Journal of Convergence for Information Technology
Vol. 11. No. 5, pp. 30-37, 2021
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2021.11.05.030

뉴럴네트워크 기반에 악성 URL 탐지방법 설계

권 현, 박상준, 김용철*
육군사관학교 전자공학과 교수

Design of detection method for malicious URL based on Deep Neural Network

Hyun Kwon, Sangjun Park, Yongchul Kim*
Professor, Department of Electrical Engineering, Korea Military Academy

요 약 사물인터넷 등을 통하여 각종 기기들이 인터넷으로 연결되어 있고 이로 인하여 인터넷을 이용한 공격이 발생하고 있다. 그러한 공격 중 악성 URL를 이용하여 사용자에게 잘못된 피싱 사이트로 접속하게 하거나 악성 바이러스를 유포하는 공격들이 있다. 이러한 악성 URL 공격을 탐지하는 방법은 중요한 보안 이슈 중에 하나이다. 최근 딥러닝 기술 중 뉴럴네트워크는 이미지 인식, 음성 인식, 패턴 인식 등에 좋은 성능을 보여주고 있고 이러한 뉴럴네트워크를 이용하여 악성 URL 탐지하는 분야가 연구되고 있다. 본 논문에서는 뉴럴네트워크를 이용한 악성 URL 탐지 성능을 각 파라미터 및 구조에 따라서 성능을 분석하였다. 뉴럴네트워크의 활성화함수, 학습률, 뉴럴네트워크 모델 등 다양한 요소들에 따른 악성 URL 탐지 성능에 어떠한 영향을 미치는 지 분석하였다. 실험 데이터는 Alexa top 1 million과 Whois에서 크롤링하여 데이터를 구축하였고 머신러닝 라이브러리는 텐서플로우를 사용하였다. 실험결과로 층의 개수가 4개이고 학습률이 0.005이고 각 층마다 노드의 개수가 100개 일 때, 97.8%의 accuracy와 92.94%의 f1 score를 갖는 것을 볼 수 있었다.

주제어 : 악성 URL, 머신 러닝, 탐지 방법, 뉴럴네트워크, 패턴인식

Abstract Various devices are connected to the Internet, and attacks using the Internet are occurring. Among such attacks, there are attacks that use malicious URLs to make users access to wrong phishing sites or distribute malicious viruses. Therefore, how to detect such malicious URL attacks is one of the important security issues. Among recent deep learning technologies, neural networks are showing good performance in image recognition, speech recognition, and pattern recognition. This neural network can be applied to research that analyzes and detects patterns of malicious URL characteristics. In this paper, performance analysis according to various parameters was performed on a method of detecting malicious URLs using neural networks. In this paper, malicious URL detection performance was analyzed while changing the activation function, learning rate, and neural network structure. The experimental data was crawled by Alexa top 1 million and Whois to build the data, and the machine learning library used TensorFlow. As a result of the experiment, when the number of layers is 4, the learning rate is 0.005, and the number of nodes in each layer is 100, the accuracy of 97.8% and the f1 score of 92.94% are obtained.

Key Words : Malicious URL, Machine learning, Detection method, Neural network, Pattern recognition

This work was supported by 2021 (21-center-2) research fund of Korea Military Academy (Cyber Warfare Research Center).
*Corresponding Author : Yongchul Kim
Received April 15, 2021 Revised May 3, 2021
Accepted May 20, 2021 Published May 28, 2021

1. 서론

사물인터넷 등을 통해서 각 장비들이 인터넷으로 연결이 되어있다. 따라서 인터넷 상에 보안이 중요 시 되고 있고 다양한 사이버상 공격이 발생 되고 있다. 이러한 사이버상 공격 중에 대표적으로 악의적인 웹 도메인 또는 악성 URL 기반 공격 [1,2]이 있다. 이러한 악성 URL을 이용한 공격은 사용자에게 피싱 사이트처럼 악의적인 URL을 이용해서 공격자가 의도한 특정 사이트로 연결시켜 바이러스나 악성 공격을 하는 방법이다. 이러한 URL 공격방법은 DNS 시스템[3]을 타겟팅 한다. DNS 시스템은 사람이 사용하기 편하게 영어로 기입을 하지만 각 주소마다 실제 IP address로 매핑 시켜주는 역할을 하는 시스템이다. 그러나 이러한 DNS 시스템은 악성 URL를 이용한 IP 주소로 바꾸어 특정 도메인으로 이동시키는 다양한 공격을 받는다. 따라서 악의적인 URL를 미리 선제적으로 식별하는 능력이 DNS 시스템에서 중요한 보안문제 중에 하나이다.

선제적으로 악의적인 URL를 식별하기 위한 일반적인 방법은 도메인의 네임과 네트워크의 주소 정보를 사전에 모으는 것이다. 사전에 알려진 도메인의 네임과 네트워크 주소 정보를 각 features마다 추출해서 ML 알고리즘을 이용하여 딥러닝 모델에 상당히 많은 양의 데이터를 학습한다. 이렇게 미리 학습한 딥러닝 모델을 이용하여 악의적인 도메인이 올 경우 탐지하는 방법을 사용할 수 있다.

이 논문에서 뉴럴네트워크[4]를 이용하여 악성 URL를 탐지하는 방법에 대하여 연구하였다. 본 논문의 목적은 뉴럴네트워크의 파라미터와 구조에 따른 악성 URL 탐지 성능을 체계적으로 분석하는 것에 있다. 실제 이용되는 악성 URL에 대한 features에 대하여 뉴럴네트워크 모델을 이용하여 학습률, 노드수, 층수, 최적화 알고리즘, 활성화 함수 등의 파라미터를 조정하여 실험적으로 성능 분석을 하였다. 또한, 추가적으로 악성 URL를 탐지하기 위하여 features에 대한 주요 특징을 정리하였다.

이 논문의 나머지 구성은 다음과 같다. 2장에서는 악성 URL의 관련 연구에 대한 소개를 하였고 3장에서는 악성 URL 탐지를 하기 위한 뉴럴네트워크 구조를 설명하였다. 4장에서는 악성 URL과 관련된 실험 구성 및 실험결과 분석을 하였고 5장에서는 이 논문의 고찰로 구성하였다. 마지막으로 6장에서는 이 논문의 결론을 구성하였다.

2. 관련연구

악의적인 URL을 탐지하는 이슈는 사이버 보안에 있어서 중요한 문제 중에 하나이다. 이 장에서 악의적인 도메인을 탐지하는 방법은 수학적인 이론 접근방법과 머신러닝을 이용한 접근방법으로 나눠서 설명하고자 한다.

2.1 수학적 이론을 이용한 접근방법

악성 URL를 식별하기 위하여 수학적인 이론 접근방법은 그래프 이론방법을 주로 사용해왔다.

Yadav et al. 연구진[5]은 다양한 악의적인 URL에서 fast flux를 이용한 악의적인 도메인으로 연결하는 공격에 대한 방어하는 방법을 제안하였다. fast flux 공격 방법은 보트넷에 기반한 DNS 기술로 피싱한 내역을 숨기고 다양한 악의적인 사이트로 주소를 바꿔가면서 프록시 서버 역할을 한다.

이러한 fast flux 공격 방법에 대응하여 Yadav et al. 연구진은 DNS 도메인에 나오는 쿼리(quires)를 분석하고 반응을 통해 악성 URL를 탐지하는 알고리즘을 제안하였다. 이 알고리즘은 알파벳에 사용되는 패턴에 대하여 통계적 방법을 이용하여 악성 URL를 계산한다. Dolberg et al. 연구진[6]은 multi-dimensional aggregation 모니터링 방법을 이용하여 악의적인 URL를 탐지하는 방법을 제안하였다. 이 방어방법에서는 도메인 네임과 IP 주소 간의 시간상 사용되는 관계성을 tree 구조로 구성하여 악의적인 URL를 탐지하는 방법을 적용하였다.

이와 같이 두 개의 연구들은 악의적인 URL에 대한 패턴이나 관계성을 이용한 관계 그래프를 이용하여 악성 URL을 탐지하는 방법을 제안하였다. 하지만 새로운 악성 URL의 경우 탐지율이 떨어지는 단점이 있다.

2.2 머신러닝을 이용한 접근방법

머신러닝을 이용한 악성 URL을 탐지하는 방법은 악의적인 URL에 대한 features를 추출하고 이에 대한 악성 URL인지 정상적인 URL인지 라벨을 붙인다. 그 이후에 많은 데이터를 학습하고 특정 URL이 입력으로 왔을 때, 머신러닝 모델이 이 URL이 악성인지 정상인지 분류한다.

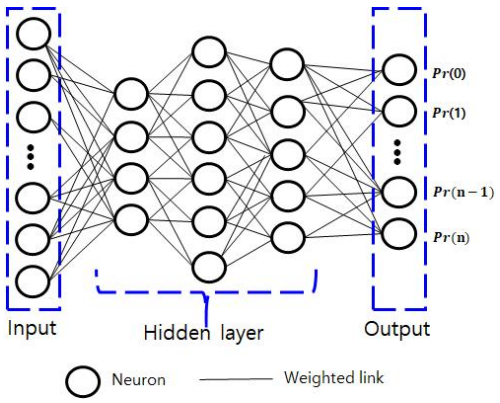

Fig. 1. Overview of the neural network

Shi et al. 연구진[7]은 악의적인 URL을 탐지하는 방법으로써, extreme learning machine (ELM)을 제안하였다. ELM은 features를 4가지로 구분하여 construction-based, IP-based, TTL-based, WHOIS-base로 각각 탐지한다. 이 방법은 높은 정확도를 가지면서 빠른 학습속도를 갖는 장점이 있다. 그러나 이 방법은 features의 fraction 현상으로 인하여 성능이 저하 되어 비효율적인 측면이 나타난다. Sun et al. 연구진[8]은 HinDom 방법을 제안하였다. 이 방법에서는 DNS와 pDNS 데이터를 수집하여서 다양한 클라이언트에 대한 정보를 수집한 후에 heterogeneous graph를 형성하여 악성 URL에 방어하는 방법을 제안하였다. Bilge et al 연구진[9]은 malicious domain names를 탐지하는 방법으로써, Exposure 방법을 제안하였다. 이 연구에서는 DNS 데이터에서 특정 시간동안 알려진 악성 URL과 정상적인 URL의 features 들을 추출한다. 추가적으로 passive DNS을 수집하는 데, passivie DNS는 DNS에서 패킷을 보내고 응답되는 정보를 토대로 저장이 된다. passive DNS에서 DNS 기반 features와 DNS의 응답시간 기반의 features, TTL 기반의 features, Domain name에 관련된 features로 구성하였다. 이러한 Exposure 방법을 통해서 멀웨어나 스팸과 같은 악의적인 정보를 탐지하는데 성능이 좋다. Rahbarinia et al 연구진[10]은 악성 URL 탐지 기법을 passive DNS의 트래픽에서 멀웨어로 알려진 악성 URL의 관계성을 통해서 탐지하는 방법을 제안하였다. 이 탐지방법은 멀웨어 컨트롤 도메인을 bipartite 그래프로 생성하여 알려진 악성 도메인과 일반 도메인간의 관계성을 기초로 멀웨어 도메인을 식별한다. 이 방법은 features 들로 Machine Behavior, Domain Activity, IP Abuse를 선정하여 탐지하는 데 사용하였다. 최근에 J. Yuan, G et al. [11] 연구진에 의해서 병렬적인 뉴럴 모델 알고리즘을 이용하여 악성 URL를 탐지하는 방법을 제안하였다. 이 방법은 URL 성분을 lexical embedding vector로 변환하여 capsule network과 independent recurrent neural network를 혼합하여 악성 URL 탐지하는 방법을 제안하였다. 또한, Patgiri, Ripon et al[12] 연구진은 bloom filtering과 머신러닝 기법을 적용하여 악성 URL 탐지하는 방법을 제안하였다. 이 방법에서 암호기법이 아닌 해쉬 함수에서 다양한 필터를 이용하여 악성 URL을 탐지하는 방법을 제안하였다. 이외에도 룰 기반의 머신러닝 기법[13]을 이용한 악성 URL 탐지방법들도 소개되고 있다.

3. 뉴럴네트워크를 이용한 악성 URL 탐지

뉴럴네트워크[4]는 사람의 신경구조를 모방하여 만든 수학적 모델이다. Fig. 1과 같이, 이 구조는 노드와 가중치의 연결로 된 여러 개의 layer로 되어 있으며 입력층(input layer), 은밀층(hidden layer), 결과층(output layer)으로 구성되어 있다.

먼저, 입력층은 여러개의 노드로 구성되어 입력값에 해당되는 내용을 1:1로 매칭하여 받는다. 받은 후에는 다음 은밀층에 처음 시작하는 층에 각 노드마다의 가중치를 곱한 값에 활성화 함수(Activation function)[14]를 이용한다. 활성화 함수는 음수이면 제거하거나 양수이면 그대로 두는 LeRU 등을 이용하여 은밀층의 첫 번째 층에 전달 한다. 은밀층에서는 여러개의 층이 존재한다. 또한, 각 층마다 여러개 노드와 가중치들이 연계되어 있다. 이러한 상황에서 각 노드마다 여러 가중치에 대하여 곱셈을 하고 이를 더한 값을 활성화 함수을 거쳐서 순차적으로 다음 층에게 결과값을 전달한다. multi-label의 경우, 결과층은 은밀층에서 나온 값을 softmax 층을 이용하여 각 class 마다의 확률값을 제공한다. softmax 층을 이용하면 어떤 특정 input 값에 대한 뉴럴네트워크에 각 class마다의 확률값이 제공이 되고 이 중에 가장 높은 확률값으로 뉴럴네트워크는 입력값에 대한 예측값으로 제공한다. 반면에, two-label (binary classifier)의 경우, 결과층은 은밀층에서 나온 값을 sigmoid를 이용하여 0 또는 1

둘 중 한가지 값인 binary value를 제공한다. 이 경우, Yes 또는 No와 같이 특정 값에 대한 이진값만 제공한다. 이 논문에서는 binary classifier로 0 또는 1로 악성 URL 인지 정상 URL 인지 분별하는 모델이다.

뉴럴네트워크가 학습하는 과정에서는 입력값에 대한 예측값과 실제 class와의 일치성을 보면서 cross entropy 손실함수[15]와 gradient descent 방법[16] 등을 이용하여 뉴럴네트워크에 각 노드와 가중치 파라미터를 최적화 시킨다. 그렇게 뉴럴네트워크가 학습이 다 된 이후에, 새로운 테스트가 뉴럴네트워크에 제공되었을 때 신뢰성 있는 예측값을 제공해준다.

4. 실험환경 및 실험결과

악성 URL 탐지에 대한 성능을 보이기 위해, 실험환경은 텐서플로우 머신러닝 라이브러리[17]를 사용하였으며, 서버는 Intel(R) Core(TM) i3-7100 CPU @ 3.90GHz와 GPU는 GeForce GTX 1050을 사용하였다.

4.1 데이터 구축

뉴럴네트워크 연구분야에서 중요한 부분은 데이터를 구축하는 부분이다. 이 논문에서 데이터를 구축할 때, Alexa top 1 million[18]과 Whois, DNS records[19]를 통해서 악의적인 URL과 정상적인 URL 둘 다 포함된 데이터를 크롤링하여 구축하였다. Whois는 인터넷 정보 리스트를 기록하여 저장하는 서비스로 도메인명, IP, 네임서버의 상세 등록 정보, 위치 평균 시간 (Time to Live, TLL)에 대한 내용을 알 수 있다. 이러한 Whois에서 제공되는 정보는 웹사이트에서 무결성을 유지하는 데 중요한 역할을 한다. 총 URL 데이터는 13,515개이며 이 중에 악성 URL은 2088개이며, 정상 URL은 11,427개 이다. 이 중에서 총 데이터의 75%는 훈련 데이터로 사용하였고 총 데이터의 25%는 테스트 데이터로 사용하여 뉴럴네트워크에 의한 악성 URL의 탐지성능을 분석하였다.

4.2 데이터 특징

뉴럴네트워크에 적용하기 위한 데이터의 특징은 13가지로 도메인 길이, 연속적인 단어의 수, 도메인의 엔트로피, IP 주소, IP의 지리적인 위치정보, TLL 평균 시간, TTL 표준편차, 도메인 유지시간, 도메인 활동시간, 접속된 나라별 순위, 접속된 URL에 대하여 관련 나라별 순위, Passive DNS 변화 빈도수, SSL 여분 시간으로 선정하였다. 이러한 추출된 데이터 특징들은 [20] 논문에서 나온 내용을 토대로 선정하였다.

4.3 뉴럴네트워크의 구성 및 파라미터

뉴럴네트워크 모델은 Table 1과 Table 2와 같이, 기본적으로 input layer 한 개와 3개의 hidden layers, 1개의 output layer로 구성하였다. 활성화 함수로는 ReLU, leaky-ReLU를 사용하였고 각 layer마다 ReLU, leakyReLU를 각각 달리해서 실험을 하였다. 배치사이즈는 150으로 하였고 학습률은 0.01, 0.05, 0.1, 0.2 단위로 각각에 대한 평가를 하였다. 최적화 알고리즘은 Adam[21]과 Stochastic gradient descent (SGD) 알고리즘[22]을 사용하여 악성 URL 탐지성능을 분석하였다.

Table 1. The architecture for the neural network

Description	Value
Fully connected layer + ReLU or Leaky ReLU	80
Fully connected layer + ReLU or Leaky ReLU	80
Fully connected layer + ReLU or Leaky ReLU	80
Fully connected layer + Sigmoid function	1

Table 2. The parameter for the neural network

Description	Value
Optimizer	Adam, SGD
Learning rate	0.001 / 0.005 / 0.01 / 0.015 / 0.02
Batch size	150
Epochs	100

4.4 실험결과

Fig. 2는 뉴럴네트워크의 학습률을 0.001, 0.005, 0.01, 0.015, 0.02로 조정하였을 때의 악성 URL 탐지 성능을 보여준다. 그림에서 보면 학습률에 따라 accuracy, precision, recall, f1 socre 점수가 다른 것을 볼 수 있다. 전체적으로 봤을 때, 학습률이 증가할수록 accuracy, recall, f1 score 성능이 전체적으로 저하되는 것을 볼 수 있다. f1 score 측면이나 accuracy를 봤을 때, 학습률이 0.001이거나 0.005일 때, 좋은 성능을 가진 것을 볼 수 있다.

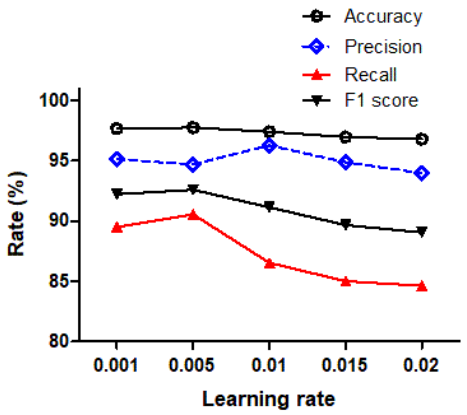

Fig. 2. Detection performance according to the learning rate

Fig. 3는 뉴럴네트워크의 은밀층의 수에 따른 성능을 보여준다. 노드의 수는 각 층마다 80개로 설정하였고 최적화알고리즘은 Adam과 활성화 함수는 ReLU로 하였다. 그림에서 보면 layer가 증가할수록 전체적으로 성능이 저하되는 것을 볼 수 있다. 이는 층이 많아질수록 vanishing gradient 현상으로 역전파 학습과정에서 학습이 잘 안되는 현상이 발생하였기 때문이다. 하지만 경우에 따라 precision 측면이 중요할 때에는 은밀층의 수를 증가시킬수록 성능이 좋은 것을 볼수 있다. 반면에 Recall 성능이 요구될 때에는 은밀층의 수를 4개나 5개로 적정수준 유지한 것이 좋은 성능을 가지는 것을 볼 수 있다.

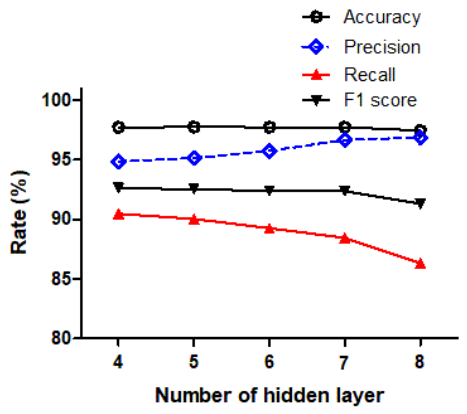

Fig. 3. Detection performance according to the number of hidden layer

Fig. 4는 뉴럴네트워크에서 각 층의 노드 수를 20, 40, 60, 80, 100개로 설정하였을 때의 악성 URL 탐지 성능을 보여준다. 총 층의 개수는 4개로 하였고 최적화 알고리즘은 Adam과 활성화 함수는 ReLU로 하였다. 그림에서, 각 층마다 노드의 수가 증가할수록 성능이 증가하는 것을 볼 수 있다. 하지만 각 노드에 따라 precision, recall 성능이 각기 다른 측면이 있고 accuracy와 f1 score는 거의 비슷하거나 노드의 수에 따라 증가하는 것을 볼 수 있다.

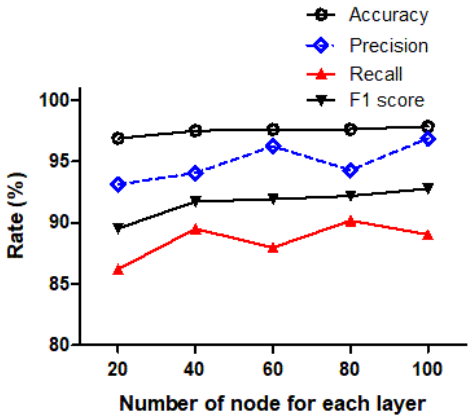

Fig. 4. Detection performance according to the number of node for each layer

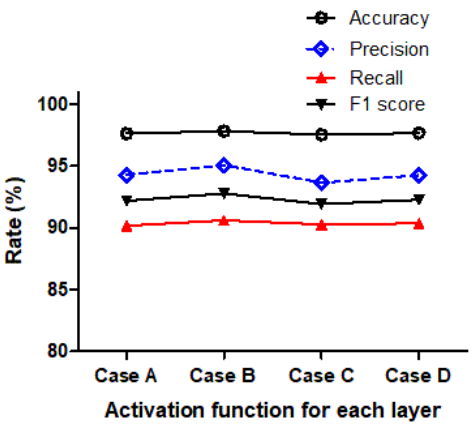

Fig. 5. Detection performance according to the activation function for each layer

Fig. 5는 각 층마다 활성화 함수에 따른 악성 URL 탐지 성능을 보여준다. 총 층의 개수는 4개와 80개 노드로 하였고 최적화알고리즘은 Adam으로 하였다. 그

림에서 Case A는 ReLu+ReLu+leaky ReLu, Case B는 leaky ReLu+leaky ReLu+leaky ReLu, Case C는 ReLu+ReLu_ReLu, Case D는 ReLu+leaky ReLu+leaky ReLu 이다. 그림에서 보면 활성화함수에서 ReLU와 leaky-ReLU 조합에서 거의 성능이 비슷한 것을 볼 수가 있다. 가장 최적의 조합을 봤을 때, Case B의 경우가 나머지 경우보다 좀 더 좋은 성능을 가지는 것을 볼 수 있다.

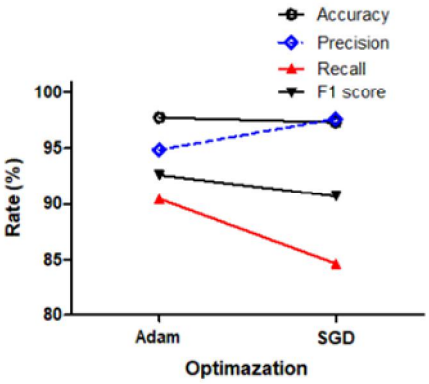

Fig. 6. Detection performance according to the optimization

Fig. 6은 뉴럴네트워크의 최적화 알고리즘에 따른 악성 URL 탐지 성능을 보여준다. 최적화 알고리즘으로 Adam과 Stochastic gradient descent (SGD)를 사용하였다. 그림에서 보면 Adam 알고리즘이 SGD 방법보다 좀 더 좋은 성능을 가지는 것을 볼 수가 있었다.

5. 고찰

악성 URL를 탐지하기 위하여 뉴럴네트워크를 구성할 때, 뉴럴네트워크의 파라미터 설정에 따른 성능 분석을 할 필요성이 있다. 왜냐하면 파라미터에 따라 뉴럴네트워크의 성능이 다르기 때문에 적절한 파라미터를 선정하는 것이 매우 중요하다. 이 장에서는 각 파라미터들에 의한 성능 분석을 정리하였다.

뉴럴네트워크를 구성할 때, 학습률, 노드의 수, 층의 수, 활성화함수, 최적화 알고리즘 등의 여러 파라미터를 이용하여 악성 URL 탐지성능을 분석하였다. 최적화 알고리즘 측면에서 SGD 알고리즘보다는 Adam 알고리즘이 전체적으로 성능이 좋은 것을 볼 수가 있었다. 학습률 측면에서 0.001과 0.005로 작은 학습률일 때 좀 더 좋은 성능을 가지는 것을 볼 수가 있었다. 노드의 수는 많을수록 전체적으로 성능이 개선이 되었지만 반면에 층 수가 많을수록 오히려 성능이 저하되는 현상이 있었다.

성능 측정 기준에서 보면, accuracy, precision, recall, f1 score를 통해서 4가지 기준으로 분석을 하였다. 각 성능마다 특징이 있기 때문에 어느 성능이 우선이 되는 지 고려될 필요가 있다. accuracy는 true 또는 false로 실제 예측한 값과 실제의 true와 false 값과의 일치율을 의미한다. precision은 true라고 분류한 것 중에 실제 true인 비율을 의미하고 recall은 실제 true인 것 중에서 모델이 true라고 예측한 비율이다. 예를 들어, precision은 웹사이트에서 사람의 얼굴을 추출하여 데이터를 저장하는 경우, 예측한 값이 실제 true와 일치한 게 더 중요하기 때문에 이때에는 precision이 중요하다. 반면에 recall은 악성 코드나 범죄 용의자의 얼굴을 탐지하는 데 true인 것을 true로 제대로 예측하는 것이 중요하기 때문에 누락이 없도록 하는 recall이 더 중요하다. 따라서 성능 요구하는 기준에 따라 어떤 기준을 중요시할 지는 사용자 입장에서 고려해야한다.

6. 결론

이 논문에서는 뉴럴네트워크를 이용한 악성 URL 탐지시 각 파라미터들이 성능에 미치는 영향을 분석하였다. 학습률, 노드의 수, 층의 개수, 최적화 알고리즘, 활성화 함수 등의 파라미터를 조정하여 악성 URL 탐지성능을 분석하였다. 분석 결과를 살펴보면 층의 개수, 학습률이 증가할수록 성능이 오히려 저하되는 것을 볼 수 있었다. 반면에 최적화 알고리즘을 Adam으로 사용하였을 때 좀 더 좋은 성능을 가지는 것을 볼 수가 있었다. 층의 개수가 4개이고 학습률이 0.005이고 각 층마다 노드의 개수가 100개 일 때, 97.8%의 accuracy와 92.94%의 f1 score를 갖는 것을 볼 수 있었다.

향후 연구로는 다른 악성 URL 데이터셋 등을 확장하여 실험을 할 수가 있다. 또한, Generative adversarial network[23]이나 Autoencoder 방법[24]을 이용하여 악의적인 URL 탐지하는 연구도 진행할 예정이다. 또한, 적대적 샘플방법[25][26][27]들을 이용하여 악성 URL 탐지기법을 회피하는 방법에 대한 공격연구도 흥미로운 주제가 될 것 이다.

REFERENCES

[1] P. Zhao & S. C. Hoi. (2013, August). Cost-sensitive online active learning with application to malicious URL detection. *In Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 919-927). DOI : 10.1145/2487575.2487647

[2] F. Yu. (2015). Malicious url detection algorithm based on bm pattern matching. *International Journal of Security and Its Applications, 9(9),* 33-44.

[3] J. Klensin. (2003). Role of the domain name system (dns). *Internet Request for Comments: RFC, 3467.*

[4] M. Anthony & P. L. Bartlett. (2009). *Neural network learning: Theoretical foundations*. cambridge university press.

[5] S. Yadav, A. K. K. Reddy, A. N. Reddy & S. Ranjan. (2012). Detecting algorithmically generated domain-flux attacks with DNS traffic analysis. *IEEE/Acm Transactions on Networking, 20(5),* 1663-1677. DOI : 10.1109/TNET.2012.2184552

[6] L. Dolberg, J. François & T. Engel. (2012). Efficient multidimensional aggregation for large scale monitoring. *In 26th Large Installation System Administration Conference ({LISA} 12)* (pp. 163-180).

[7] Y. Shi, G. Chen & J. Li. (2018). Malicious domain name detection based on extreme machine learning. *Neural Processing Letters, 48(3),* 1347-1357. DOI : 10.1007/s11063-017-9666-7

[8] X. Sun, M. Tong, J. Yang, L. Xinran & L. Heng. (2019). Hindom: A robust malicious domain detection system based on heterogeneous information network with transductive classification. *In 22nd International Symposium on Research in Attacks, Intrusions and Defenses ({RAID} 2019)* (pp. 399-412).

[9] L. Bilge, S. Sen, D. Balzarotti, E. Kirda & C. Kruegel. (2014). Exposure: A passive dns analysis service to detect and report malicious domains. *ACM Transactions on Information and System Security (TISSEC), 16(4),* 1-28. DOI : 10.1145/2584679

[10] B. Rahbarinia, R. Perdisci & M. Antonakakis. (2016). Efficient and accurate behavior-based tracking of malware-control domains in large ISP networks. *ACM Transactions on Privacy and Security (TOPS), 19(2),* 1-31. DOI : 10.1145/2960409

[11] J. Yuan, G. Chen, S. Tian & X. Pei. (2021). Malicious URL Detection Based on a Parallel Neural Joint Model. *IEEE Access, 9,* 9464-9472. DOI : 10.1109/ACCESS.2021.3049625.

[12] R. Patgiri, A. Biswas & S. Nayak. (2021). deepBF: Malicious URL detection using Learned Bloom Filter and Evolutionary Deep Learning. *arXiv preprint arXiv:2103.12544.*

[13] B. M. Kim, Y. W. Han, G. Y. Kim, Y. B. Kim & H. J. Kim. (2020). Development of Rule-Based Malicious URL Detection Library Considering User Experiences. *Journal of the Korea Institute of Information Security & Cryptology, 30(3),* 481-491. DOI : 10.13089/JKIISC.2020.30.3.481

[14] D. F. Specht. (1990). Probabilistic neural networks. *Neural networks, 3(1),* 109-118.

[15] D. M. Kline & V. L. Berardi. (2005). Revisiting squared-error and cross-entropy functions for training neural network classifiers. *Neural Computing & Applications, 14(4),* 310-318. DOI : 10.1007/s00521-005-0467-y

[16] S. Du et al. (2019, May). Gradient descent finds global minima of deep neural networks. *In International Conference on Machine Learning* (pp. 1675-1685). PMLR.

[17] M. Abadi et al. (2016). Tensorflow: A system for large-scale machine learning. *In 12th {USENIX} symposium on operating systems design and implementation* ({OSDI} 16) (pp. 265-283).

[18] https://www.alexa.com

[19] https://gnso.icann.org

[20] N. Hason, A. Dvir & C. Hajaj. (2020, July). Robust Malicious Domain Detection. *In International Symposium on Cyber Security Cryptography and Machine Learning* (pp. 45-61). Springer, Cham. DOI : 10.1007/978-3-030-49785-9_4

[21] D. P. Kingma & J. Ba. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[22] L. Bottou. (2010). Large-scale machine learning with stochastic gradient descent. *In Proceedings of COMPSTAT'2010* (pp. 177-186). Physica-Verlag HD. DOI : 10.1007/978-3-7908-2604-3_16

[23] A. Creswell et al. (2018). Generative adversarial networks: An overview. *IEEE Signal Processing Magazine, 35(1),* 53-65.
DOI : 10.1109/MSP.2017.2765202

[24] E. Kodirov, T. Xiang & S. Gong. (2017). Semantic autoencoder for zero-shot learning. *In Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3174-3183).

[25] H. Kwon, H. Yoon & D. Choi. (2019). Restricted evasion attack: Generation of restricted-area adversarial example. *IEEE Access, 7,* 60908-60919.
DOI : 10.1109/ACCESS.2019.2915971

[26] H. Kwon, Y. Kim, H. Yoon & D. Choi. (2018). Random untargeted adversarial example on deep neural network. *Symmetry, 10(12),* 738.
DOI : 10.3390/sym10120738

[27] H. Kwon, H. Yoon & K. W. Park. (2019, November). POSTER: Detecting audio adversarial example through audio modification. *In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (pp. 2521-2523).
DOI : 10.1145/3319535.3363246

[28] H. Kwon, Y. Kim, K. W. Park, H. Yoon & D. Choi. (2018). Advanced ensemble adversarial example on unknown deep neural network classifiers. *IEICE TRANSACTIONS on Information and Systems, 101(10),* 2485-2500.
DOI : 10.1587/transinf.2018EDP7073

[29] H. Kwon, H. Yoon & K. W. Park. (2020). Acoustic-decoy: Detection of adversarial examples through audio modification on speech recognition system. *Neurocomputing, 417,* 357-370.
DOI : 10.1016/j.neucom.2020.07.101

권 현 (Hyun Kwon) [정회원]

· 2010년 2월 : 육군사관학교 수학과 학사 졸업
· 2015년 8월 : 한국과학기술원 전산학부 석사 졸업
· 2020년 2월 : 한국과학기술원 전산학부 박사 졸업
· 2020년 2월 ~ 현재 : 육군사관학교 전자공학과 교수
· 관심분야 : 인공지능 보안, 시스템 보안, 머신러닝
· E-Mail : hkwon.cs@gmail.com

박 상 준 (Sangjun Park) [정회원]

· 2000년 2월 : 육군사관학교 독일어 학사 졸업
· 2010년 2월 : 한국과학기술원 정보통신공학 석사 졸업
· 2020년 3월 ~ 현재 : 아주대학교 박사과정
· 2019년 11월 ~ 현재 : 육군사관학교 전자공학과 교수
· 관심분야 : C4I체계, 정보통신공학
· E-Mail : sigpsj13438@naver.com

김 용 철 (Yongchul Kim) [정회원]

· 1998년 2월 : 육군사관학교 전자공학 학사 졸업
· 2001년 11월 : University of Surrey 전자공학과 석사 졸업
· 2012년 1월 : North Carolina State University 전자공학과 박사 졸업
· 2012년 1월 ~ 현재 : 육군사관학교 전자공학과 교수
· 관심분야 : 무선통신네트워크, 통신공학. 전자공학
· E-Mail : kyc6454@kma.ac.kr