https://doi.org/10.33778/kcsa.2019.19.4.181

# 학습 데이터 개선을 통한 Anomaly-based IDS의 성능 향상 방안

문 상 태*, 이 수 진**


## 요 약

최근 Anomaly 기반 침입탐지시스템에서의 탐지 기준점 생성을 위해 인공지능 기술을 적용하려는 시도가 활발하게 진행되고 있다. 그러나 인공지능 기술의 적용을 제안한 기존 연구들은 대부분 인공 신경망의 구조 개선과 최적의 하이퍼파라미터 값을 찾는데 중점을 두고 있으며, 학습 데이터의 잘못된 구성으로 인해 발생할 수 있는 다양한 문제점들은 해결하지 못하고 있다. 이에 본 논문에서는 학습 데이터의 잘못된 구성으로 인해 나타날 수 있는 주요 문제점을 실험을 통해 식별하고 학습 데이터의 재구성을 통해 그러한 문제점을 개선함으로써 침입탐지 성능을 향상시킬 수 있는 방안을 제안한다.


# A Study on the Performance Improvement of Anomaly-Based IDS Through the Improvement of Training Data

Sang Tae Moon*, Soo Jin Lee **


## ABSTRACT

Recently, attempts to apply artificial intelligence technology to create the normal profile in Anomaly-based intrusion detection systems have been made actively. But existing studies that proposed the application of artificial intelligence technology mostly focus on improving the structure of artificial neural networks and finding optimal hyper-parameter values, and fail to address various problems that may arise from the misconfiguration of learning data. In this paper, we identify the main problems that may arise due to the misconfiguration of learning data through experiment. And we also propose a novel approach that can address such problems and improve the detection performance through reconstruction of learning data.


**Key words : Anomaly based Intrusion Detection System, Artificial Neural Network, Machine learning**

접수일(2019년 9월 5일), 게재확정일(2019년 9월 21일)

* 국방대학교 국방과학학과(주저자)

** 국방대학교 국방과학학과(교신저자)

# 1. 서 론

침입탐지시스템은 탐지기법에 따라 크게 Misuse와 Anomaly 방식으로 구분할 수 있다. Misuse 방식은 사전에 적용된 시그니처와 일치할 경우 공격으로 탐지하며, 시그니처가 확보된 공격에 대해서는 정확한 탐지가 가능하다. 그러나 시그니처 정보를 확보하지 못한 신규 공격은 탐지하지 못한다. Anomaly 방식은 공격이 없는 네트워크상에서 수집된 패킷을 이용하여 정상 패킷에 대한 기준점(normal profile)을 생성하고 이를 벗어난 경우 공격으로 간주하여 탐지하는 방식으로 신규 공격을 탐지하는 것이 가능하지만, 기준점 생성이 어렵고 오탐율(false positive)이 높다[1][2].

이러한 이유로 인해 현재 운용되고 있는 침입탐지시스템 대부분은 Misuse 방식을 사용하나, 신규 악성코드의 급증으로 인해 탐지능력이 한계에 직면하면서 그에 대한 대안으로 Anomaly 방식의 침입탐지시스템이 새롭게 주목을 받고 있다[3].

일반적으로 어렵다고 알려진 Anomaly 기반 침입탐지시스템에서의 기준점 생성 문제를 해결하기 위해 최근 들어 인공지능 기술을 활용한 연구가 활발하게 진행되고 있다. 인공지능 기술의 적용을 제안한 기존 연구들은 최적의 기준점을 만들기 위해 인공 신경망 구조 개선에 적합한 하이퍼파라미터 값을 찾아내고, 효과적인 기계학습 알고리즘 식별을 통해 모델의 학습 수준을 향상시키는 것에 초점을 맞춰 진행되었다. 그러나 기계학습 과정에서 적용한 학습 데이터에 문제가 있는 경우에는 다양한 문제가 발생할 수 있음에도 기존 연구들은 그러한 문제점을 제대로 다루지 못하고 있다. 이에 본 논문에서는 학습 데이터가 기준점 생성과 탐지 성능에 미치는 영향을 실험을 통해 분석하고 학습 데이터의 잘못된 구성으로 인해 발생할 수 있는 문제점들을 해결할 수 있는 방안을 제시한다.

본 논문의 구성을 다음과 같다. 2장에서는 기계학습 원리를 간단하게 고찰하고 인공지능 기술의 적용을 제안한 기존 연구들을 정리한다. 3장에서는 침입탐지 모델 생성과 학습 데이터 적용을 통해 잘못된 학습 데이터가 침입탐지 성능에 미칠 수 있는 영향을 분석하고, 문제점 해결을 위한 학습 데이터 개선방안을 제안한다. 마지막으로 4장에서 연구결과를 요약하고 향후 연구방향을 제시한다.

# 2. 관련 연구

## 2.1 기계학습 원리

인간의 두뇌는 약 1,000억 개의 뉴런으로 구성되어 있고, 1개의 뉴런은 약 6,000개의 다른 뉴런과 연결된 구조를 가진다. 뉴런과 뉴런은 시냅스로 연결되어 있으며, 외부 자극이 임계값을 넘으면 신호가 다음 뉴런으로 전달되고 그렇지 않은 경우 아무런 변화가 없다. 이러한 구조를 본떠 만든 것이 인공 신경망이다.

인공 신경망에서 뉴런의 역할을 하는 요소를 퍼셉트론(perceptron)이라 하며, 입력값이 들어오면 가중치와 바이어스(bias)를 부여하여 가중치 합을 만들고 활성화 함수를 통해 최종 예측값을 도출한다. 인공 신경망은 입력층, 은닉층, 출력층으로 구성되어 있으며, 은닉층은 다시 여러 층으로 구성될 수 있다[4].

컴퓨터에게 입력값과 결괏값만 알려주면 컴퓨터 스스로 분석하여 숨어있는 함수를 찾는 과정이 학습이다. 컴퓨터가 학습을 통해 찾은 함수에 입력값을 적용하여 나온 결과를 예측값이라고 한다. 그리고 실제값인 결괏값과 예측값을 비교하여 오차가 줄어들도록 함수에 적용되는 가중치와 바이어스를 수정하는 과정을 반복한다. 이때 예측값에서 입력값 방향으로 수정해나가는 방식을 '역전파 알고리즘'이라 하며, 학습의 결과물을 '모델'이라고 한다[5].

## 2.2 Dataset

기계학습을 활용하여 Anomaly 기반의 침입탐지시스템을 구축함에 적용할 수 있는 데이터는 여러 가지가 있으며, 그 중 가장 대표적인 데이터는 NSL-KDD Dataset[6][7]이다. 그러나 NSL-KDD Dataset은 현재의 공격 추세가 제대로 반영되어 있지 않기 때문에 최신 침입탐지시스템의 성능평가에는 부적절하며, 실세계에서의 발생한 네트워크 트래픽을 표현하지 못한다[8][9]. 따라서 본 연구에서는 비교적 최신 공격 추세가 반영된 CIC-IDS2017[10]과 CSE-CIC-IDS2018 Dataset[11]을 활용하였다.

CIC-IDS2017과 CSE-CIC-IDS2018 Dataset은 가상

의 사용자가 HTTP, HTTPS, FTP, SSH, email 등의 프로토콜을 사용하는 과정에서 발생된 트래픽 정보를 CICFlowMeter로 분석 후 트래픽 정보에 라벨을 표시하여 CSV(Comma Separated Values) 파일형식으로 만든 데이터이다. CIC-IDS2017은 Port, IP, Protocols 등 78개의 속성정보와 패킷이 정상 또는 어떤 공격 유형에 해당되는지를 나타내는 1개 클래스 정보로 구성되어 있으며, CSE-CIC-IDS2018은 79개 속성정보와 1개 클래스 정보로 구성되어 있다[10]. 클래스 정보는 'benign', 'botnet', 'brute-force attack', 'DoS (Denial-of-Service)', 'DDoS(Distributed Denial-of-Service)', 'heartbleed attack', 'web attacks', 'infiltration'으로 구분되어 있다.

### 2.3 기존 연구

전술한 것처럼 인공지능 기술을 적용하는 연구들은 대부분 특정 기계학습 알고리즘의 적용을 통해 침입탐지 성능 향상을 시도하는 연구에 집중되어 있다.

[12]은 비지도 학습에 사용되는 Support Vector Machine, Decision Tree 및 Random Forest 등의 알고리즘을 비교 분석하여 Random Forest가 최고의 성능을 나타냄을 입증하였다. [13]는 침입탐지 성능 향상을 위해 PCV(Principal Components Analysis)를 사용하여 데이터를 추출하고, MLP(Multi Layer Perceptron)를 활용하여 공격을 탐지하는 기법을 제안하였다.

[14]은 다목적 유전자 알고리즘을 이용하여 특징을 추출할 때 Pearson 상관계수를 적용하여 특징 집합의 크기는 약 30% 줄이면서도 정확성이 유지되는 방안을 제안하였다. [15]는 악성코드 정보를 그레이스케일 이미지로 변환 후 CNN(Convolutional Neural Network)을 활용하여 탐지함으로써 94.5%까지 탐지 성능을 향상시킬 수 있는 기법을 제안하였다.

[16]는 희소 클래스에 대한 탐지능력을 향상시키기 위해 기존 SMOTE(Synthetic Minority Over-sampling Technique)에 부스팅 절차를 조합하여 SMOTEBoost 알고리즘을 제안하였다. [17]은 SMOTE와 K-Nearest Neighbor 알고리즘을 사용하여 희소 클래스가 전체 데이터에서 0.5%이상 차지하도록 하고 탐지성능을 향상하는 방안을 제시하였다.

## 3. 실험 및 탐지성능 평가

### 3.1 실험 환경

MATLab R2019a으로 인공 신경망을 구성하였으며, 첫 번째 은닉층에 Sigmoid, 두 번째 층에 Softmax를 적용하였다. 학습에는 역전파알고리즘의 일종인 SCG (Scaled Conjugate Gradient)를 사용하였다.

모델 생성 및 침입탐지 성능평가를 위한 데이터는 CIC-IDS2017과 CSE-CIC-IDS2018을 사용하여 만들었으며, 속성정보는 두 데이터에 공통으로 존재하는 77개의 속성만 사용하였다. 그리고 infiltration 클래스는 CIC-IDS2017에 36개만 포함되어 있어 정보 부족으로 제외하였고, Heartbleed attack은 CSE-CIC-IDS2018에는 포함되어 있지 않아 제외하였으며, DDoS는 침입탐지가 아닌 다른 방식으로도 효율적인 탐지가 가능함을 고려하여 제외하였다. 이상의 과정을 거쳐 정상과 4개의 공격(botnet, brute-force attack, DoS, web attacks) 클래스만 활용하여 Dataset을 생성하고 구축된 Anomaly 기반 침입탐지시스템 모델로 학습시켜 탐지성능을 평가하였다.

### 3.2 주요 문제점 및 개선방안

#### 3.2.1 학습 데이터 불균형으로 인한 문제점

실제 네트워크에서 발생되는 트래픽은 대부분 정상 패킷에 대한 정보이고, 공격 패킷에 대한 정보는 많지 않다. 그리고 공격 패킷 중에서도 트래픽이 많이 발생하는 클래스가 있고 적게 발생하는 클래스가 있다. CIC-IDS2017 Dataset에도 실제 네트워크 환경의 특성이 잘 반영되어 정상 패킷이 많이 포함되어 있으며, 공격 유형별로도 패킷 개수에 차이가 있다.

실험 데이터로 Train-1과 Test-1을 생성했다. 정상 패킷은 랜덤하게 일부를 추출하였고, 공격 패킷 역시 랜덤하게 반을 나눠서 포함시켰으며, Train-1과 Test-1의 세부 정보는 <표 1>에서 보는 바와 같다.

Train-1과 Test-1을 활용하여 실험1을 진행하였다. Train-1은 학습 데이터로 사용하였고, Test-1은 기계학습을 통해 생성된 모델의 탐지성능을 평가하는데 사용하였다. 실험1의 세부 결과는 <표 2>, <표 3>에서 보는 바와 같다.

<표 1> Train-1과 Test-1의 세부 정보

| 구분 | 클래스 | | | | |
|---|---|---|---|---|---|
| | benign | botnet | brute | DoS | web |
| Train-1 | 264,959 | 983 | 6,918 | 126,332 | 1,091 |
| | 66.19% | 0.25% | 1.73% | 31.56% | 0.27% |
| Test-1 | 264,959 | 983 | 6,917 | 126,329 | 1,089 |
| | 66.19% | 0.25% | 1.73% | 31.56% | 0.27% |

<표 2> Train-1의 학습 결과 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 261,385 | 654 | 2,028 | 2,844 | 1,015 | 97.6%<br>2.4% |
| botnet | 23 | 295 | 0 | 0 | 0 | 92.8%<br>7.2% |
| brute | 575 | 0 | 4,884 | 7 | 74 | 88.2%<br>11.8% |
| DoS | 2,755 | 27 | 4 | 123,007 | 2 | 97.8%<br>2.2% |
| web | 0 | 0 | 0 | 0 | 0 | NaN%<br>NaN% |
| | 98.7%<br>1.3% | 30.2%<br>69.8% | 70.6%<br>29.4% | 97.7%<br>2.3% | 0.0%<br>100% | 97.5%<br>2.5% |
| | benign | botnet | brute | DoS | web | |

<표 3> Train-1 모델의 Test-1 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 261,755 | 641 | 2,010 | 3,354 | 1,016 | 97.4%<br>2.6% |
| botnet | 25 | 322 | 0 | 0 | 0 | 92.8%<br>7.2% |
| brute | 538 | 0 | 4,896 | 4 | 71 | 88.9%<br>11.1% |
| DoS | 2,641 | 20 | 11 | 122,971 | 2 | 97.9%<br>2.1% |
| web | 0 | 0 | 0 | 0 | 0 | NaN%<br>NaN% |
| | 98.8%<br>1.2% | 32.8%<br>67.2% | 70.8%<br>29.2% | 97.3%<br>2.7% | 0.0%<br>100% | 97.4%<br>2.6% |
| | benign | botnet | brute | DoS | web | |

실험1의 결과를 살펴보면, 학습 결과와 성능평가 결과가 비슷하게 나타나 학습이 제대로 이뤄졌다고 볼 수 있다. 그러나 클래스별로 탐지된 결과에서는 유의미한 차이가 나타났는데, 특히 전체 학습 데이터에서 클래스가 차지하는 비율과 유사한 탐지결과가 나타남을 확인하였다. 큰 비율을 차지하는 benign은 98.85%, DoS는 97.3%의 높은 탐지결과를 보인 반면 비율이 낮은 botnet의 경우 32.5%로 낮은 탐지성능을 보였고, Web은 0%로 전혀 탐지하지 못하였다.

클래스가 전체 데이터에서 차지하는 비율이 모델의 탐지성능에 어떠한 영향을 미치는지 알아보기 위해 실험2를 진행하였다. 이를 위해 클래스별로 데이터 개수가 균등하게 분포되어 있는 Train-2, Train-3을 생성하였으며. 세부 정보는 <표 4>와 같다. 그리고 Train-2와 Train-3을 통해 생성된 모델의 탐지성능을 분석하기 위해 Test-1을 사용하였으며, 그 결과는 <표 5>, <표 6>에서 보는 바와 같다.

<표 4> Train-2와 Train-3의 세부 정보

| 구분 | 클래스 | | | | |
|---|---|---|---|---|---|
| | benign | botnet | brute | DoS | web |
| Train-2 | 1,616 | 400 | 398 | 401 | 415 |
| | 50.03% | 12.38% | 12.32% | 12.41% | 12.85% |
| Train-3 | 4,000 | 983 | 1,000 | 1,011 | 1,091 |
| | 49.47% | 12.16% | 12.37% | 12.5% | 13.49% |

<표 5> Train-2 모델의 Test-1 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 234,114 | 7 | 787 | 9,305 | 59 | 95.8%<br>4.2% |
| botnet | 10,937 | 976 | 5 | 231 | 0 | 8.0%<br>92.0% |
| brute | 3,994 | 0 | 6,121 | 685 | 76 | 56.3%<br>43.7% |
| DoS | 11,404 | 0 | 1 | 113,309 | 4 | 90.9%<br>9.1% |
| web | 4,510 | 0 | 3 | 2,799 | 950 | 11.5%<br>88.5% |
| | 88.4%<br>11.6% | 99.3%<br>0.7% | 88.5%<br>11.5% | 89.7%<br>10.3% | 87.2%<br>12.8% | 88.8%<br>11.2% |
| | benign | botnet | brute | DoS | web | |

<표 6> Train-3의 Test-1 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 235,480 | 35 | 785 | 9,832 | 16 | 95.7%<br>4.3% |
| botnet | 10,175 | 946 | 5 | 269 | 0 | 8.3%<br>91.7% |
| brute | 2,273 | 0 | 6,123 | 1,624 | 130 | 60.3%<br>39.7% |
| DoS | 12,753 | 2 | 3 | 112,557 | 8 | 89.8%<br>10.2% |
| web | 4,278 | 0 | 1 | 2,047 | 935 | 12.9%<br>87.1% |
| | 88.9%<br>11.1% | 96.2%<br>3.8% | 88.5%<br>11.5% | 89.1%<br>10.9% | 85.9%<br>14.1% | 88.9%<br>11.1% |
| | benign | botnet | brute | DoS | web | |

Train-2와 Train-3에서 각 클래스의 구성 비율은 benign, botnet, brute-force, DoS, web이 4:1:1:1:1로 유사하나 데이터 개수는 Train-3이 Train-2보다 약 2배 정도 더 많다. 그러나 Test-1에 대한 성능평가 결과, Train-2와 Train-3 모델이 비슷한 탐지결과를 나타내어 학습 데이터가 많아진다고 무조건 모델의 탐지성능이 향상되지는 않음을 확인하였다. 그리고

Train-2에서 botnet, brute-force, web의 데이터 수가 Train-1보다 적지만 Test-1에 대한 성능평가 결과 Train-2 모델이 더 높은 탐지결과를 나타내 학습 시 데이터의 개수뿐만 아니라 데이터의 구성비율도 탐지 성능에 영향을 미칠 수 있음을 확인하였다.

네트워크에서 발생되는 트래픽 중 정상 패킷에 대한 정보 수집이 용이하여 학습 데이터에서 차지하는 비율까지도 높게 설정하면 공격 패킷이 전체 데이터에서 차지하는 비율이 낮아져 희소 클래스 탐지능력은 현저하게 저하되는 현상이 나타날 수 있다. 따라서 학습 데이터를 생성할 때 각 클래스가 균등한 비율이 유지되도록 구성하는 것이 바람직하다.

### 3.2.2 학습하지 않은 클래스 미식별 현상

Anomaly 기반 침입탐지시스템은 정상 패킷에 대한 기준점을 만들고 이를 벗어난 패킷은 공격 패킷으로 식별하는 것이다. 이러한 원리를 바탕으로 기계학습 시 모델에 정상 및 공격 패킷 중 하나만 학습을 시키고 탐지성능을 평가할 수 있다.

실험의 진행을 위해서는 4개의 클래스로 구별되어 있는 공격 패킷을 모두 합쳐 1개의 공격 클래스로 구성할 필요가 있다. 이에 Train-2를 바탕으로 새로운 데이터를 생성하였으며, 정상 패킷의 정보만 포함된 Train-4와 공격 패킷의 정보만 포함된 Train-5로 구분하였다.

성능평가를 위해 Test-1의 공격 클래스를 합쳐서 새로운 Test-2 데이터를 생성하였다. 실험3에서는 Train-4와 Train-5로 학습을 시키고 Test-2로 성능을 평가하였으며, 데이터에 대한 세부 정보와 실험 결과는 <표 7>, <표 8>, <표9>에서 확인할 수 있다.

<표 7> Train-4, Train-5, Test-2의 세부 정보

| 구분 | 클래스 | | | | |
|---|---|---|---|---|---|
| | benign | attack | | | |
| | | botnet | brute | DoS | web |
| Train-4 | 1,616 | 0 | | | |
| | | 0 | 0 | 0 | 0 |
| Train-5 | 0 | 1,614 | | | |
| | | 400 | 398 | 401 | 415 |
| Test-2 | 264,959 | 135,18 | | | |
| | | 983 | 6,917 | 126,329 | 1,089 |

<표 8> Train-4 학습 및 성능 Confusion matrix

| Train-4 학습 결과 | | | | Test-2 성능 평가 | | | |
|---|---|---|---|---|---|---|---|
| benign | 1,616 | 0 | 100%<br>0.0% | benign | 264,595 | 135,318 | 100%<br>0.0% |
| attack | 0 | 0 | NaN%<br>NaN% | attack | 0 | 0 | NaN%<br>NaN% |
| | 100%<br>0.0% | NaN%<br>NaN% | 100%<br>0% | | 100%<br>0.0% | 0.0%<br>100% | 66.2%<br>33.8% |
| | benign | attack | | | benign | attack | |

<표 9> Train-5 학습 및 성능 Confusion matrix

| Train-5 학습 결과 | | | | Test-2 성능 평가 | | | |
|---|---|---|---|---|---|---|---|
| benign | 0 | 0 | NaN%<br>NaN% | benign | 0 | 0 | NaN%<br>NaN% |
| attack | 0 | 1,614 | 100%<br>0.0% | attack | 264,595 | 135,318 | 100%<br>0.0% |
| | NaN%<br>NaN% | 100%<br>0.0% | 100%<br>0% | | 0.0%<br>100% | 100%<br>0.0% | 33.8%<br>66.2% |
| | benign | attack | | | benign | attack | |

실험3의 결과에서 알 수 있듯이, 모델이 정상 패킷만을 학습하게 되면 모든 패킷을 정상으로 분류하게 되어 공격 패킷을 식별할 수 없고, 공격 패킷만 학습한 경우에도 결과는 동일하다. 따라서 모델의 탐지능력을 향상시키기 위해서는 정상과 공격 패킷 정보 모두를 학습시키는 것이 필수적이다.

### 3.2.3 모델의 효과적인 업데이트 방안

기존 연구들은 동일 기간에 수집된 데이터를 활용하여 모델의 성능을 평가하였고, 다른 기간에 수집한 데이터를 대상으로 성능을 비교한 경우가 없었다. 이에 본 연구에서는 2017년도 데이터인 CIC-IDS2017과 2018년도 데이터인 CSE-CIC-IDS2018을 활용하여 모델을 생성하고 성능을 비교하는 실험4를 진행하였다. 실험을 위해 CSE-CIC-IDS2018을 사용하여 새로운 데이터인 Train-6과 Test-3을 만들었고, 세부 정보는 <표 10>과 같다. 실험4는 구축된 모델이 새로운 공격을 탐지할 수 있는지를 알아보기 위해 2017년 자료인 Train-1로 학습을 시키고 2018년 자료인 Test-3을 사용하여 모델의 탐지성능을 평가하였고, 과거 공격에 대한 탐지성능을 알아보기 위해 2018년 자료인 Train-6로 학습을 하고 2017년 자료인 Test-1을 사용하여 모델의 탐지성능을 평가하였다. 실험4의 세부결과는 <표 11>, <표 12>에서 보는 바와 같다.

<표 10> Train-6과 Test-3의 세부 정보

| 구분 | 클래스 | | | | |
|---|---|---|---|---|---|
| | benign | botnet | brute | DoS | web |
| Train-6 | 1,863 | 466 | 466 | 467 | 466 |
| | 49.97% | 12.5% | 12.5% | 12.53% | 12.5% |
| Test-3 | 136,050 | 143,096 | 190,474 | 79,334 | 462 |
| | 24.76% | 26.05% | 34.67% | 14.44% | 0.08% |

<표 11> Train-1의 Test-3 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 106,087 | 141,460 | 46,994 | 11,685 | 155 | 34.6%<br>65.4% |
| botnet | 3,703 | 1,567 | 0 | 685 | 0 | 26.3%<br>73.7% |
| brute | 8,773 | 28 | 6,959 | 4,513 | 127 | 34.1%<br>65.9% |
| DoS | 7,726 | 41 | 3 | 16,544 | 138 | 67.7%<br>32.3% |
| web | 9,761 | 0 | 136,518 | 45,907 | 42 | 0.0%<br>100% |
| | 78.0%<br>22.0% | 1.1%<br>98.9% | 3.7%<br>96.3% | 20.9%<br>79.1% | 9.1%<br>90.9% | 23.9%<br>76.1% |
| | benign | botnet | brute | DoS | web | |

<표 12> Train-6의 Test-1 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 243,485 | 669 | 5,478 | 86,378 | 915 | 72.3%<br>27.7% |
| botnet | 610<br>0.2% | 0 | 0 | 412 | 0 | 0.0%<br>100% |
| brute | 2,553 | 4 | 1,437 | 291 | 79 | 32.9%<br>67.1% |
| DoS | 9,393 | 310 | 1 | 10,932 | 38 | 52.9%<br>47.1% |
| web | 8,918 | 0 | 1 | 28,316 | 57 | 0.2%<br>99.8% |
| | 91.9%<br>8.1% | 0.0%<br>100% | 20.8%<br>79.2% | 8.7%<br>91.3% | 5.2%<br>94.8% | 63.9%<br>36.1% |
| | benign | botnet | brute | DoS | web | |

실험4의 결과에서 확인할 수 있듯이 새로운 공격에 대한 탐지능력은 23.9%이고, 과거 공격에 대한 탐지능력도 63.9%로 낮게 나타나, 모델의 업데이트가 지속적으로 수행되지 않으면 탐지성능이 심각하게 저하될 수 있음을 확인하였다. 그래서 과거 공격과 새로운 공격 모두를 탐지할 수 있도록 Train-1과 Train-6을 합쳐 Train-7을 만들어 학습을 시키고, Test-1과 Test-3으로 탐지성능을 평가하는 실험5를 진행하였다. 실험의 세부 결과는 <표 13>, <표 14>에서 보는 바와 같다.

<표 13> Train-7의 Test-3 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 122,188 | 458 | 94 | 2,482 | 9 | 97.6%<br>2.4% |
| botnet | 642 | 142,257 | 0 | 272 | 0 | 99.4%<br>0.6% |
| brute | 5,674 | 284 | 186,660 | 41,843 | 3 | 79.6%<br>20.4% |
| DoS | 3,668 | 8 | 3,628 | 29,692 | 24 | 80.2%<br>19.8% |
| web | 3,878 | 89 | 92 | 5,045 | 426 | 4.5%<br>95.5% |
| | 89.8%<br>10.2% | 99.4%<br>0.6% | 98.0%<br>2.0% | 37.4%<br>62.6% | 92.2%<br>7.8% | 87.6%<br>12.4% |
| | benign | botnet | brute | DoS | web | |

<표 14> Train-7의 Test-1 Confusion matrix

| | | | | | | |
|---|---|---|---|---|---|---|
| benign | 234,546 | 45 | 783 | 40,584 | 7 | 85.0%<br>15.0% |
| botnet | 8,073 | 938 | 4 | 332 | 0 | 10.0%<br>90.0% |
| brute | 2,363 | 0 | 6,106 | 732 | 78 | 65.8%<br>34.2% |
| DoS | 12,342 | 0 | 11 | 58,216 | 3 | 82.5%<br>17.5% |
| web | 7,635 | 0 | 13 | 26,465 | 1,001 | 2.9%<br>97.1% |
| | 88.5%<br>11.5% | 95.4%<br>4.6% | 88.3%<br>11.7% | 46.1%<br>53.9% | 91.9%<br>8.1% | 75.1%<br>24.9% |
| | benign | botnet | brute | DoS | web | |

실험5의 결과, Test-3에 대한 탐지능력은 23.9%에서 87.6%로 높아졌고, Test-1에 대한 탐지능력도 63.9%에서 75.1%로 높아짐을 확인하였다. 즉 과거와 현재의 공격을 망라하여 효과적인 탐지를 수행하기 위해서는 과거의 학습 데이터에 새로운 공격에 대한 정보를 지속적으로 추가하여 업데이트하면서 침입탐지 모델에 학습을 시켜나가는 것이 필요하다.

## 4. 결론

인공지능 기술을 적용하여 Anomaly 기반 침입탐지 시스템의 탐지성능을 향상시키기 위해서는 최적의 하이퍼파라미터 값을 찾아 인공 신경망의 구조를 개선하는 것도 중요하다. 그러나 아무리 좋은 인공 신경망을 구성하여도 부적절한 데이터로 학습을 진행하면 모델의 탐지성능이 심각하게 저하되는 문제가 발생할 수 있다. 본 연구는 최신 경향을 반영한 Dataset을 활용하여 다양한 학습 데이터를 구성하고 실험을 통해 학습 데이터가 탐지성능에 미치는 부정적인 영향을 식별하고 그에 대한 개선 방안을 제안하였다.

효과적인 학습 데이터를 생성하기 위해서는 우선 전체 학습 데이터에서 각 클래스가 차지하는 비율이

균등해야 한다. 둘째, 학습하지 않은 클래스는 식별하지 못하기 때문에 학습 시 모든 클래스를 포함하여 학습이 될 수 있도록 데이터를 구성해야 한다. 셋째, 기존 학습 데이터에 새로운 학습 데이터를 추가하여 업데이트하고 학습을 지속적으로 수행하는 것이 과거 공격과 새로운 공격 모두를 탐지하는데 효과적이다.

향후에는 학습 데이터에서 개별 속성이 탐지성능에 어떠한 영향을 미치는지 조사하고, 정상 패킷과 공격 패킷의 구성비율도 추가적인 실험을 통해 분석하여 탐지성능을 극대화할 수 있는 최적의 학습 데이터를 구성하는 방안을 모색해 나갈 예정이다.

## 참고문헌


[1] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Macia-Fernandez and E. Vazquez, “Anomaly-based network intrusion detection: Techniques, systems and challenges”, Computers & Security, 28(1-2), pp.18-28, 2009

[2] 이윤환, 이수진, “국방통합보안관제체계에서의 협업 침입탐지를 위한 탐지규칙 교환 기법”, 융합보안논문지 제11권 제1호, pp.57-69, 2011.

[3] 김태희, 강승호, “실시간 탐지를 위한 인공신경망 기반의 네트워크 침입탐지 시스템”, 융합보안논문지, 제17권 1호, pp.31-38, 2017.

[4] 조태호, ‘모두의 딥러닝’, 길벗, 2019.

[5] 나카이 에츠지. 김범주(역), ‘머신러닝 이론 입문’, 위키북스, 2017.

[6] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, “A Detailed Analysis of the KDD CUP 99 Data Set”, Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA), 2009

[7] https://www.unb.ca/cic/datasets/nsl.html

[8] C. Brown, A. Cowperthwaite, A. Hijazi, and A. Somayaji, “Analysis of the 1999 darpa/lincoln laboratory ids evaluation data with netadhict”, IEEE SCISDA, pp.1-7, 2009.

[9] J. McHugh, “Testing intrusion detection systems: A critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory”, ACM Transaction of Information, System and Security, pp.262-294, 2000.

[10] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, “Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization”, 4th International Conference on Information Systems Security and Privacy (ICISSP), pp.108-116, January 2018.

[11] https://registry.opendata.aws/cse-cic-ids2018/

[12] 정윤경, 박기남, 김현주, 김종현, 현상원. “클래스 불균형 데이터에 적합한 기계 학습 기반 침입 탐지 시스템”, 정보보호학회논문지 vol.27, no.6, pp.1385- 1395, 2017

[13] V. Golovko, L. Vaitsekhovich, “Neural Network Approaches for Intrusion Detection and Recognition”, International Journal of Computing, vol.5, no.3, pp.118-125, 2014

[14] 강승호, 정인선, 임형석, “실시간 공격 탐지를 위한 Pearson 상관계수 기반 특징 집합 선택 방법”, 융합보안논문지 제18권 제5호, pp.59-66, 2018.

[15] Zhihua Cui, Fei Xue, Xingjuan Cai, Yang Cao, Gai-ge Wang and Jinjun Chen, "Detection of Malicious Code Variants Based on Deep Learning", IEEE Transactions on Industrial Informatics, 14(7), pp.3187-3196, 2018

[16] Nitesh V. Chawla, Aleksandar Lazarevic, Lawrence O. Hall, and Kevin W. Bowyer, “SMOTEBoost: Improving Prediction of the Minority Class in Boosting”, European Conference on Principles of Data Mining and Knowledge Discovery, pp.107-119, 2003.

[17] 서재현, “기계학습 방법에 기반 한 불균형 침입탐지 데이터 분류법의성능평가에 관한 연구”, 한국지능시스템학회 논문지, vol.27, no.5, pp.466-474, 2017

## 〔저자소개〕

문 상 태 (Sangtae Moon)
2008년 3월 육군사관학교 학사
2018년 1월 ~ 현재
국방대학교 국방과학학과 석사과정

email : sangtae850321@gmail.com

이 수 진 (Soojin Lee)
1992년 3월 육군사관학교 학사
1996년 2월 연세대학교 석사
2006년 2월 한국과학기술원 박사
2006년 3월 ~ 현재
국방대학교 국방과학학과 교수

email : cyberkma@gmail.com