or.kr)

Geographic, Connection speed 등) 콘텐츠 정보(HTML, Javascript, ActiveX 등), 링크 그래프 정보(노드 깊이, 중심, 클러스터링 계수, 링크 그래프의 메타데이터)가 있다. 동적 분석 방식은 Honey-pot(HoneyC, Capture-HPC, MITRE Honey-client) 같은 제어 환경에서 웹 페이지의 실행을 기반으로 하고, 악성 웹의 비정상 행위(launched process, registry changes, memory heap allocation, open ports 등)를 파악할 수 있다. 하이브리드 분석 방식은 식별 한계와 시간 제약을 제거하기 위해 상기 두 분석 방식을 사용한다[1].

2.1.1. 블랙리스트 기반 분석[2]

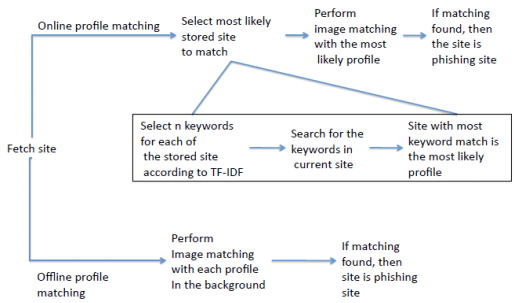
이 분석의 해결 목적은 웹 사이트에 숨겨진 악성코드를 자동으로 탐지하고 능동적으로 대응하는 것이다. 동작 구조는 정적분석을 통해 홈페이지 은닉 악성코드 유포 사이트를 분석 및 탐지하는 블랙리스트 방식으로 패턴 기반 탐지를 수행하거나, 동적 분석을 통해 가상화 환경에서 모의실행을 통해 내부 링크를 포함한 모든 연속된 페이지에 대해서 악성코드 은닉 여부를 검사한다. 입력 데이터 유형은 웹 페이지의 HTML, 소스 및 링크 URL이다. 처리 알고리즘은 악성코드 유포 패턴의 포함 여부를 검사할 웹 페이지를 수집하기 위한 웹 크롤링(crawling) 기능을 통해 수집된 웹 페이지의 소스에서 중계 및 유포사이트로 의심되는 링크 URL를 수집한다. 악성코드 유포 패턴이 확인된 웹 페이지의 URL과 패턴 등은 데이터베이스에 저장하거나 업데이트된다. 분석 결과로 새로운 악성코드 패턴을 찾기 위해서 소스코드 내에 악성코드 패턴이 은닉되어 있는지 확인할 때 난독화가 되어 있으면 패턴의 형태를 찾을 수 없고 도메인 리스트의 수가 증가하면 검사 시간이 길어지는 한계가 있다.

2.1.2. 사용자 평판 기반 의심 URL 분석[3]

이 분석의 해결 목적은 대량의 웹 트래픽에서 악성 웹 사이트를 찾는데 크롤러 기반 방식은 효율적이지 못하므로 사용자의 웹 요청만 검사하는 방식으로 더 효율성을 높이는 것이다. 동작 구조는 대량의 웹 트래픽에서 DBD(drop-by download) 공격을 탐지하는 구조로 도메인 평판 기반으로 의심스러운 웹 사이트를 식별하고, 식별된 의심 사이트를 샌드박스를 통해 분석하여 탐지 시간을 줄인다. 입력 데이터 유형은 실제 네트워크상의 대량의 웹 트래픽이다. 처리 알고리즘은 WHOIS 데이터베이스의 모든 도메인 쿼리에 대해 신뢰할 수 없으므로 DNS 서버로부터 오는 쿼리로부터 정상과 의심을 구분하는 새로운 평판 속성을 제안하였다. 분석 결과로 실제 네트워크 환경에서 실험 결과는 94% 정확도였고, 컴퓨팅 시간은 개선된 샌드박싱 접근 방식보다 12배 이상 효율적이었다. 또한 1일 560,000개의 URL 요청을 처리했고, 실제 운영 시에도 블랙리스트에 없었던 알려지지 않은 악성 웹 사이트도 식별하였다.

2.1.3. 하이브리드 방식 조합 분석[4]

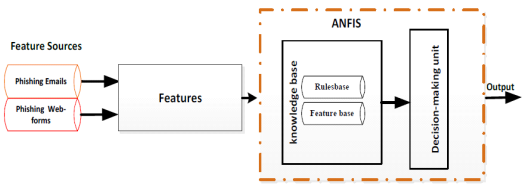
이 분석의 해결 목적은 정확도를 높이기 위해 블랙리스트, 휴리스틱, 화이트리스트 방식을 조합한 기법을 사용하여 피싱 여부를 분류하는 것이다. 동작 구조는 대상 웹 사이트의 URL, SSL, Image, HTML, contents, script 정보를 대상으로 특징을 추출하고 프로파일링 하여 저장하고, 웹 사이트 방문 시 저장된 프로파일링 정보와 매칭(키워드, 이미지)하여 피싱 여부를 판단한다. 입력 데이터 유형은 Phishing websites(Phishtank에서 검증된 1,000개 피싱 사이트)이다. 처리 알고리즘은 피싱을 탐지하기 위해 신뢰 웹 사이트의 프로파일을 사용하는 PhishZoo 알고리즘, 프로파일 생성 시 사이트 로고의 특징 추출은 SIFT(Scale Invariant Feature Transform) 알고리즘을 사용한다. 분석 결과로 블랙리스트 방식과 유사한 96% 정확도였고, 제로데이 피싱 공격과 작은 사이트에 대한 표적 공격을 분류한다.



〔그림 1〕 PhishZoo 알고리즘 적용 프로세스〔4〕

2.1.4. 퍼지 매칭 분석[5]

이 분석의 해결 목적은 블랙리스트 기반보다 좀 더 개선된 방식으로 뉴로퍼지 기법을 사용하여 분류 정확도를 향상시킨다. 동작 구조는 효과적인 특징을 추출하기 위해 소스를 식별하여 적합한 형식으로 형상 준비와 정규화(0-1 범위) 수행, Adaptive Negro-Fuzzy 추론 시스템(ANTIS)으로 퍼지 모델을 훈련 및 시험하여 모델의 효과를 측정한다. 입력 데이터 유형은 Phasing web-forms, e-mails, e-web-forms 형태의 56개 특징을 사용하고, 2-fold 교차검증 방법으로 훈련 데이터셋과 시험 데이터셋을 구성한다. 처리 알고리즘 ANFIS는 규칙 기반과 특징 기반으로 지식베이스와 의사결정단위(decision-making unit)를 만들고, IF-THEN 퍼지 규칙을 사용하여 특징을 학습하고 검증한다. 분석 결과로 전체 평균 오류율이 훈련과 시험 모두 1.6%에 도달했고, 전체 시험 평균 정확도는 98.4%였다.



〔그림 2〕 지능형 피싱 보안 아키텍처〔5〕

2.1.5. 공격자의 행위 분석[6]

이 분석의 해결 목적은 공격자의 습관적인 URL 조작을 통해 새로운 URL을 식별하기 위해 매우 작은 블랙리스트 셋으로도 많은 수의 악성 웹 페이지를 탐지할 수 있는 URL 기반 웹 필터링 모델을 구축한다. 동작 구조는 데이터셋을 정상 URL과 악성 URL(경유지 또는 유포지 URL)로 나누고, 의미 있는 문자열(토른들의 집합)로 특징을 추출하고 도메인, 경로, 파일명으로 그룹을 나누어 퍼지 기반 유사도 매칭을 통해 악성 확률의 합을 계산한다. 입력 데이터 유형은 Tier-1(SK broadband)에서 수집한 1,529,433 악성 URL(1,509,230 경유지, 20,203 유포지), 알려지지 않은 URL 3.42%, 악성 URL의 특징 유형은 Alexa 기반 속성, 익스플로잇 킷의 코드 변경, 경유지 URL의 경로 깊이와 IP 존, 경로명의 속성, 파일명, 쿼리 문자열과 확장자가 있다. 처리 알고리즘은 모델에서 후보 URL의 악성을 측정하는 스코어링 알고리즘을 공식화하고, 유사도 매칭 모델은 먼저 후보 URL을 호스트, 경로명과 파일명 깊이 간격으로 구문을 분석하고 유사한 IP 접두사 또는 유사한 호스트명을 검색하여 가장 적합한 유사도를 측정한다. 이와 유사하게 URL 깊이뿐만 아니라 국가코드 최상위 수준 도메인(ccTLD)과 일반 최상위 수준 도메인(gTLD)을 기반으로 IP 위치를 확인하고 IP/Domain의 유사성은 심각도가 높고 유사한 경로명과 파일명도 높은 의심으로 처리한다. 분석 결과로 공격자 습관 중심 행위는 호스트, 경로, 파일명 특징 기반의 유사도 매칭에서 정확도 70%이상으로 탐지율이 높았고, 일반적인 머신 러닝보다 빠르게 악성 URL을 탐지하고 메모리 사용량도 1.5% 미만이었다.

2.1.6. 웹 페이지 시각적 유사도 분석[7]

이 분석의 해결 목적은 웹 페이지에 있는 텍스트 유사도, 폰트 색상 및 크기, 이미지의 특징을 시각적 유사도 분석을 통해 피싱 사이트인지 여부를 탐지하는 것이다. 동작 구조는 웹 페이지에서 다양한 특징들을 기반으로 추출하고 식별할 수 있는 시그니처를 생성하고, 시각적 유사도로 탐지하는 방법은 DOM, 시각 특징(텍스트 폰트의 색상, 크기, 배경색, 폰트 패밀리 등), CSS, 이미지 픽셀, 시각 퍼셉션(Gestalt Laws 적용), 혼합 특징 기반 모델을 사용하고 각 모델의 효과를 분석한다. 입력 데이터 유형은 웹 페이지의 HTML, DOM tree, 시각 특징, CSS 유사도, 이미지 픽셀, 시각적 퍼셉션 관련 특징들과 전자 우편의 의심스러운 키워드나 URL로 구성된다. 처리 알고리즘은 텍스트 특징의 제목 키워드, URL 키워드를 대상으로 TF-IDF, 프로파일(URL, SSL, certificate, contents) 유사도, Block/Layout 유사도, EMD, Bayesian 알고리즘을 사용하여 검증한다. 피싱을 탐지하는 많은 접근 방법들의 장점과 한계를 분석한 결과, 적합한 한 가지 기술은 없었고 아직은 더 연구와 개발이 필요한 영역이다.

2.2. 머신러닝 악성코드 유포 사이트 탐지 기술

분류 기법에서 머신러닝 방식은 Naive Bayes, Decision Tree, Boosted Decision Tree, Support

Vector Machine, Logistic Regression이 있다. 그래프 분석 방식은 그래프 속성을 사용하여 악의적인 웹 사이트를 결정하는 것으로 링크 그래프는 서로 다른 깊이와 링크 그래프의 집계 하위 그래프를 사용하여 웹 페이지의 스팸 유형의 악의적인 특성을 결정하며 SVM-light 기술이 사용된다.

2.2.1. 빠른 특징 추출로 악성 URL 분석[8]

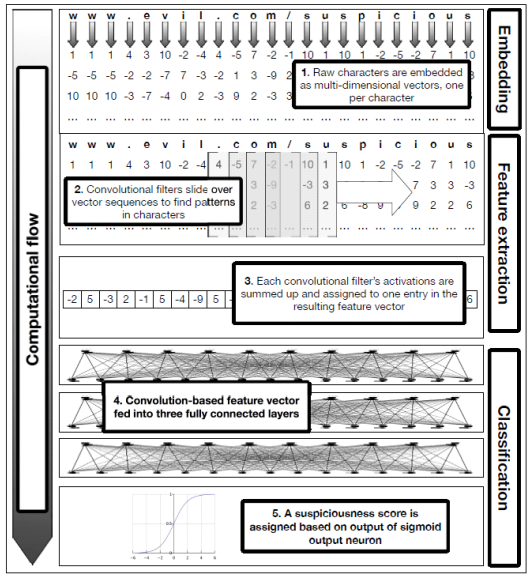
이 분석의 해결 목적은 피싱 공격을 탐지하기 위해 URL을 분석하고 분류하는 것이다. 동작 구조는 URL의 분류 및 분석을 위해 문장 부호, 특수 기호 또는 단일 문자 분포로 구성된 특징을 포함하는 URL 데이터셋으로부터 문자 N-grams으로 다양한 특징 집합을 추출한다. 입력 데이터 유형은 Phishing URL(Static URL, Live URL), Spam URLs and Domains이다. 처리 알고리즘은 다양한 온라인 학습기(Perceptron and Averaged Perceptron, Passive-Aggressive, Confidence Weighted 등)와 배치 학습기(J48, PART, Random-Forest)를 모두 사용하여 빠르고 강력하며 정확한 분류기를 제시하고, 모델 구축을 위해 분류기 성능을 즉성 및 재훈련 방법과 비교한다. 분석 결과로 Static URL, 데이터셋에 대한 분류는 평균 75%이였고, 일부 온라인 학습기(AROW, AP, CW)의 경우에는 99%에 근접하였다. 피싱 URL 분석은 URL 및 도메인 분석을 통한 스팸 웹 사이트 탐지와 달랐다.

2.2.2. 데이터마이닝 분석 기반 보안 빅데이터[9]

이 분석의 해결 목적은 인터넷 범죄 행위를 탐지하기 위해 URL을 분석하는 분류 모델 데이터마이닝이다. 동작 구조는 URL을 수집하고 특징을 추출하여 훈련/시험 데이터셋을 생성한 후 RIPPER 알고리즘으로 훈련하고 생성된 룰셋을 사용하여 시험 데이터에 적용하고 각 파라미터를 계산한다. 입력 데이터 유형으로 훈련 데이터셋은 악성 URL 400개와 정상 URL 200개로 구성하고, 시험 데이터셋은 악성 300개와 정상 150개로 구성한다. 처리 알고리즘은 RIPPER(jRip 데이터 마이닝 알고리즘, weka)을 사용한다. 1050개의 URL 시험 데이터셋에 대한 분석 결과는 TP 0.802, TN 0.8714, FP 0.1285, FN 0.1971, 정확도 82%가 나왔다.

2.2.3. 문자 수준 임베딩 기반 딥러닝[10]

이 분석의 해결 목적은 기존 머신러닝 학습 방식의 수동적 Feature Engineering의 한계를 넘어서기 위해 원시(raw) 입력에서 특징을 자동 추출하는 CNN 방식 딥러닝 학습이다. 동작 구조는 문자 임베딩, 특징 탐지, 분류기의 3개 컴포넌트로 구성되어 있고, 문자 임베딩은 인쇄 가능한 영어 문자의 알파벳을 다차원 특성 공간에 임베드하여 입력 문자열의 원시 문자 시퀀스를 2차원 텐서로 인코딩한다. 특징 탐지는 전체 문자 시퀀스 내에서 중요한 로컬 시퀀스 패턴을 감지한 다음 이 정보를 고정길이 특징벡터로 집계한다. 분류는 밀집된 신경망을 사용하여 탐지된 특징을 분류하고 이러한 모든 구성 요소는 확률적 경사 하강을 사용하여 공동으로 최적화한다. 입력 데이터 유형은 VirusTotal에서 약 2개월 동안 무작위로 추출한 19,067,879개의 URL을 기반으로 한 시험에서는 87개의 URL에서 사용가능한 유효문자로 구성된 입력 어휘를 사용하였으며, VirusTotal에서 기록된 1천8백만 건의 Cuckoo 샌드박스 실행으로 추출된 파일경로 및 레지스트리 키를 기반으로 한 시험에는 100개의 유효한 인쇄 가능한 문자로 구성된 입력 어휘를 사용한다. 처리 알고리즘은 정규화에서는 모델 훈련과 과적합(overfit)을 방지하기 위해 BatchNorm과 Dropout(0.5)(registry key의 경우 0.2)을 정의하여 사용한다. 고밀

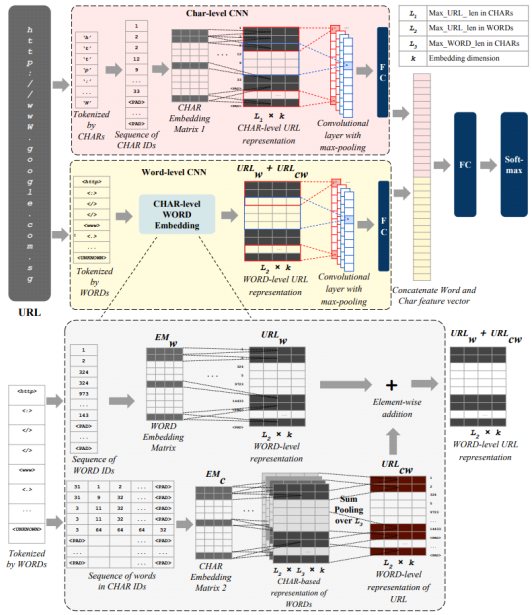


〔그림 3 eXpose 뉴럴 네트워크 아키텍처 개념도〔10〕

도 신경망은 Dense 유닉과 DenseSigmoid 계층으로 구성하여 Convolution 기반 특징이 주어지면 비선형 커널을 학습하고 시그모이드 계층 출력은 최종 고밀도 계층의 출력에서 입력 문자열이 악성일 확률을 제공한다. 이진 교차(binary-cross) 엔트로피를 사용하여 탐지 예측 손실을 측정한다. 분석 결과로 문자열이 200보다 길면 문자열의 시작 부분을 잘라냈고, 경험적으로 문자 임베딩 층이 32차원이 정확성과 계산 복잡성 사이의 좋은 균형을 이룬다고 판단한다. ROC 곡선을 사용하여 분석 결과를 제시하고 경험상 합리적인 임계값은 FPR 비율이 10-4와 10-3에 중점을 둔다. 더 긴 문자열은 훈련하는 계산 비용으로 인해 더 복잡한 아키텍처를 시도하지 못하였다.

2.2.4. End-to-End 딥러닝 프레임워크[11]

이 분석의 해결 목적은 URL의 어휘 속성 기반 접근 방식 한계를 극복하고, 악성 URL 탐지를 위해 비선형 URL 임베딩을 학습하는 End to End 학습 프레임워크이다. 동작 구조는 입력 URL에 대해 임베딩 계층, 컨볼루션 계층, 완전히 연결(fully connected) 계층을 거쳐 URL 분류 Softmax로 출력된다. 여기서 URL 입력 문자열은 임베딩 계층에서 문자 수준 분기와 단어 수준 분기로 나누어 처리한다. 입력 데이터 유형은 VirusTotal에서 레이블된 URL 정보를 수집(2017.05~06), 중복 URL을 제거하고 편향(bias)을 줄이기 위해 URL 도메인의 빈도를 5% 미만으로 제한한다. 훈련 데이터셋은 정상 4,683,425개와 악성 316,575개이고 시험 데이터셋은 정상 9,366,850개와 악성 633,150개(정상94%:악성6%)이다. 처리 알고리즘은 CNN을 URL 문자열의 문자와 단어 모두에 적용하여 공동으로 최적화된 프레임워크에 포함되는 URL을 학습한다. 이 접근법을 통해 모델은 여러 유형의 시멘틱 정보를 캡처하며, 기존 모델에서는 어렵다. 컨볼루션 연산은 문자 수준과 단어 수준 모두에 적용한 후에 완전히 연결된 계층이 적용되며 두 가지 모두에 대한 드롭아웃으로 정규화하고, 모델은 역전파를 사용하여 옵티마이저에 의해 학습된다. 분석 결과로 URLNet 내에서 3가지 형태의 성능을 측정한 결과에서 문자 수준과 단어 수준 URLNet의 성능은 비슷했지만, URLNet(Full)은 이 두 가지보다 더 우수하고 일관된 성능을 제공한다(FPR이 낮으면 단어 수준 URLNet이



〔그림 4 URLNet 아키텍처〔11〕

문자 수준 URLNet보다 성능이 뛰어나고 FPR이 높을수록 그 반대가 됨).

2.2.5. 보안 URL 주소 분석[12]

이 분석의 해결 목적은 악성 URL의 시퀀스 탐지를 위한 ED(Event De-nosing) CNN을 사용한다. 동작 구조는 훈련 단계에서 URL 시퀀스 정보를 입력받아 이력 도메인(historic domain) 기반과 순간(momentary) URL 기반 특징을 추출하여 특징 벡터들을 레이블로 분류한다. 입력 데이터 유형으로 웹 사이트 훈련 데이터셋은 2015년도 각 시기별(총 6번)로 악성(총 16,573개)과 정상(총 35,368개)을 구분하여 수집한다. 처리 알고리즘 EDCNN은 악성 URL 시퀀스가 포함된 웹 사이트에서 리디렉션되는 정상 URL의 부정적인 영향을 줄이기 위한 새로운 CNN 알고리즘이다. 분석 결과로 EDCNN은 사용자가 감염된 웹 사이트에 접속하지만 브라우저 핑거프린팅으로 인해 익스플로잇 코드를 얻지 못하는 경우 CNN에 비해 잘못된 경고의 47%를 줄여 악성코드 감염의 운영비용을 낮추는 것으로 나타났다.

Ⅲ. 학습 모델 특징 분석

3.1. 학습 데이터 및 모델 특징 분석

3.1.1. URLDeep[13]

1) 학습데이터

- P2P Social Network의 악성 URLs (VirusTotal, PhishTank)
- 모바일 프로파일링과 활동 행위에 대한 1백만 개 URL 주소로 구성된 데이터셋(Malicious dataset : Training 1,016,575 & Testing 933,150)

〔표 1〕 URLDeep 모델에서 사용한 데이터셋 구성

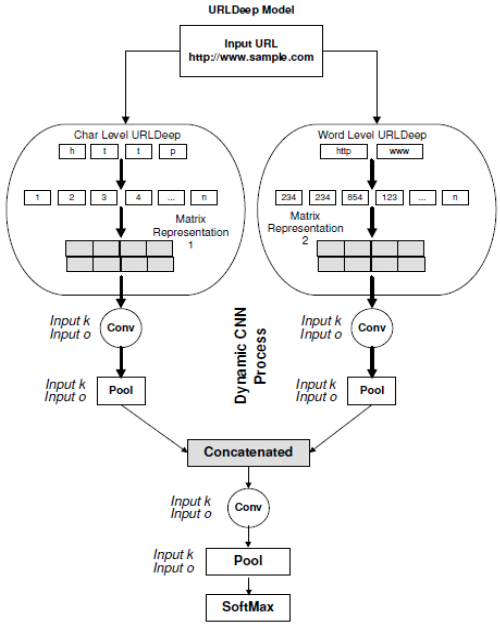
	Benign	Malicious	Total
Training	4,983,425	1,016,575	6,000,000
Testing	9,066,850	933,150	10,000,000
Total	14,050,275	1,049,725	16,000,000

2) 학습방식/모델유형

- CNN 개념에 dynamic max pooling을 적용하는 Dynamic CNN(D-CNN) 방식 Deep Learning으로 실시간 소셜 네트워크상의 악성 URL을 예측하였다.
- URLDeep 모델은 결정된 레이어 번호를 사용하여 그래프를 다시 계산하고, 유사한 신호 부분을 동일한 CNN 채널에 할당함으로써 가중치의 최적 정렬을 결정한다.
- 특징 맵(feature map)과 필터(filters)를 입력으로 받았으며 URL 주소와 다른 데이터 변수에서 노이즈 데이터 문제를 더 잘 처리하였다.
- 이 모델은 첫 번째 단계에서 원시(raw) URL을 특징 벡터로 변환하기 위해 어휘 특징(Lexical features)을 적용한다.
- 전문가의 어떤 도움 없이도 raw URL 문자열 표현을 직접 학습하는 모델이다.
- 수치형 벡터로 레이블이 없는 데이터를 분석하여 의심스러운 URL을 분석하는 비지도(unsupervised) 학습 모델이다.
- Character-level CNN : 데이터셋에 있는 unique 문자들을 식별하고 각 문자들을 수치형 벡터로 변환함. URL 시퀀스는 행렬 표현으로 변환하고 합성 단계에서 행렬을 계산하여 악의가 포함된 중요한 정보를 식별한다.
- Word-level CNN : 훈련 데이터셋에서 unique 단어를 URL 주소의 특수 문자로 식별함. 합성 단계에서 URL 또는 단어 시퀀스의 행렬 표현을 얻고 함께 나타나는 특정 단어에서 유용한 패턴을 식별한다.
- 각 신경망 계층에서 pooling 개념을 사용하는 것 외에도 SGD(Stochastic Gradient Descent)를 Optim과 함께 적용하였고, 손실 함수를 계산하기 위해서 로컬 변수(Optim library, Gradparams) 개념을 적용하였다.

3) 알고리즘 특징

- 기존 전형적인 CNN 알고리즘(static max-pooling, fixed graph) 방식 대신에 동적 CNN 방식(Dynamic pooling, activation output operation)을 사용하였다.
- 기존 CNN 학습은 filter와 back-propagation 모델을 사용하는 정적 방식이지만, 합성(convolution) 과정에서 동적 계층의 계산 방식을 제안하였다.
- 비선형(Non-linear) URL 주소(문자, 단어)를 학습하는 새로운 딥러닝 프레임워크이다.



〔그림 5〕 딥러닝 프레임워크 기반의 URLDeep 모델〔13〕

- 같은 CNN 채널에서 유사한 단일 시그널(feature maps)을 할당할 수 있고, URLDeep의 그래프는 각 신경망 계층마다 동적으로 업데이트(k-max 개념을 사용한 그래프 재계산 방식) 한다.
- 가장 가까운 이웃의 k-max에서 dynamic pooling을 사용하여 그래프를 다시 계산하게 되면 지속적으로 악성 URL을 예측하는데 효율적이다.
- Dynamic CNN에서 한 모델이 학습되면 동적 할당 방식이 세그먼트별로 입력 신호에 맞게 조정된다.(동적 할당 단계: Data Partition, Channel Fitting)
- 분류 절차의 첫 번째 단계는 특징 표현을 얻는 것이고, 두 번째 단계는 예측 CNN 함수 f : Rn → R을 계산하였다.(여기서, R은 어떤 URL의 점수 예측을 위한 파라미터)

3.1.2. POSTER[14]

1) 학습 데이터

- Ant Financial 사이트로 매일 도착하는 URL 요청 샘플을 데이터셋
- 레이블이 없는 URL들과 소수의 알려진 악성 URL(XXE, XSS, SQL 인젝션 등)
- 매일 요청에서 10억 개의 URL을 샘플링하고 기존 시스템에서 탐지된 악성 URL의 수는 수만~수십만

2) 학습방식/모델유형

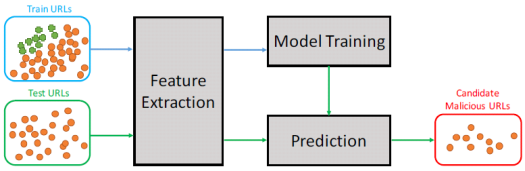
- 잠재적 URL 공격 탐지를 위한 PU learning (Positive and Unlabeled learning) 기반 방식으로 two-stage 전략과 cost-sensitive 전략을 결합하였다.
- 보통 URL은 scheme, authority, path, query, fragment 파트로 구성되어 있고, 처음 몇 군데 파트는 제한되고 공격은 주로 fragment 파트에서 발생한다는데 중점을 두었다. (‘key1=value&..&keyn=value’형식)
- PU learning은 semi-supervised learning의 특수한 경우로 positive이고 레이블이 없는 인스턴스만 제공하고, negative 인스턴스는 제공되지 않는 작업을 처리한다.
- Two-Stage Strategy : 첫 번째 단계에서 레이블이 없는 인스턴스에서 신뢰할 수 있는 negative 인스턴스를 선택하고, 두 번째 단계에서 positive 인스턴스와 신뢰할 수 있는 negative 인스턴스를 선택하여 기존의 감독 모델을 학습하고 새로운 인스턴스를 예측하는데 사용하였다.
- Cost-Sensitive Strategy : 레이블이 없는 인스턴스에는 positive 인스턴스가 거의 없다고 가정하고, 레이블이 없는 모든 인스턴스는 negative 레이블로 추가하였다.
- Prediction : 예측 단계에서 새로운 URL은 먼저 특징 추출 모듈로 전달되어 원래 URL을 (N + 1) * 8 차원 벡터로 전송하고, 추출된 특징 벡터는 상기 두 단계의 모델에 제공하여 각 모델은 URL이 악성일 가능성을 점수로 출력하였다.
- 점수가 높은 URL을 후보 악성 URL 세트로 구성하여 K-URL 세트를 필터링하고, 사이버 보안 엔지니어가 필터링된 URL을 수동으로 확인하여 결과를 검증하였다.
- 연속 7일 동안 수집된 데이터를 사용하여 학습하고, 매일 새로 레이블이 지정되지 않는 URL의 점수를 예측하는데 사용하였다.

3) 알고리즘 특징

- 레이블이 없는 많은 인스턴스와 소수의 알려진 공격 URL들이 포함되어 있는 경우 지도학습을 적용하는데 한계가 있는 PU 학습문제에 대해서 잠재적 URL 공격을 찾는데 효과적인 알고리즘이다.
- 지도학습 방식이라도 과거의 데이터의 결과 부분이 완벽하지 않거나 모호한 경우 정상적으로 적용하기 힘들고 결과도 왜곡되거나 무의미하게 되는 문제를 PU learning으로 해결 가능하다.
- Two-step 전략은 분류된 negative 인스턴스가 없다는 것을 피하기 위해 먼저 신뢰할 수 있는 negative 인스턴스를 발굴한 다음 문제를 전통적인 supervised 또는 semi-supervised learning 문제로 전환하였다.
- Cost-sensitive 전략은 동일하지 않은 오분류 비용을 갖는 이진 분류에 대해 PU 학습 문제를 처리하기 위해 쉽게 이용 가능하다.

scheme:[///[user[:password]@]host[:port]][/path][?query][#fragment]

〔그림 6〕 URL의 일반적인 구문 형식



〔그림 7〕 POSTER 시스템〔14〕

- 상기 두 전략은 예측 모델을 형성하기 위해 결합하였다.

3.1.3. RF-based Analysis[15]

1) 학습데이터

- 특정 안티 바이러스에 의해 표시된 다수의 URL 데이터(1개월 간 수집)
- 한 유형의 위협과 관련된 URL(악성/피싱 캠페인 관련 URL)을 명시적으로 추출하지 않았기 때문에 URL 구조는 부분적으로 관련성이 없다.
- 각 호스트에 동일한 최상위 도메인(.com) 또는 국가코드 2차 도메인(.com)이 포함된 URL만 포함하는 방법의 통계적 유의성을 시험하기 위해 URL의 호스트 부분에서 여러 개의 작은 데이터셋을 생성하였다.(co.uk 등)
- 서로 관련 없는 19개의(대부분 최상위 수준) 도메인의 URL, 호스트 부분을 포함하는 데이터셋, 호스트 부분을 포함하는 큰 데이터셋, 많은 최상위 도메인들의 혼합에서 URL의 경로 부분을 포함하는 큰 데이터셋으로 구성하였다.

2) 학습방식/모델유형

- URL의 어휘 부분만을 기반으로 하는 분류 모델로 정밀도가 높다.
- 두 가지 성능 측정은 정밀도(Precision)-재현율(Recall) 곡선 아래의 면적 계산 방법과 최대 recall 계산 방법을 사용하였다.
- 동일한 레이블이 지정된 URL을 포함하는 데이터셋을 나누고, 다른 수의 트리 T와 최대 깊이 D를 사용하여 몇 개의 RF를 학습하였다.
- 훈련용 데이터 80%, 시험용 데이터 20% 로 구성되었다.

3) 알고리즘 특징

- 사이트 URL의 어휘적 특징만 사용하여 Random-Forest(무작위로 훈련된 결정트리들의 앙상블)로 분류하는 경량의 예측 방법이다.
- 블랙리스트 방식만큼 낮은 오분류 URL 수를 갖는 빠른 분류 방법으로 일반화가 가능하고 더 정밀한 사전 필터링 방법이다.
- 기존 연구에서 다중 알고리즘을 사용하는 방식과 달리 같은 URL 데이터셋에서 서로 다른 유형의 특징들을 비교하는 random-forest 하나의 알고리즘만 사용하였다.
- 많은 수의 특징을 처리하여 확률적인 결과를 제공하고 여러 클래스의 자연 문제들(naturally problem)을 다루었다.
- 사이트 URL의 어휘적 특징에 대한 제한에도 불구하고 URL의 경로 부분에 대해 정밀도-재현율 곡선 아래의 영역에서 인상적인 성능을 보였다.

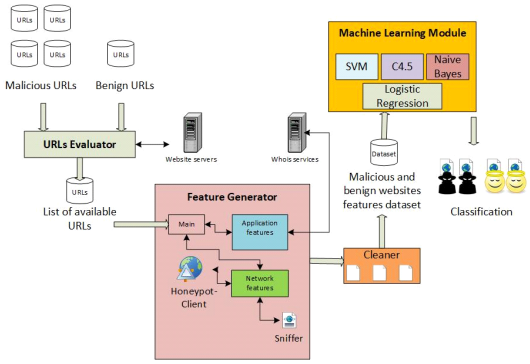
3.1.4. Malicious Websites Detection[16]

1) 학습 데이터

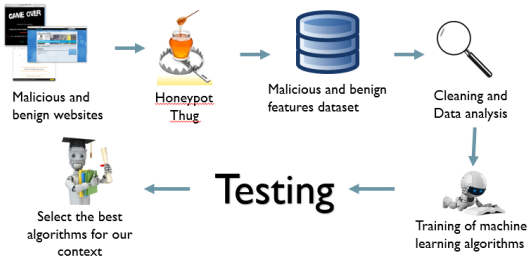
- 3가지 블랙리스트
  - machinelearning.inginf.units.it/data-andtools/hidden-fraudulent-urls-dataset
  - malwaredomainlist.com
  - zeuztacker.abuse.ch
- 약 185,181 개의 URL을 수집하였고, 악성으로 간주하고 다음 연구단계에서 VirusTotal과 같은 보안 도구를 통해 검증할 것을 권장하였다.
- https://github.com/faizann24/Using-machinelearning-to-detect-malicious-URLs.git에서 정상 URL 345,000개를 수집하였고, 이전 단계와 마찬가지로 다른 보안 도구를 통해 검증할 것을 권장하였다.

2) 학습방식/모델유형

- 먼저 각 URL의 정보를 체계적으로 분석하고 생성하기 위해 Python으로 스크립트(HunterThreats.py)를 제작하였다.
- Python의 라이브러리를 통해 각 URL을 사용할 수 있는지 확인했으며 약 530,181 개의 샘플로 시작했지만, 이 단계의 결과로 샘플이 필터링 되어



〔그림 8〕 악성 웹사이트 탐지 프레임워크〔16〕



〔그림 9〕 악성 웹사이트 탐지 프로세스〔16〕

63,191 개의 URL이 생성하였다.

3) 알고리즘 특징

- SVM, J48(C4.5), Naive Bayes, Logistic Regression 알고리즘 사용되었다.

3.1.5. Predict Malicious Websites[17]

1) 학습 데이터

- 상기 3.1.4 연구의 학습 데이터 사용하였다.

2) 학습방식/모델유형

- Numpy, Pandas, Scikit-Learn 패키지를 활용한 기계학습 모델을 사용하였다.

3) 알고리즘 특징

- XGBoost, Neural network, RF/SGD/KNN 등 다양한 기계학습 알고리즘을 사용하여 웹 사이트가 악성인지 정상인지 예측하였다.

```
import numpy as np
import pandas as pd
import xgboost as xgb

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.utils.multiclass import unique_labels

data = pd.read_csv("../input/dataset.csv")

# clean up column names
data.columns = data.columns.\
    str.strip().\
    str.lower()

# remove non-numeric columns
data = data.select_dtypes(['number'])

# split data into training & testing
train, test = train_test_split(data, shuffle=True)

# peek @ dataframe
train.head()
```

〔그림 10〕 XGBoost 구현 소스〔17〕

Ⅳ. 머신러닝 알고리즘 분석

4.1. 알고리즘 선정 및 실험결과 분석

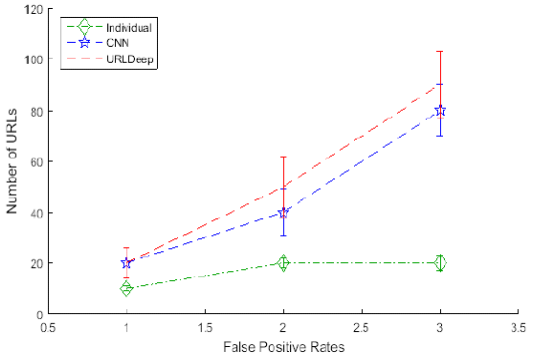
4.1.1. URLDeep

1) 알고리즘 선정

- 레이블이 없는 URL 데이터를 대상으로 전문가의 지식 없이 raw URL 문자열을 표현하고 분석하는 비지도 학습 모델이다.
- 비선형적 URL 주소를 문자와 단어 단위로 분리하고 Dynamic CNN 방식으로 exploit URL의 특징 추출을 통해 악성 URL을 분류하였다.

2) 실험결과 및 분석

- k-max pooling 연산에 의한 dynamic CNN 결과는 URL 주소가 정상인지 악성인지 분류하는데 좋은 정확도가 나왔다.
- 데이터셋을 훈련시키기 위해 평균 뉴런 수를 사용했고 k-max pooling 연산으로 비교적 높은 정확도를 생성할 수 있었다.



〔그림 11〕 URLDeep 실험결과〔13〕

- 본 실험에서는 CNN보다 많은 레이어들과 뉴런 수가 예측 능력을 향상시키는데 관여하지 못하여 컴퓨팅 프로세스에서 리소스를 확대하였다.
- 뉴런 수와 k-max pooling의 제한은 효율적인 신경망을 동적으로 달성하는 효과적은 방법이다.
- 악성 URL 탐지에 대한 잘못된 경고를 평가한 결과에서는 낮은 FPR(False Positive Rates)로 악의적인 리다이렉션의 특성을 탐지하였다.
- 개별 및 CNN 접근 방식은 일부 결과에서만 악성 URL을 탐지했지만, URLDeep 모델은 URL을 악용하기 위한 리다이렉션의 특징을 성공적으로 탐지했으며 FPR 계산 기반으로 URL 시퀀스의 탐지율이 가장 높았다.
- URLDeep은 exploit URL 특징 추출 기반으로 악성 URL을 탐지하고, CNN 접근 방식은 landing URL 기반으로 악성 URL 시퀀스를 탐지하였다.

4.1.2. POSTER

1) 알고리즘 선정

- 레이블이 없는 다수의 URL들과 소수의 알려진 악성 URL들이 포함되어 있는 PU 학습 문제에서도 효과적으로 분석하는 준지도 학습 모델이다.
- 매일 요청되는 다수의 URL들 중에서 일부를 샘플링하고 기존에 탐지된 악성 URL의 수가 소수인 경우에 효율적인 학습 알고리즘이다.

2) 실험결과 및 분석

- 매일 요청되는 레이블 없는 URL에 대해 사이버 보안 엔지니어들이 선택한 URL이 악성인지 아닌지 검사하고 시스템의 효과성을 검증하였다.
- 크기 K의 후보 악성 URL 세트는 최대 150으로 설정하였다.
- 필터링된 후보 세트의 정확도는 90% 이상으로 기존 시스템에서 수집되지 않은 잠재적인 악성 URL을 효과적으로 탐색할 수 있었다.

〔표 2〕 POSTER 실험 결과〔14〕

	Date A	Date B	Date C
# Candidate Ins.	113	103	141
# Malicious Ins.	91	97	130
Accuracy	80.5	94.2	92.2

4.1.3. Random-Forest

1) 알고리즘 선정

- 사이트 URL의 어휘적 특징만 사용하여 무작위로 훈련된 결정트리들의 앙상블로 분류하고 블랙리스트 방식만큼 성능을 갖는 빠른 예측 모델이다.
- 동일한 레이블이 지정된 URL을 포함하는 데이터셋을 나누고, 다른 수의 트리 T와 최대 깊이 D를 사용하여 몇 개의 random-forest를 학습하였다.

2) 실험결과 및 분석

- 정밀도-재현율 곡선 아래의 면적과 0.8 이상의 정밀도를 갖는 분류기의 최대 재현율을 고려한 3가지 특징 추출 방법을 비교한 결과로 forest 크기와 tree 깊이의 4가지 조합을 정리하였다.
- 정밀도-재현율 곡선 아래 면적과 관련하여 방법들 간에 유의미한 차이가 없었다.
- 주어진 정밀도를 가진 분류기의 최대 재현율과 관련하여 forest크기와 tree 깊이의 단일 조합, 무관 기호(don't care symbol)를 가진 4-gram(4GDC)과 4-gram(4G) 사이, 4-gram과 BoW(Bag-of-Word)에 대해서만 중요한 차이가 있었다.
- 큰 데이터셋에서 얻은 결과는 어휘 기능에 대한 제한에도 불구하고, 세 가지 방법 모두 URL의 경로 부분에 대한 정밀도-재현율 곡선 하에서 상당히 인상적이었다.

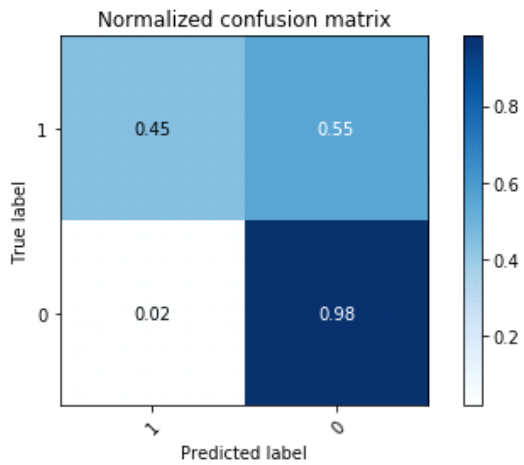
4.1.4. XGBoost

1) 알고리즘 선정

- 동일한 레이블이 지정된 URL을 포함하는 데이터 셋을 나누고, 다른 수의 트리 T와 최대 깊이 D를 사용하여 몇 개의 random-forest를 학습하였다.

2) 실험결과 및 분석

- 정규화된 혼동 행렬(Normalized confusion matrix) 결과에서, 전체 정밀도(Precision)는 96%이고, 전체 정확도(Accuracy)는 72%였다.



〔그림 12〕 XGBoost 실험 결과〔17〕

Ⅴ. 결 론

본 논문에서는 악성코드 유포사이트 탐지를 위한 기존의 연구를 정리하였다. 악성코드 유포사이트 탐지를 위해 다양한 알고리즘을 활용한 탐지 기법들이 제안되고 있으며, 기존의 탐지 기법인 White, Blacklist 기반 탐지, Rule-based 탐지 기법은 새롭게 추가되는 유포사이트 등 악성 URL 대응이 어려운 문제가 있다. 이를 해결하기 위해 많은 연구들에서 기계학습, 다양한 특징 정보 분석 등을 활용하고 있으며, 실험 결과 등을 통해 기존의 탐지 방법과 기계학습 기반의 탐지 방법을 조합하여 좀 더 효율적인 탐지 기법을 찾을 수 있을 것이다. 향후 본 논문에서 분석한 분석 결과와 다양한 악성 URL 특징 정보를 활용하여 악성 URL 탐지 기법을 제안하고자 한다.

참 고 문 헌

[1] Dr.Jitendra Agrawal et al., “Malicious Web Page Detection through Classification Technique: A Survey,” Intl. Journal of Computer Science and Technology, 2017.

[2] 악성코드 유포 탐지기술 현황 조사 및 발전모델 연구, KISA-WP-2015-0061, 2015.

[3] Chia-Mei Chen, et al., “Efficient suspicious URL filtering based on reputation,” Journal of Information Security and Applications, 2015.

[4] Sadia Afroz and Rachel Greenstadt, “PhishZoo: Detecting Phishing Websites By Looking at Them,” Fifth IEEE Intl. conference on Semantic Computing, 2011.

[5] G. Fehringer and P.A. Barraclough, “Intelligent Security for Phishing Online using Adaptive Neuro Fuzzy Systems,” Intl. Journal of Advanced Computer Science and Applications, 2017.

[6] S. Kim, et al., “Malicious URL protection based on attackers’ habitual behavioral analysis,” Computers & Security, 2018.

[7] A.K. Jain and B.B. Gupta, “Phishing Detection: Analysis of Visual Similarity Based Approaches,” Security and Communication Networks, 2017.

[8] R. Verma and A. Das, “What’s in a URL : Fast feature extraction and malicious URL detection,” in ResearchGate, 2017.

[9] Sonika Thakur, et al., “Detection of malicious URLs in big data using RIPPER algorithm,” The 2nd IEEE Intl. Conf. on Recent Trends in Electronics, Information & Communication Technology, 2017.

[10] J. Saxe and K. Berlin, “eXpose: A Character-Level Convolutional Neural Network with Embeddings For Detecting Malicious URLs, File Paths and Registry Keys,” arXiv preprint arXiv:1702.08568, 2017.

[11] L. Hung, et al., “URLNet: Learning a URL representation with deep learning for malicious URL detection,” Cryptography and Security, 2018.

[12] Shibahara et al., “Malicious URL sequence detection using event de-noising convolutional neural network,” IEEE Intl. Conf. on Communications, 2017.

[13] Putra Wanda, “URLDeep: Continuous Prediction of Malicious URL with Dynamic Deep Learning in Social Networks,” Int. Journal of Network Security 22(4), March 2019.

[14] Ya-Lin Zhang et al., “POSTER : A PU Learning based System for Potential Malicious URL Detection,” CCS ‘17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017.

[15] J. Puchyi and M. Holena, “Random-Forest-Based Analysis of URL Paths,” CEUR Workshop Proceedings, 2017.

[16] Christian Urcuqui, “Malicious and Benign Websites”, https://www.kaggle.com/xwolf12/malicious-and-benign-websites

[17] Chelsea Raerek, “Predict Malicious Websites: XG Boost”, https://www.kaggle.com/craerek/predict-malicious-websites-xgboost/data

〈저자소개〉

오 성 택 (Sungtaek Oh)

정회원

2013년 2월 : 아주대학교 정보컴퓨터공학부 졸업

2016년 2월 : 아주대학교 컴퓨터공학과 석사

2015년 12월~현재 : 한국인터넷진흥원 침해대응기술팀 선임연구원

<관심분야> 인공지능, 사물인터넷, 정보보호

신 삼 신 (Sam-Shin Shin)

정회원

2006년 : 전남대학교 멀티미디어학과 졸업

2008년 : 전남대학교 일반대학원 정보보호협동과정 석사

2008년~현재 : 한국인터넷진흥원 침해대응기술팀 수석연구원

<관심분야> 머신러닝, 정보보호, 데이터사이언스