J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

이산화 전처리 방식 및 컨볼루션 신경망을 활용한 네트워크 침입 탐지에 대한 연구☆

A Research on Network Intrusion Detection based on Discrete Preprocessing Method and Convolution Neural Network

유 지 훈[1] 민 병 준[1] 김 상 수[2] 신 동 일[1] 신 동 규[1*]
JiHoon Yoo Byeongjun Min Sangsoo Kim Dongil Shin Dongkyoo Shin

요 약

새롭게 발생되는 사이버 공격으로 인해 개인, 민간 및 기업의 피해가 증가함에 따라, 이에 기반이 되는 네트워크 보안 문제는 컴퓨터 시스템의 주요 문제로 부각되었다. 이에 기존에 사용되는 네트워크 침입 탐지 시스템(Network Intrusion Detection System: NIDS)에서 발생되는 한계점을 개선하고자 기계 학습과 딥러닝을 활용한 연구 이뤄지고 있다. 이에 본 연구에서는 CNN(Convolution Neural Network) 알고리즘을 이용한 NIDS 모델 연구를 진행한다. 이미지 분류 기반의 CNN 알고리즘 학습을 위해 기존 사용되는 전처리 단계에서 연속성 변수 이산화(Discretization of Continuous) 알고리즘을 추가하여 예측 변수에 대해 선형 관계로 표현하여 해석에 용이한 데이터로 변환 후, 정사각형 행렬(Square Matrix) 구조에 매칭된 픽셀(Pixel) 이미지 구조를 모델에 학습한다. 모델의 성능 평가를 위해 네트워크 패킷 데이터인 NSL-KDD를 사용하였으며, 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 및 조화평균(F1-score)을 성능 지표로 사용하였다. 실험 결과 제안된 모델에서 85%의 정확도로 가장 높은 성능을 보였으며, 학습 표본이 적은 R2L 클래스의 조화평균이 71% 성능으로 다른 모델에 비해서 매우 좋은 성능을 보였다.

☞ 주제어: NSL-KDD, 네트워크 이상 탐지, CNN, 연속형 변수 이산형화

ABSTRACT

As damages to individuals, private sectors, and businesses increase due to newly occurring cyber attacks, the underlying network security problem has emerged as a major problem in computer systems. Therefore, NIDS using machine learning and deep learning is being studied to improve the limitations that occur in the existing Network Intrusion Detection System. In this study, a deep learning-based NIDS model study is conducted using the Convolution Neural Network (CNN) algorithm. For the image classification-based CNN algorithm learning, a discrete algorithm for continuity variables was added in the preprocessing stage used previously, and the predicted variables were expressed in a linear relationship and converted into easy-to-interpret data. Finally, the network packet processed through the above process is mapped to a square matrix structure and converted into a pixel image. For the performance evaluation of the proposed model, NSL-KDD, a representative network packet data, was used, and accuracy, precision, recall, and f1-score were used as performance indicators. As a result of the experiment, the proposed model showed the highest performance with an accuracy of 85%, and the harmonic mean (F1-Score) of the R2L class with a small number of training samples was 71%, showing very good performance compared to other models.

☞ keyword : NSL-KDD, Network Intrusion Detection, CNN, Discretization of Continuous

1. 서 론

사이버 공격이 발전함에 따라 알려지지 않은 취약점 악용 및 알려진 취약점 우회와 같은 다양한 공격을 통해 개인, 기업 및 국가를 가리지 않고 많은 피해를 발생 시키고 있다 [1, 2]. 이러한 사이버 공격의 기반이 되는 네트워크 보안 문제는 컴퓨터 시스템의 주요 문제 중 하나로 떠오르고 있으며, 가장 대표적인 네트워크 보안 메커니즘인 NIDS(Network Instruction Detection System)에 대해서 많은

1 Dept. of Computer Science, Sejong University, Region(city), 05006, Korea
2 Agency for Defense Development, Daejeon 305600, South Korea
* Corresponding author: shindk@sejong.ac.kr
[Received 23 November 2020, Reviewed 30 November 2020(R2 5 January 2021), Accepted 16 February 2021]
☆ 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다.(UD200014ED)

연구가 이루어지고 있다 [3, 4]. 전통적인 NIDS의 침입 탐지 방식은 설정된 시그니쳐를 기반으로 공격을 탐지하는 오용 탐지 방법과, 정상적인 사용 패턴을 기반으로 비정상 패턴을 공격으로 탐지하는 이상 탐지 방식을 사용하였다. 하지만 다양한 침입 방법과 빠르게 생성되는 새로운 형태의 사이버 공격은 더 이상 기존의 NIDS 메커니즘으로 안정성을 보장할 수 없게 되었다 [5]. 오용 탐지의 경우 설정된 시그니쳐의 의존성에 의해 알려지지 않은 공격(Zero-day Attack)을 탐지할 수 없다. 이상 탐지의 경우 정상적인 패턴에서 벗어나는 것을 공격으로 인지하는 탐지 방법으로 알려지지 않은 공격에 대해서 탐지가 가능하지만, 현실의 문제에서 정상적인 패턴을 정의하기 어렵기 때문에 오경보(High-False Alarm) 문제를 발생한다 [6].

이에 연구자들은 공격과 정상에 대한 패턴 데이터를 기계 학습(Machine Learning)과 딥러닝(Deep Learning) 모델을 통해 스스로 학습하고, 공격과 정상을 판단하여 예측할 수 있는 학습 모델 기반의 NIDS에 대한 연구가 진행되었다 [7].

본 연구에서는 정상적인 네트워크 트래픽과 악의적인 네트워크 트래픽을 탐지하는 NIDS를 위해 CNN(Convolution Neural Network) 기반의 NIDS 연구를 진행하며, 대표적인 네트워크 트래픽인 NSL-KDD 데이터를 사용해 모델 학습 및 평가를 진행한다. 다양한 특성(Mixed-Type Feature)으로 구성되어 있는 NSL-KDD 데이터의 경우 예측 변수에 대한 관계 해석이 어렵기 때문에, 이를 해결하기 위해 Min et al. 연구에서 사용된 기존 전처리 과정에 연속성 변수 이산화(Discretization of Continuous) 알고리즘을 추가한다 [8]. 이를 통해 가공된 데이터는 다양한 속성 관계에서 벗어나 관계 해석이 용이한 데이터로 변환된다. 이렇게 가공된 네트워크 트래픽은 이미지 분류 기반에 적합한 데이터로 변환하기 위해 정사각형 행렬(Square Matrix)에 맵핑되어 3차원 픽셀(Pixel) 이미지 구조로 학습을 진행한다. 학습을 위한 전처리 공정이 완료되면 다양한 공격으로 구성되어 있는 NSL-KDD 데이터 세트는 각 공격의 대표성을 지니는 5개의 범주(Normal, DoS, Probe, R2L 및 U2R)로 분류되어, 모델 성능 평가에 사용되는 정확도(Accuracy), 정밀도(Precision), 재현율 (Recall) 및 조화평균(F1-score)을 통해 제안된 모델의 성능을 평가한다.

2. 관련 연구

해당 섹션에서는 제안된 모델 학습 및 평가에 사용되는 NSL-KDD 데이터 세트와 기존 연구된 기계 학습 및 딥 러닝 기반의 네트워크 침입 탐지 시스템에 대해 설명한다.

2.1 NSL-KDD 데이터 세트

NSL-KDD 데이터는 DARPA(Defense Advanced Research Prjoects Agency)에서 개발한 KDD CUP 1999 데이터에서 McHugh가 제기한 Overload TCP Dump Packet 유실, 공격에 대한 정의 부족, 많은 중복 및 무의미한 레코드를 개선한 데이터이다 [9, 10]. NSL-KDD 데이터 세트는 많은 기계 학습 및 딥러닝 기반의 IDS 연구에서 성능을 비교하기 위해 사용하며, 표1 과 같이 Normal, DoS, Probe, U2R 및 R2L으로 공격 및 정상 패턴에 대해서 5개의 대표성을 가지는 클래스로 범주화하여 사용된다.

(Table 1) NSL-KDD Representative Class

Class	구성된 공격	총합
DoS	Apache2, Back, … Worm	11
Probe	Ipsweep, Mscan, … Stan	6
U2R	Buffer_overflow, Loadmodule, …, Xterm	7
R2L	Ftp_write, Guess_passwd, …, Spy,	15
Normal	Normal	1

표 2는 5개 클래스의 실제 데이터 분포와 비율에 대해서 확인할 수 있다. 데이터의 비율을 보면 U2R와 R2L 클래스 데이터가 매우 적은 표본을 가지고 있는 것을 확인할 수 있으며, KDDTest 데이터에만 포함된 하위 공격을 가지고 있다 [11]. 해당 데이터는 처음 나올시 기계 학습 및 딥러닝에 대한 고려가 되어 있지 않은 데이터이기에, 해당 데이터로 학습시 실제 정확도에 비해 U2R과 R2L의 성능이 비교적으로 낮게 나오는 것을 확인 할 수 있다. 본 연구에서는 이러한 부분을 고려하여 적은 비율을

가지는 2개 클래스에 대해서 일반화 성능을 높이는 방향으로 학습을 진행한다.

(Table 2) NSL-KDD Class Data Distribution

Dataset	Class	Number	Rate
KDDTrain+	Total	125,973	100%
	Normal	67,343	53%
	DoS	45,927	37%
	Probe	11,656	9.11%
	U2R	52	0.04%
	R2L	995	0.85%
KDDTest+	Total	22,544	100%
	Normal	9,711	43%
	DoS	7,458	33%
	Probe	2,421	11%
	U2R	200	0.9%
	R2L	2,654	12.1%

2.2 기계 학습 기반 NIDS

기계 학습은 데이터를 바탕으로 데이터에 대한 패턴을 학습하여 스스로 성능을 향상 시키는 기술 의미하며, 지도학습(Supervised Learning)과 비지도학습(Unsupervised Learning) 방식으로 구분된다. NIDS에서 지도학습은 네트워크 트래픽에 대한 공격과 정상과 같은 레이블(Label)이 존재하는 데이터를 학습하여, 공격 또는 정상이라는 분류(Classification)할 수 있는 패턴을 학습한다. 비지도 학습의 경우 레이블이 달려있지 않은 데이터에 대한 특징 압축을 통해 네트워크의 주성분을 분석하거나, 군집화를 통해 가까운 거리 또는 유사한 분포의 데이터를 군집화 할 수 있다 [12].

Chae et al의 연구에서는 Decision Tree 알고리즘에 대하여 학습 데이터과 테스트 데이터를 10-fold cross-validation하는 연구를 진행했다 [13]. Sabnani et al의 연구에서는 4개의 공격 클래스에 대해서 MLP(Multi-Layer Perceptron), K-means 및 Gaussian Classifer를 사용하여 높은 탐지 정확도를 보여주었다 [14]. Yao et al의 연구에서는 SVM의 데이터 특성에 맞는 가중 커널 함수를 사용하여 기존 SVM보다 더 나은 성능을 보여준다 [15]. Guan et al의 연구서는 K-mean을 변형한 Y-mean 알고리즘을 통해 새로운 클러스터링 방법을 제안하였다 [16]. Lakhina et al의 연구에서는 PCA(Principal Component Analysis) 알고리즘을 통해 축소된 입력 특성으로 학습에 필요한 자원과 시간을 줄이며, 더 좋은 분류 성능이 나타나는 것을 확인할 수 있었다 [17].

2.3 딥 러닝 기반 NIDS

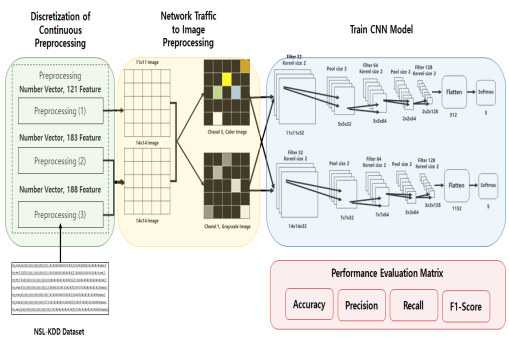
딥러닝은 인간의 심층신경망 이론을 기반으로 고안된 알고리즘으로, 기존 기계 학습을 이용한 NIDS에서 발생되는 시간 복잡도, 데이터 및 패턴 식별과 같은 데이터 처리에 대한 한계점을 해결하기 위해서 연구되었다. 딥러닝의 핵심은 데이터를 바탕으로 기대 출력과 가깝게 만드는 표현(Representation)을 추출하는 것으로, 공격과 정상에 대한 네트워크 특징 표현을 추출하여, 기존의 기계 학습보다 더 추상화된 특징을 표현하게된다 [18]. Potluri etl al은 가속화된 DNN(Deep Neural Network)을 활용하여 NSL-KDD 데이터에 대한 훈련 시간 및 탐지에 대한 효과적인 분석이 가능한 모델 설계 연구 했다 [19].

Torres etal은 RNN(Recurrent Neural Network) 학습을 통해 네트워크 트래픽에 대한 시간 특성을 학습하는 연구를 진행하였다 [20].

Imamverdiyev et al은 DoS 공격 탐지 정확도를 높이기 위해서 7개의 Layer로 구성된 Gaussian-Bernoulli 유형의 RBM(Restricted Boltzmann Machine)을 설계하여 DBN (Deep Belif Network)과 정확도를 비교하였다 [21]. Lopez-Martin의 연구에서는 기존 딥러닝 연구 방식이 아닌 DRL(Deep Reinforcement Learning)을 NSL-KDD 및 다른 데이터 세트에 적용하는 연구를 진행한다. 해당 연구는 다양한 강화 학습 모델을 통해 얻은 분류기(Classifier)를 NIDS에 적용할 수 있음을 확인할 수 있었다 [22].

3. 제안된 CNN 기반의 NIDS

본 연구에서는 NSL-KDD 데이터 세트에 대한 학습 모델들의 분류 성능을 높이기 위해서 아래의 그림 1과 같은 시스템 구조를 제안한다.



(Figure 1) Architecture of the Proposed System

먼저 3.1절을 통해 NSL-KDD의 데이터 특성 유형을 파악한 후, 3.2절에서 제안된 Min et al.의 일반적인 전처리 방식에 연속성 변수 이산화 알고리즘이 추가된 전처리를 진행한다. 이를 통해 예측 변수에 대한 관계 해석에 선형 관계를 지니는 데이터로 가공되며, CNN 알고리즘 입력 형태로 변환하기 위해 정사각형 행렬(Square Matrix)를 이용한 이미지 픽셀 형태로 변환하여 실험에 사용한다. 제안된 모델의 성능을 평가하기 위해서 기계 학습과 딥 러닝 모델에서 사용되는 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), 조화평균(F1-score)를 사용한다

3.1 NSL-KDD 특성 분석

NSL-KDD 데이터는 네트워크에 대한 정보뿐만 아니라, 데이터의 다양성을 추가하기 위해 시간에 대한 정보와 호스트에 대한 일련의 정보도 포함되어 있다. 이를 통해 NIDS 뿐만 아니라, HIDS에서도 분석이 가능한 데이터 세트로 HIDS 연구 모델에서도 NSL-KDD 데이터로 분석한 사례들이 존재한다. 표3은 데이터를 구성하는 4개의 범주에 대해서 분류되어 있다. 1~22번까지의 특성은 일반적은 네트워크에 대한 특성 값이며, 23-41까지의 속성은 2초 동안 분석된 호스트 및 입력 트래픽에 대한 개수 및 비율로 구성되어 있다.

(Table 3) NSL-KDD Data Collection Information

Feature Number	Features Information
1-9	Network Packet Header
10-22	Network Packet Payload
23-31	Time-Based Feature
32-41	Host-Based Feature
42	Normal & Attack Type

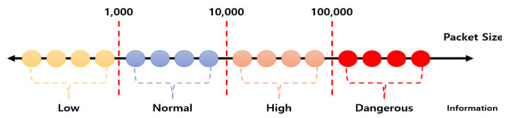
표 3에 의해 생성된 데이터 세트를 기계 학습에 사용가능한 특성 유형으로 분류하게 되면 표 4와 같이 분류된다. Network Packet 특성들은 4개의 분류에 고르게 분포되어 있는 것을 확인할 수 있으며, Time-Based와 Host-Based 특성은 개수와 비율로 구성되기 때문에 주로 Discrete 유형에 속하는 것을 알 수 있다. 특성의 유형별로 데이터의 범위 차이, 문자열로 구성된 데이터 및 다양한 특성으로 구성된 데이터를 그대로 학습하는 경우 기계 학습과 딥 러닝 모델에서 정상적인 학습을 진행할 수 없기에 반드시 전처리가 필요하다.

(Table 4) NSL-KDD Attribute Type Classification

Type	Feature	Number
Binary	7, 12, 14, 20, 21, 22	6
Categorical	2, 3, 4, 42	4
Discrete	8, 9, 15, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34 ,35, 36, 37, 38, 39, 40, 41, 43	23
Continuous	1, 5, 6, 10, 11, 13, 16, 17, 18, 19	10

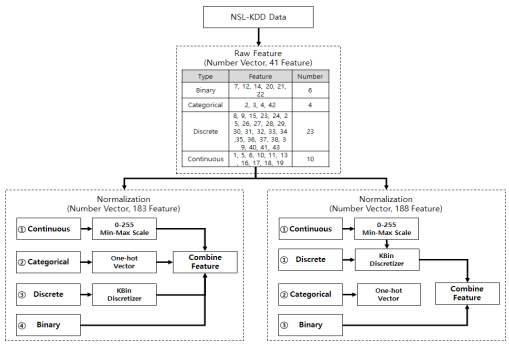
3.2 연속 변수 이산화 전처리

Continuous와 Discrete 와 같은 연속된 데이터는 제한된 자유도(Degrees of Freedom, DOF) 문제를 수반하게 되는데, 이는 예측 변수에 대한 비선형 상관 관계를 만들어 모델의 학습을 복잡하게 만들게 된다.



(Figure 2) Discretization of Continuous

그림 2와 같이 네트워크 패킷에서 DoS를 구분하는 가장 큰 특징인 Packet size로 설명하면 10,000,000과 100,000,000 패킷 크기는 모두 '패킷의 크기가 크다'라는 동일한 정보를 전달하기 때문에, Packet_size = 1 if size > 10,000,000 else 0 과 같이 이산화하여 표현하는 경우 예측 변수와 선형 관계로 표현되어 해석에 용이한 데이터로 변화할 수 있다. 이에 본 연구에서는 Min et al에서 제안한 일반적인 기계 학습 전처리 과정에서 Continuous와 Discrete 속성에 대한 이산화(Discretization) 알고리즘을 추가한 전처리 방법을 제안한다.



(Figure 3) Discretization of Continuous Architecture

제안하는 전처리 과정은 그림 3에 제시되어 있으며, 각 과정에 대한 설명은 아래와 같다:

1. 3.1절 표4에 제시된 NSL-KDD 데이터 세트 특성인 Binary, Categorcial, Discrete 및 Continuous로 데이터를 구분한다.
2. 전체 데이터 속성 중에서 num_outbound_cmds 특성은 표준편차가 0으로 제거한다.
3. Object 타입으로 구성된 Nominal 데이터들은 모두 정수형으로 인코딩한 뒤 One-hot Vector 표현으로 변환한다.
4. Nominal 데이터들에 대해서는 [0-255]의 이미지 색상 채널값을 특성으로 가져야 하기 때문에, Min-max Scaler를 통해 [0-255] 범위로 데이터를 정규화한다.
5. Binary는 0과 1로 구성된 데이터로 별다른 전처리를 수행하지 않고 사용한다.

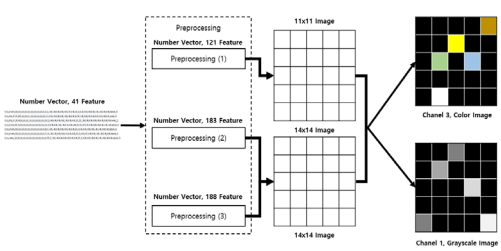
6-1. Continuous 속성만 KBinDiscretizer 알고리즘을 통해 전처리를 완료한다.

6-2. Continuous 속성과 Discrete 속성 모두 KBinDiscretizer 알고리즘을 통해 전처리를 완료한다.

전처리 과정을 통해 라벨을 제외한 41개의 속성이 183개(Continuous), 189개(Continuous & Discrete)로 가공된다. 예측 변수에 대한 성능을 높이기 위한 전처리 단계가 마무리 되면, CNN 모델에 학습하기 위해서 데이터를 이미지로 변경하는 작업을 진행한다.

3.3 네트워크 트래픽 이미지 변환

3.2절의 전처리 과정이 완료된 데이터는 이미지 색상 채널로 표현 가능한 [0-255]의 범위를 가지는 데이터로 변환되어 있으며, 이를 합성곱(Convolution) 연산으로 처리하기 위한 픽셀(pixel) 이미지와 같이 변환하기 위해서 그림 4와 같이 정사각형 행렬(Square Matrix)로 변환한다.



(Figure 4) Network Traffic to Image

Min et al의 전처리 방법은 121개의 속성 값으로 11x11의 정사각형 행렬이 완성되지만, 본 연구에서 제안된 2개의 전처리 방법은 각각 183, 189개의 속성으로 가장 인접한 14x14 정사각형 행렬을 생성하고 빈칸은 Zero-Padding으로 채워 넣는다. 정사각형 행렬로 매칭되는 과정에서 학습에 사용될 두 가지 유형의 이미지 데이터를 생성하게 한다. 첫번째는 RGB(Red, Green, Blue) 3개의 색상 채널을 가지는 Color 이미지로, 3개의 컬러가 중첩되어 MxNx3의 픽셀 배열로 매칭된다. 두 번째는 Grayscale 1개의 색상 채널을 가지는 이미지를 생성하며, MxNx1의 픽셀 배열로 매칭된다. 최종적으로 전처리 과정이 마무리 되면 표5과 같은 학습 및 평가 데이터 세트가 구성된다. M-1과 M-2의 경우 Min et al. 방법을 통해 가공된 데이터이며, D-1, D-2, CD-1 및 CD-2의 경우 본 논문에서 제안된 전처리 방법을 통해 가공된 데이터이다.

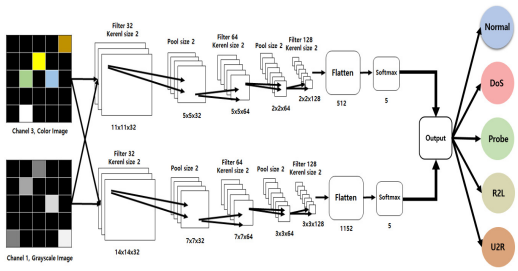
(Table 5) Dataset Processed through Preprocessing

Name	Method	Channel	Feature
M-1	Min et al	Grayscale	121
M-2		Color	
D-1	Discretization of Continuous Feature	Grayscale	183
D-2		Color	
CD-1	Discretization of Continuous and Discrete Feature	Grayscale	189
CD-2		Color	

3.4. CNN 학습 모델

CNN은 합성곱(Convolution)을 사용하는 변형된 신경망으로 데이터의 특징 표현을 학습하는 것을 목표로 하며, 딥러닝의 가장 기본적인 신경망인 DNN에 비해 다음과 같은 차이점이 존재한다. CNN은 각 계층(Layer)에 서로 다른 Feature Map을 생성하여 많은 컨볼루션 커널(Convolution Kernel)을 구성할 수 있다. 계층의 인접한 뉴런(Neuron)의 각 영역은 다음 계층의 Feature Map의 뉴런과 연결되기 때문에, 입력의 모든 공간에서 커널을 공유할 수 있다. CNN의 기본적인 계층 구성은 합성곱 계층과 폴링 계층(Pool Layer)으로 구성되며, 학습 용도에 따라서 완전 연결 계층 (Fully Connected Layer)를 사용하여 분류 및 속성간 거리 계산과 같은 다양한 출력을 사용할 수 있다. 풀링 레이어는 합성곱 계층간 연결되는 파라미터를 줄이므로 연산량을 줄이며, 후속으로 이어지는 합성곱

계층의 수용 필드를 개선한다. 이러한 특성을 가지는 CNN은 이미지 및 신호처리 분야에서 높은 성능을 보여주고 있기에, 본 연구에서는 이미지로 변경된 네트워크 트래픽 데이터에 대한 성능 평가 모델로 선택하여 설계를 진행한다.



(Figure 5) Architecture of Proposed CNN Model

(Table 6) Parameters of Proposed CNN Model

Layer	Parameter	Shape
Input	11, 11, 1 or 3 14, 14, 1 or 3	
Conv_layer 1	Fileter : 32, Kernel : 2, Stride : 1,	11, 11, 32 14, 14, 32
Leaky_Relu	Alpha=0.3	
Max Pooling	Pool size : 2	5, 5, 32 7, 7, 32
Dropout	Rate : 0.2	
Conv_layer 2	Fileter : 64, Kernel : 2, Stride : 1,	5, 5, 64 7, 7, 64
Leaky_Relu	Alpha=0.3	
Max Pooling	Pool size : 2	2, 2, 64 3, 3, 64
Dropout	Rate : 0.2	
Conv_layer 3	Fileter : 128, Kernel : 2, Stride : 1,	2, 2, 128 3, 3, 128
Leaky_Relu	Alpha=0.3	
Flatten	Flattens Input	512 1152
Output	5, Softmax	

제안된 모델은 기본적인 CNN 구조를 가지고 있으며, 모델을 구성할때 사용되는 각 계층의 구조와 파라미터는 그림 5 와 표 6을 통해 명시된다. 모델의 입력으로는 Grayscale 또는 Color 이미지를 입력 데이터로 받아 사용하며, 출력 계층의 Softmax 함수를 통해 정상과 공격 5개의 라벨에 대한 다중 클래스 분류를 통해 성능 평가를 진행한다.

4. 실 험

실험은 3절에서 제안된 전처리 방법과 설계된 모델에 대한 성능과 기존 연구의 성능 비교를 통해 진행된다.

4.1 실험 환경

성능 비교를 위해 학습에 사용되는 데이터는 2.1절 관련 연구의 NSL-KDD 데이터 세트의 KDDTrain+(125,973) 및 KDDTest+ (22,544)를 사용하며, 테이블 형식으로 구성된 데이터를 3.2절과 3.3절에서 제안된 전처리 방식을 통해서 가공된 데이터 세트를 사용하여 진행한다. 3.4절에 설계된 학습 모델을 구성하기 위한 실험 환경과 학습 매개변수(Hyper Parameter)는 표 7, 8을 통해 제시된다.

(Table 7) Experiment Environment

Environment	Name
Language	Python
Library	Tensorflow 2.1.0 Keras 2.2.4-tf
GPU	Nvidia Geforce RTX 2070 Super
Memory	64GB

(Table 8) Parameters used to train CNN

Parameter	Value
Epoch	50
Batch	256
Optimizer	Adam
Learning Rate	2e-4
Loss	Sparse Categorical crossentropy

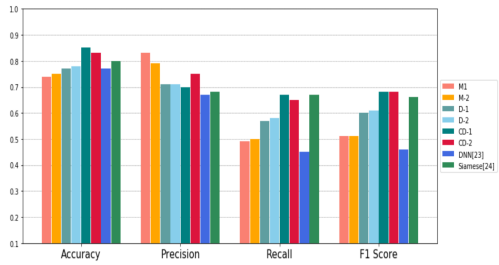
정량화된 성능을 평가하기 위해 표 9에서 제시된 기계학습과 딥러닝 분류 모델에서 일반적인 성능 척도로 사용하는 4개의 성능 지표인 정확도, 정밀도, 재현율, 조화평균을 사용하여 성능 평가를 진행한다.

(Table 9) Experiment Performance Matrix

Matrix	Equation
Accuracy (정확도)	$\frac{TP+TN}{TP+FN+FP+TN}$
Precision (정밀도)	$\frac{TP}{TP+FP}$
Recall (재현율)	$\frac{TP}{TP+FN}$
F1-Score (조화평균)	$2*\frac{Precision*Recall}{Precision+Recall}$

4.2 실험 결과

각 성능지표에 대한 실험 결과는 그림 6~9와 표 10~13에 제시되어 있으며, 그림 6과 표 10의 경우 전체 성능에 대한 평균치와 정확도를 보여준다. 4개의 평균지표 모두에서 CD-1 데이터를 사용하여 학습한 모델의 성능이 가장 높게 나온 것을 확인 할 수 있으며, Grayscale과 Color로 구분되는 CD-2도 유사한 성능이 나오는 것을 확인할 수 있다. 이는 다른 전처리 방법에서도 공통적으로 나오는 것으로, Grayscale과 Color에 대한 성능 차이가 크게 차이를 보이지 않는 것을 확인할 수 있었다.

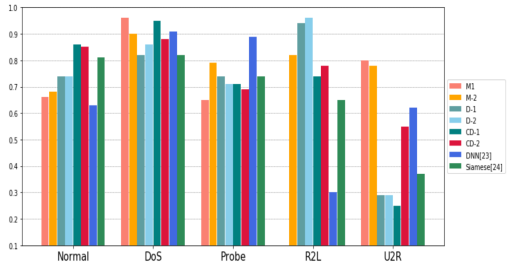


(Figure 6) Experimental Results : Accuracy & Average

(Table 10) Detailed Results : Accuracy & Average

No	Performance			
	Accuracy	Precision	Recall	F1
M-1	74%	83%	49%	51%
M-2	75%	79%	50%	51%
D-1	77%	71%	57%	60%
D-2	78%	71%	58%	61%
CD-1	85%	70%	67%	68%
CD-2	83%	75%	65%	68%
DNN [23]	77%	67%	45.6%	46.8%
Siamese [24]	80%	68%	67%	66%

그림 7과 표 11은 개별 클래스에 대한 정밀도 성능이 나열되어 있으며, 정답으로 예측한 대상에서 예측과 실제 값이 정답으로 일치하는 데이터 비율을 나타낸다. 이는 침입 탐지의 관점에서 거짓 양성(False Positive)에 해당하며, 정상적인 네트워크 트래픽을 공격 트래픽으로 분류하는 비율이다. 거짓 양성은 침입 탐지에서 공격을 정상으로 잘못 분류하는 것에 비해서 낮은 우선순위를 가지기에 뒤의 나오는 거짓 음성(False Negative) 성능을 중요시 한다. 정밀도에 대한 클래스 별로 각 모델의 결과가 상이하게 나오는 것을 볼 수 있으며, 학습 데이터 표본이 적은 R2L 및 U2R의 경우 D-2와 M-1 데이터를 학습한 모델에서 가장 좋은 성능을 보이는 것을 확인 할 수 있다.

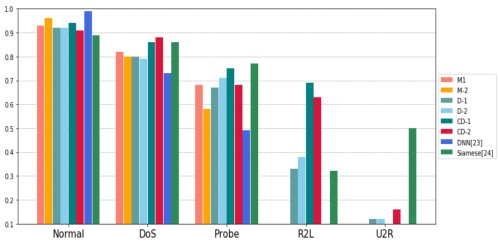


(Figure 7) Experimental Results : Precision

(Table 11) Detailed Results : Precision

No	Precision				
	Normal	DoS	Probe	R2L	U2R
M-1	65%	93%	77%	93%	86%
M-2	68%	90%	79%	82%	78%
D-1	74%	82%	74%	94%	29%
D-2	74%	86%	71%	96%	29%
CD-1	86%	95%	71%	74%	25%
CD-2	85%	88%	69%	78%	55%
DNN [23]	63%	91%	89%	30%	62%
Siamese [24]	81%	82%	74%	65%	37%

그림 8과 표 12는 개별 클래스에 대한 재현율 성능이 나열되어 있으며, 실제 정답인 것 중에서 모델이 정답으로 예측한 비율로, 침입 탐지 관점에서 거짓 음성으로 공격을 정상으로 잘못 탐지하는 것에 대한 비율을 나타낸다. 정밀도와 상관되는 개념이지만, 성능 평가 관점에 따라 중요시 하는 지표가 달라질 수 있기에 2개의 성능 지표가 모두 높을수록 가장 좋은 모델로 볼 수 있다. 재현율에 대한 클래스 별 성능을 확인해 보면, 데이터 표본이 적은 R2L 및 U2R 클래스의 경우 몇몇 모델에서는 전혀 분류되지 않지만 CD-1 모델과 Siamese 모델에서 각각 R2L과 U2R 모델에서 69%, 50%로 다른 모델들에 비해서 좋은 성능을 보여 준다.



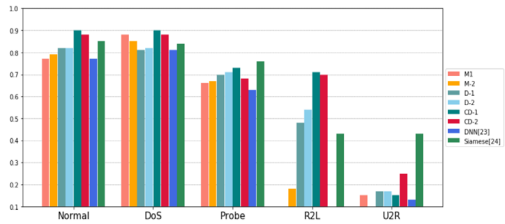
(Figure 8) Experimental Results : Recall

(Table 12) Detailed Results : Recall

No	Recall				
	Normal	DoS	Probe	R2L	U2R
M-1	95%	77%	59%	10%	3%
M-2	96%	80%	58%	10%	4%
D-1	92%	80%	67%	33%	12%
D-2	92%	79%	71%	38%	12%
CD-1	94%	86%	75%	69%	10%
CD-2	91%	88%	68%	63%	16%
DNN [23]	99%	73%	49%	0%	7%
Siamese [24]	89%	86%	77%	32%	50%

그림 9와 표 13은 개별 클래스의 조화평균 성능이 나열되어 있으며, 클래스가 불균형한 구조를 가진 경우 정확한 평가를 내릴 수 있는 평가 지표이다. 정밀도와 재현율에 대한 우선 순위가 있더라도 2개의 지표가 모두 중요한 영향을 끼칠 경우 어느 한쪽으로도 치우치지 않는 평가 값으로 가장 효과적인 성능지표로 사용될 수 있다. 이는 결국 정밀도와 재현율은 서로 상호보완적 관계(Trade-off)이기 때문에, 두 지표가 모두 높을수록 조화 평균값 또한 높은 성능을 보이게 된다. 조화평균에 대한 클래스 별 성능을 살펴 보면, U2R 클래스를 제외한 4개의 클래스에서 CD-1 데이터를 이용한 모델에서 가장 좋은 성능을 보이는 것을 알 수 있다. 특히 R2L 클래스와 같은 학습 데이터 표본이 작은 데이터에서 71%로 매우 좋은 성능이 나온 것을 확인 할 수가 있었다. 하지만 더 적은 학습 표본을 가지는 U2R의 경우, 너무 작은 학습 표본으로 명확한 특징을 추출하지 못한것으로 예상된다.

실험 결과 데이터에 대해 이미지로 변환하여 처리하는 M-1, M-2, D-1, D-2, CD-1, CD-2 알고리즘에서 일반적으로 네트워크 트래픽을 학습하는 알고리즘에 비해서 성능이 개선되는 것을 볼 수 있었다. 그 중에서 Continuous와 Discrete 변수에 대해 이산화 전처리를 적용한 알고리



(Figure 9) Experimental Results : F1-score

(Table 13) Detailed Results : F1-score

No	F1-Score				
	Normal	DoS	Probe	R2L	U2R
M-1	77%	84%	67%	19%	6%
M-2	79%	85%	67%	18%	7%
D-1	82%	81%	70%	48%	17%
D-2	82%	82%	71%	54%	17%
CD-1	90%	90%	73%	71%	15%
CD-2	88%	88%	68%	70%	25%
DNN [23]	77%	81%	63%	0%	13%
Siamese [24]	85%	84%	76%	43%	43%

즘인 CD-1과 CD-2는 R2L 클래스에 대한 조화평균 값이 71%, 70%로 매우 높게 상승하였다. 이는 기존 R2L 클래스가 Continuous와 Discrete 변수에 대해서 비선형 관계를 유지하고 있던 것을 이산화 전처리를 통해서 모델이 해석가능한 선형 관계로 변환되었기 때문이다. 또한 조화평균 성능의 경우 Precision과 Recall에 영향을 받는 수치로, 서로, 서로 상호보완적인 관계를 가지는 두 개의 성능 지표의 변화로 인해 성능 차이가 발생되는 것을 볼 수 있다. 4.2절에서 설명한 각 성능 지표의 특징을 살펴보면 Precision과 Recall은 서로 상호보완적인 관계이므로, CD-1와 CD-2의 Recall 성능이 M-1과 M-2에 비해 향상되는것에 따라 Precision의 성능이 줄어드는 것을 확인할 수 있다. 하지만 가장 적은 데이터 표본과 테스트 데이터의 비율이 더 큰 특징을 가지는 U2R에 대해서는 Continuous와 Discrete 변수에 대한 상관 관계를 통해서 성능을 개선과 성능 평가 결과치에 따른 증감 비율이 비정상적으로 나오는 것을 확인할 수 있었다. 이는 학습 알고리즘(CNN)이 U2R 클래스를 가지는 데이터 표본에 대해서 정상적인 학습이 안되는 것으로 보이며, Siamese와 같은 One-shot Image Recongnition Task를 사용한 학습 방법에서 해당 클래스가 크게 개선되는 것을 확인할 수 있었다.

5. 결 론

본 연구에서는 정상적인 네트워크 트래픽과 악의적인 네트워크 트래픽을 탐지하기 위해 CNN 기반의 NIDS 모델 설계했다. 설계된 모델에 대해 학습 및 평가를 위해 대표적인 네트워크 트래픽 데이터인 NSL-KDD 데이터를 사용하였으며, 예측 변수에 대한 분류 성능을 높이기 위해 기존 기계학습 전처리 방식에 연속 변수 이산화 알고리즘을 추가했다. 기존 기계 학습은 Min et al. 에서 제안된 방식을 사용하였으며, 이전 연구를 통해 분석된 NSL-KDD의 다양한 속성에 대해 이산화 알고리즘을 통해 예측 변수에 대한 모델의 해석 능력을 증가시켰다. 또한 CNN 알고리즘에 학습하기 위해서 네트워크 트래픽을 정사각형 행렬에 맵핑하여 이미지 픽셀 구조로 변환 하였다. 학습된 모델을 평가하기 위해서 기계 학습과 딥 러닝 평가에 자주 사용되는 정확도, 정밀도, 재현율, 조화평균 4개의 지표를 사용하였으며, 본 연구에서 제안된 전처리를 통해 생성된 CD-1을 이용하여 학습된 CNN 모델에서 85%, 70%, 67%, 68%로 전체적으로 가장 좋은 성능을 보였다. 특히 적은 학습 표본을 지니는 R2L 클래스에 대해서 조화평균 값이 71%로 다른 연구 보다 월등한 성능을 보이는 것으로, 기존 비선형 관계를 가지는 R2L 클래스에 대해서 성능 개선을 확인할 수 있었다. 하지만 가장 적은 학습 표본을 지니는 U2R 클래스의 경우 기존의 연구 성능 만큼의 개선을 보이지 않으며, 이를 통해 U2R 클래스에 개선 추세를 보이는 One-shot Image Recongnition Task 학습 방식과 제안된 전처리 방식을 사용하여 더 나은 연구 성과를 보일 것으로 기대된다.

References

[1] Il Seok Oh, Seok-Yun Lee, “Information Security : A Study on cost damage of Cyber Attacks and their Impact on Stock Market,” The KIPS Transactions:PartC, Vol. 13C, Issue 1, 63-68, 2006. https://doi.org/10.3745/KIPSTC.2006.13C.1.063

[2] F. Amiri, R. Y. Mohammad, L. Caro, S. Azadeh and Y. Nasser, "Mutual Information - Based Feature Selection for Intrusion Detection System", Journal of Network and Computer Applications, vol. 34, pp. 1184-1199, 2011. https://doi.org/10.1016/j.jnca.2011.01.002

[3] Raghunath, Bane Raman, and Shivsharan Nitin Mahadeo. "Network intrusion detection system (NIDS)." 2008 First International Conference on Emerging Trends in Engineering and Technology. IEEE, 2008. https://doi.org/10.1109/icetet. 2008.252

[4] Alhomoud, Adeeb, et al. "Performance evaluation study of intrusion detection systems," Procedia Computer Science 5, 173-180, 2011. https://doi.org/10.1016/j.procs.2011.07.024

[5] Yu Su, Kaiyue Qi, Chong Di, Yinghua Ma, and Shenghong Li, “Learning Automata based Feature Selection for Network Traffic Intrusion Detection,” 2018 IEEE Third International Conference on Data Science in Cyberspace, pp.622-627, 2018. https://doi.org/10.1109/DSC.2018.00099

[6] Liao, Hung-Jen, et al. "Intrusion detection system: A comprehensive review." Journal of Network and Computer Applications 36.1, 16-24, 2013. https://doi.org/10.1016/j.jnca.2012.09.004

[7] Casas, Pedro, Johan Mazel, and Philippe Owezarski. "Unsupervised network intrusion detection systems: Detecting the unknown without knowledge." Computer Communications 35.7, 772-783, 2012. https://doi.org/10.1016/j.comcom.2012.01.016

[8] Min, Byeongjun, Dongkyoo Shin, and Dongil Shin. "Network intrusion detection Model through Hybrid Feature Selection and Data Balancing." Proceedings of the Korea Information Processing Society Conference. Korea Information Processing Society, 2020. https://doi.org/10.3745/PKIPS.y2020m05a.526

[9] KDD. KDD CUP. Available online: https://kdd.ics.uci.edu/databases/kddcup99/task.html (accessed on 17 March 2020).

[10] McHugh, John. "Testing intrusion detection systems: a critique of the 1998 and 1999 darpa intrusion detection system evaluations as performed by lincoln laboratory." ACM Transactions on Information and System Security (TISSEC) 3.4, 262-294, 2000. https://doi.org/10.1145/382912.382923

[11] Rodda, Sireesha, and Uma Shankar Rao Erothi. "Class imbalance problem in the network intrusion detection systems," 2016 international conference on electrical,

electronics, and optimization techniques (ICEEOT). IEEE, 2016.
https://doi.org/10.1109/ICEEOT.2016.7755181

[12] Laskov, Pavel, et al. "Learning intrusion detection: supervised or unsupervised?" International Conference on Image Analysis and Processing. Springer, Berlin, Heidelberg, 2005.
https://doi.org/10.100 7/11553595_6

[13] H. S. Chae, B. O. Jo, S. H. Choi, and T. K. Park, "Feature Selection for Intrusion Detection using NSL-KDD," Recent Advances in Computer Science, pp. 184 – 187, 2013.
http://www.wseas.us/e-library/conferences/2013/Nanjing/ACCIS/ACCIS-30.pdf

[14] Sabhnani M., Serpen G., "Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset within Misuse Detection Context." In Proceedings of the International Conference on Machine Learning; Models, Technologies and Applications, pp. 209 – 215, Las Vegas, NV, USA, 23 – 26 June 2003.
https://neuro.bstu.by/ai/To-dom/My_research/Papers-0/For-research/D-mining/Anomaly-D/KDD-cup-99/CD4/mlmta03.pdf

[15] Yao, J.T., Zhao S., Fan L. "An enhanced support vector machine model for intrusion detection," in Proceedings of the International Conference on Rough Sets and Knowledge Technology, pp. 538 – 543, Chongqing, China, 24 – 26 July 2006.
https://doi.org/10.1007/11795131_78

[16] Guan, Y., Ghorbani, A.A., Belacel, N.: "Y-means: a clustering method for intrusion detection," in Canadian Conference on Electrical and Computer Engineering, IEEE CCECE 2003, vol. 2, pp. 1083 – 1086, IEEE, 2003. https://doi.org/10.1109/CCECE.2003.1226084

[17] Lakhina, Shilpa, Sini Joseph, and Bhupendra Verma, Feature reduction using principal component analysis for effective anomaly – based intrusion detection on NSL-KDD, 2010.
http://www.ijest.info/docs/IJEST10-02-06-56.pdf

[18] Javaid, Ahmad, et al. "A deep learning approach for network intrusion detection system," in Proceedings of the 9th EAI International Conference on Bio-inspired Information and Communications Technologies (formerly BIONETICS). 2016.
https://doi.org/10.4108/eai.3-12-2015.2262516

[19] Potluri, Sasanka, and Christian Diedrich. "Accelerated deep neural networks for enhanced intrusion detection system," 2016 IEEE 21st international conference on emerging technologies and factory automation (ETFA), IEEE, 2016.
https://doi.org/10.1109/ETFA.2016.7733515

[20] Torres, P., Catania, C., Garcia, S., Garino, C.G., "An analysis of recurrent neural networks for botnet detection behavior," in 2016 IEEE Biennial Congress of Argentina (ARGENCON), pp. 1 – 6, IEEE, 2016.
https://doi.org/10.1109/ARGENCON.2016.7585247

[21] Imamverdiyev, Yadigar, and Fargana Abdullayeva, "Deep learning method for denial of service attack detection based on restricted boltzmann machine," Big Data 6.2, 159-169, 2018.
https://doi.org/10.1089/big.2018.0023

[22] Lopez-Martin, Manuel, Belen Carro, and Antonio Sanchez-Esguevillas, "Application of deep reinforcement learning to intrusion detection for supervised problems," Expert Systems with Applications 141, 112963, 2020.
https://doi.org/10.1016/j.eswa.2019.112963

[23] Gao, Minghui, et al. "Malicious Network Traffic Detection Based on Deep Neural Networks and Association Analysis." *Sensors* 20.5, 1452, 2020.
https://doi.org/10.3390/s20051452

[24] Bedi, Punam, Neha Gupta, and Vinita Jindal. "I-SiamIDS: an improved Siam-IDS for handling class imbalance in network-based intrusion detection systems," Applied Intelligence, 1-19, 2020.
https://doi.org/10.1007/s10489-020-01886-y

◐ 저 자 소 개 ◑

유 지 훈(Jihoon Yoo)
2018년 세종대학교 대학원 컴퓨터공학과 (공학석사)
2019년~현재 세종대학교 대학원 박사과정
관심분야: 분산 처리, 데이터 마이닝, 딥러닝, etc
E-mail: yoojihoon@sju.ac.kr

민 병 준(Byeongjun Min)
2019년 세종대학교 대학원 컴퓨터공학과 (공학석사)
2019년~현재 세종대학교 대학원 박사과정
관심분야: 강화 학습, 이상 탐지, 딥러닝, etc
E-mail : bang@sju.ac.kr

김 상 수(Sangsoo Kim)
2003년 경북대학교 대학원 컴퓨터공학과(공학석사)
관심분야: 사이버전 기술, 위협 헌팅, 사이버 상황인식, etc
E-mail : wisdory@naver.com

신 동 일(Dongil Shin)
1988년 연세대학교 컴퓨터 과학과 (공학사)
1993년 Washington State University 컴퓨터과학과 (공학석사)
1997년 North Texas University 컴퓨터과학과 (공학박사)
1998년~현재 세종대학교 컴퓨터공학과 교수
관심분야: 정보 보안, 기계 학습, 데이터 마이닝, 생체 데이터 처리, etc
E-mail : dshin@sejong.ac.kr

신 동 규(Dongkyoo Shin)
1986년 서울대학교 계산통계학과 (공학사)
1992년 Illinois Institute of Technology 컴퓨터과학과(공학석사)
1997년 Texas A&M University 컴퓨터과학과(공학박사)
1998년~현재 세종대학교 컴퓨터공학과 교수
관심분야: 정보 보안, 기계 학습, 데이터 마이닝, 생체 데이터 처리
E-mail : shindk@sejong.ac.kr