Journal of Next-generation Convergence Information Services Technology Vol.10, No.2, April (2021), pp.123-132
ISSN 2384-101X(p) 2672-1163(e) *http://dx.doi.org/10.29056/jncist.2021.04.02*

합성곱신경망을 이용한 KDD Cup 1999 데이터분석

Analysis for the KDD Cup 1999 Data Using the Convolutional Neural Network

이석준[1], 심동희[2]*

Sukjoon Lee[1], Donghee Shim[2]*

요 약

KDD Cup 1999 데이터는 네트워크침입탐지 연구에 많이 이용된 데이터셋이다. 본 연구에서는 합성곱신경망을 이용해서 이 데이터를 분석하였다. 그 동안 이 데이터에 대한 TASVM, 의사결정트리 알고리즘, 랜덤포레스트알고리즘 등의 인공지능기법을 이용한 많은 분석이 이루어졌다. 그리고 다층신경망을 통한 분석도 이루어져왔지만 합성곱신경망을 이용한 접근은 거의 없었다. 합성곱신경망중 LeNet-5를 이용하여 Keras를 사용해서 분석을 하였다. 첫 번째 방법에서는 KDD Cup 1999의 10%훈련셋을 훈련셋과 테스트셋으로 나누어 분석하였다. 그리고 두 번째 방법에서는 10%훈련셋으로는 훈련하고 테스트는 테스트셋을 사용하여 분석하였다. 그리고 이 두 가지 방법의 결과를 캐글 웹사이트에 올려진 다층신경망과 비교하였다. KDD Cup 1999 데이터에는 침입유형을 크게 4가지로 분류하는데 각 유형별로 성능척도를 계산하였다. 그 결과 2가지 방법 모두에서 LeNet-5가 캐글에 올려진 다층신경망보다 우수한 성능을 나타냈다.

핵심어 : KDD Cup 1999 데이터, 합성곱신경망, LeNet-5, Keras

Abstract

KDD Cup 1999 data has been used for the network intrusion detection studies. This data is analyzed using the convolutional neural networks in this paper. Many researchers have analyzed this data using the aritificial intelligence techniques such as TAVSM, decision tree algorithm and random forest algorithm etc. Although multilayer neural networks also have been used in these analysis, convolutional neural networks models have not been used. LeNet-5 among the various convolutional neural networks models is used with Keras for this analysis. At the first method, the 10% training set of the KDD Cup 1999 dividing into training set and testing set is used. At the second method, the 10% training set is used for train and testing set is used for test. The results are compared with the result of analysis using the mutlilayer neural networks uploaded in Kaggle web site. Because attacks are classified into 4 types in KDD Cup 1999 data, the performance measures are calculated by each attack types. LeNet-5 shows the better perfomance than the multilayer neural networks uploaded to Kaggle web site at both methods in all performance measures.

Keyword : KDD Cup 1999 Data, Convolutional Neural Networks, LeNet-5, Keras

1 Department of Carbon Convergence Engineering, Jeonju University, Jeonju, Korea [Graduate Student]
e-mail: qbox3d@gmail.com

2 Department of Computer Science & Engineering, Jeonju University, Jeonju, Korea [Professor]
e-mail: dhshim@jj.ac.kr (Corresponding author)

Received(March 3, 2021), Review Result(1st: March 17, 2021), Accepted(April 9, 2021), Published(April 30, 2021)

1. 서론

MIT 대학에서 수행한 DARPA(Defence Advanced Research Projects Agency) 1998 침입탐지시스템 평가 프로그램에서 수집된 데이터를 기반으로 만들어진 데이터집합이 바로 KDD Cup 1999이다 [1]. 이 데이터집합은 네트워크침입탐지시스템 등 네트워크보안 분야에서 아주 유용하게 사용되고 있다. 이 데이터는 지속기간, 프로토콜종류 등 총 42개의 속성으로 구성되어 있다. 이 자료를 이용하여 컴퓨터네트워크의 이상현상 탐지 및 분류 등 네트워크보안 연구가 많이 이루어졌다 [2-10]. 그리하여 이 데이터를 분석하기 위하여 인공신경망, 의사결정트리, k-최근접이웃(k-Nearest Neighbors) 알고리즘, Naive Bayes 알고리즘, 랜덤포리스트 등의 방법이 사용되었다 [2-10]. 이 인공지능적인 방법에서 인공신경망을 이용한 연구 [2-4][7-10]도 있었지만 일반적인 다중계층신경망을 이용하였으며 합성공신경망(CNN:Convolutional Neural Networks) [11]을 이용한 분석은 거의 없었다.

본 논문에서는 KDD Cup 1999 데이터에 대하여 Keras 라이브러리 [12]를 이용하여 합성공신경망에 해당하는 LeNet-5 [11]로 모델링하여 구축하였다. 데이터를 이용하여 훈련을 거친 후에 성능평가를 하였다. 성능평가결과 다른 인공신경망에서의 성능보다 더 좋은 성능을 나타냈다.

본 연구의 2장에서는 KDD Cup 1999 데이터 분석에 사용된 기법들을 간략히 소개하고, 3장에서는 데이터 설명, LeNet-5 모델구축에 대하여 설명하고, 4장에서는 구현과 성능평가 결과를 나타냈으며, 5장에서는 결론을 기술하였다.

2. KDD Cup 1999 데이터에 대한 분석방법

일반적으로 기계학습에서는 데이터에 대한 검토, 데이터에 대해 부수적으로 필요한 사전처리, 학습모델 및 알고리즘선정, 학습실행과 성능평가의 절차로 이루어진다 [13]. 특히 데이터에 대한 사전처리에서는 형태변경, 누락 경우에 대한 처리, 표준화 등의 작업을 한다.

KDD Cup 1999 데이터에 대한 연구가 많이 이루어졌는데 많이 사용된 방법을 간단히 살펴본다. 많은 연구에서 인공신경망을 이용하였는데 인공신경망 [14]은 네트워크를 형성한 노드들이 학습을 통해 시냅스의 결합 세기를 변화시켜, 문제해결 능력을 가지는 모델 전반을 가리킨다. 각 층은 이러한 노드들로 구성되는 데, 층은 입력층과 중간의 은닉층, 출력층으로 구성된다. 중간의 은닉층이 여러 개인 경우를 다층신경망(Mltilayer Neural Networks)라고 한다.

Tang외 2인 [2]의 연구에서는 정보획득방법을 이용해서 특성을 감축한 후 TASVM(Triangle Area Based Support Vector Machine)을 이용하였다. TASVM은 k-평균 군집화와 SVM을 결합해서 사용한 방법이다. k-평균 군집화 [15]는 입력받은 데이터를 k개의 그룹으로 묶는 데 각 그룹의 중심과 데이터간의 거리차이를 최소화하는 방식으로 작동한다. SVM(Support Vector Machine) 알고리즘 [16]

은 두 카테고리 중 어느 하나에 속한 데이터 집합이 주어졌을 때, 주어진 데이터 집합을 바탕으로 하여 새로운 데이터가 어느 카테고리에 속할 지 판단하는 비확률적 이진선형분류 모델을 만든다. 만들어진 분류모델은 데이터가 사상된 공간에서 경계로 표현되는 데 그 중 가장 큰 폭을 가진 경계를 찾는 알고리즘이다. 선형분류와 더불어 비선형 분류에서도 사용될 수 있다. 비선형분류를 하기 위해서 주어진 데이터를 고차원 특징 공간으로 사상하는 작업이 필요한 데, 이를 효율적으로 하기 위해 커널 트릭을 사용하기도 한다. Devarajud외 1인의 연구 [3]에서는 인공신경망중 FFNN(Feed Forward Neural Network), PNN(Probabilistic Neural Network) 그리고 RBNN(Radial Basis Neural Network)을 사용했으며 소프트웨어는 MATLAB을 이용했다. FFNN은 신경회로망 단위간의 연결이 주기를 형성하지 않은 것으로서 정보는 입력노드에서 앞으로만 이동하며 망에서 순환이나 루프가 없다. PNN은 FFNN의 일종으로서 각 클래스의 부모확률분포함수가 Parzen창과 비모수함수에 의해서 근사추정된다. RBNN은 은닉노드가 1계층으로만 구성되는데 이를 특성벡터라고 부른다. 그리고 이 특성벡터에 비선형변환함수를 적용하여 분류문제에 적용하는데 특성벡터의 차원을 증가시키면 선형분리능력이 증가한다.

Shah외 1인 연구 [4]에서는 정보획득방법을 이용해서 특성을 감축한 후 BPNN(Back Propagation Neural Network)을 이용했다. BPNN에서는 최종출력값과 실제값의 오차가 최소가 되도록 각 층에서 가중치와 편차를 계산하면서 순전파와 역전파를 반복한다. Shi외 1인 연구 [5]에서는 sGHSOM을 이용했다. 신경망알고리즘에 해당하는 GHSOM(Growing Hierarchical Self-Organizing Maps)은 구조적인 클러스터링을 제공하는데 계층적으로 너무 커지는 단점이 있었다. GHSOM의 이런 단점을 해결하는 향상된 것이 바로 sGHSOM이다. Li 연구 [6]에서는 의사결정트리를 이용했다. 의사결정 트리 [17]는 의사결정 규칙과 그 결과들을 트리구조로 도식화한 의사결정 지원도구의 일종으로 변수들 간의 규칙, 관계 등으로 레이블을 분류하는 트리구조의 모델을 생성하고, 관측값을 해당 모델에 대입하여 레이블을 예측하는 방식이다. 이러한 의사결정트리의 알고리즘도 많이 있는 데 ID3와 C4.5를 이용하였다. Alphonsovna외 1인 연구 [7]에서는 랜덤포레스트알고리즘을 이용하여 분석하였다. Roempluk 연구 [8]에서는 KDD Cup 1999로부터 레코드수를 감소시킨 NSL-KDD [18]를 같이 이용했는데 k-최근접이웃(k-Nearest Neighbors) 알고리즘, 다층신경망 그리고 SVM을 이용해서 분석했다. k-최근접 이웃알고리즘 [19]은 회귀나 분류에 사용되는 비모수 방식에 해당한다. 분류나 회귀 모두 입력이 특징 공간내에서 k개의 가장 가까운 훈련 데이터로 구성되어 있다. 그래서 테스트 데이터 셋의 레이블이 없는 각 레코드에 대해 훈련데이터에서 유사도 기준으로 K개의 가장 가까운 값을 찾는다. 출력은 분류에서는 소속된 항목이며, 회귀에서는 객체의 특성 값이다.

[9], [10]번 연구는 캐글(Kaggle) [20]에 올려진 KDD Cup 1999 데이터를 이용했다. 캐글은 사용자가 해결과제와 데이터를 먼저 등록하면, 이에 대한 해결책으로 다른 사용자들이 예측모델을 개발해서 올림으로써 경쟁하는 플랫폼이다. [9]번에서는 Naive Bayes, 의사결정트리, 랜덤포레스트,

SVM 그리고 다층신경망 등의 여러 방법으로 분석하였는데 랜덤포레스트 알고리즘의 결과가 가장 좋은 성능을 나타냈다. [10]번 연구에서는 GAN(Generative Adversal Network)을 이용했다. GAN은 비지도학습에 많이 사용되는데, 저차원의 무작위 잡음을 받아서 고차원 허구데이터를 생성하는 생성망, 입력된 데이터가 실제인지 생성된 것인지를 판별하는 판별망을 사용한다 [21]. 그래서 판별망은 실제데이터들을 생성망이 만든 생성데이터들과 구분할 수 있도록 학습한다.

3. LeNet-5 모델과 KDD Cup 1999 데이터 구성

3.1 모델의 구성

합성곱신경망은 Yann Lecun 연구팀이 1998년에 LeNet을 발표하면서 처음으로 개발했다 [11]. 그 후 합성곱신경망으로 AlexNet, ZFNet 등이 발표되면서 발전하고 있다. LeNet에도 여러 버전이 있는데 이 연구에서는 LeNet-5를 이용하였다. LeNet-5는 입력층, C1과 C2 그리고 C3의 컨볼루션층 3개, S2와 S4의 서브샘플링층 2개, F6 완전연결층 1개 그리고 출력층으로 구성된다. 이들의 입력층, C1, S2, C3, S4, C5, F6, 출력층의 순서로 구성된다.

3.2 KDD Cup 1999 데이터의 구성

KDD Cup 1999 데이터는 41개의 항목으로 구성되어 있는데 여기에 각 레코드가 어떤 공격인 지를 나타내는 유형을 추가하면 총 42개의 속성으로 이루어진다 [7]. 이들 속성중 공격유형, Protocol_type, flag는 범주형으로 문자로 구분되어 있다. 한편 공격유형은 [표 1]에 나타낸 바와 같이 총 22개로 분류되어 있는 데 이 공격유형은 총 4가지의 대분류 유형으로 분류된다 [7]. 이 4가지 대분류 유형은 바로 U2R(User to Root), DoS(Denial of Service), R2L(Remote to User), 그리고 Probe이며 여기에 정상적인 경우 Normal을 추가해서 5가지의 대분류 유형으로 나누어진다. U2R유형은 공격자가 피해시스템에 일반사용자권한으로 접근한 후 취약점을 이용해서 관리자권한을 획득하여 공격하는 것이다. DoS 유형은 서비스거부공격으로서 공격자가 피해시스템 자원을 점유하여 정당한 사용자들의 접근이 되지 않도록 공격하는 것이다. R2L유형은 공격자가 피해시스템의 취약점을 이용해서 부당한 접근권한을 획득하여 공격하는 것이다. Probe 유형은 공격자가 공격을 위하여 피해시스템의 여러 가지 정보를 획득하는 것에 해당한다.

이 KDD Cup 1999는 3개의 데이터셋로 구성된다. 먼저 가장 큰 전체 KDD 셋는 총 4,898,430개의 레코드가 있는데 이는 훈련셋로 사용된다. 이 훈련셋가 너무 방대하여 이 중에서 약 10%를 추출한 셋를 10% KDD라고 하여 여기에는 494,020개의 레코드로 구성되어 있다. 그리고 시험에 사용되는 시험셋가 있는데 이는 Corrected 셋로 불리는데 311,029개의 레코드로 구성되어 있다. 침입유

형 대분류 기준으로 레코드수를 계산하면 이 3가지 셋 모두에서 DoS, Normal, Probe, R2L 그리고 U2R 순으로 레코드가 포함되어 있다.

[표 1] KDD Cup 1999의 공격유형
[Table 1] Attack Types of KDD Cup 1999

공격의 주요 클래스	22개의 공격클래스
U2R(User to Root)	buffer_overflow, perl, loadmodule, rootkit
DoS(Denial of Service)	back, land, neptune, pod, smurf, teardrop
R2L(Remote to User)	ftp_write, guess_passwd, imap, multihop, phf, spy, warezmaster, warezclient
Probe	ipsweep, nmap, portsweep, satan

3.3 데이터의 전처리

10% KDD 데이터셋의 각 속성들의 값을 확인하였고, 공격유형을 속성에 추가하였으며 22개의 공격클래스를 해당되는 주요클래스 Normal, U2R, DoS, R2L 그리고 Probe에 따라 각각 0부터 4로 변환하였다. protocol_type을 icmp는 0으로, tcp는 1로, udp는 2로 변환하였다. flag는 TCP문자열에 따라서 0부터 9로 변환하였다. 그리고 속성값의 개수중에서 2개 미만인 것은 2개가 있었는데 이는 제외했다. 그리고 속성간의 상관관계를 계산하여 높은 것 9개의 속성을 제거했다. 그래서 합성곱신경망모델에 입력되는 데이터 속성은 모두 31개가 선정되었다. 공격클래스는 지도학습의 레이블(종속변수)에 해당되므로 독립변수는 총 30개가 된다.

시험셋에 대해서도 같은 방법으로 전처리를 하였다. 추가적으로 10% KDD에 나타난 공격유형22개에 해당하는 데이터만 선별해서 가져왔다.

4. 모델의 구현 및 평가

4.1 모델구현

합성곱신경망모델은 입력층, 출력층외에 컨볼루션층 2개, 서브샘플링층 2개, 은닉증 2개로 총 8개 층으로 구현한다. 합성곱에서는 Conv2D, 서브샘플링에서는 MaxPooling2D를 사용한다. 그리고 활성함수로 ReLU를 사용하였으며, 출력층에서는 Softmax를 사용하였다.

모델에서 훈련된 각 층별 파라미터수를 별로 [표 2]에 나타냈는데 C1층에서는 320개, C3층에서는 8256개, F6층에서는 520개, F7층에서는 45로 총 9141개 이다.

[표 2] 훈련된 파라미터 수

[Table 2] Number of Parameters Trained

층	파라미터 갯수 산출식	파라미터 갯수
C1	(필터사이즈*입력맵개수 + 바이어스)*특성맵개수	(3*3*1+1)*32 = 320
C3	(필터사이즈*입력맵개수 + 바이어스)*특성맵개수	(2*2*32+1)*64 = 8256
F6	(입력개수 + 바이어스)*출력개수	(64 +1)*8 = 520
F7	(입력개수 + 바이어스)*출력개수	(8+1)*5=45
누계		**9141**

4.2 평가결과

평가를 위하여 데이터셋을 2가지 방법으로 이용했다. 첫 번째 방법에서는 Kaggle에 올려진 방법과 동일하게 10%데이터셋을 훈련셋과 테스트셋으로 나누어서 사용했다. 두 번째 방법에서는 10% 데이터셋을 훈련에만 적용했고, 성능평가에서는 테스트셋을 사용했다.

첫 번째 방법에서는 10%데이터셋을 잎서 3.2절에서 설명한 바와 같이 전처리한 후 2/3는 훈련셋, 1/3은 테스트셋으로 임의로 분리하였다. 그리하여 훈련셋으로 Kaggle의 다층신경망 [9]과 우리 연구의 합성곱신경망을 각각 훈련하였으며, 테스트셋으로 성능평가를 하였다.

두 번째 방법에서는 테스트셋을 사용했다. 전처리한 10%데이터셋 전체를 이용하여 훈련을 한 후에 테스트셋을 이용해서 성능평가를 했다.

첫번째 방법으로 LeNet-5, Kaggle의 다층신경망을 테스트한 결과를 [표 3]과 [표 4]에 각각 나타냈다. KDD Cup 1999데이터에서의 침입유형의 대분류가 총 5가지이므로 진(True)과 부(False)를 적용하는데 있어서 평가척도 계산이 다소 복잡하다. 그래서 각 유형별로 평가척도를 계산하고 이들의 평균을 다시 산출하였다. 이 평가척도를 [표 5]에 LeNet-5는 침입유형별로 계산한 결과를 포함하여 표시하였으며, Kaggle의 다층신경망에 대해서는 평균치만 표시하였다. 해당 침입유형이 상대적으로 작은 U2R과 R2L에서는 정밀도가 낮은 것을 알 수 있다. 본 연구에서 LeNet-5로 생성한 모델과 Kaggle의 다층신경망을 비교한 결과 평균적으로 정확성은 0.9994와 0.9966로, 정밀도는 0.8654와 0.5995로 나타났다. 모든 평가척도에서 본 연구모델이 더 우수함을 보여주고 있다.

[표 3] LeNet-5의 10%훈련데이터셋 1/3을 이용한 평가결과

[Table 3] Evaluation Result of LeNet-5 using one third of 10% trained data set

구분		예측					
		정상	U2R	DoS	R2L	Probe	소계
실제	정상	1,323	0	2	0	23	1,348
	U2R	1	291	1	1	93	387
	DoS	0	0	129,094	0	12	129,106
	R2L	0	1	0	11	7	19
	Probe	20	29	61	1	32,056	32,167
	소계	1,344	321	129,158	13	32,191	16,3027

[표 4] Kaggle 다층신경망의 10%훈련데이터셋 1/3을 이용한 평가결과

[Table 4] Evaluation Result of Kaggle DNN using one third of 10% trained data set

구분		예측					
		정상	U2R	DoS	R2L	Probe	소계
실제	정상	479	0	542	0	327	1,348
	U2R	75	0	99	0	213	387
	DoS	17	0	129,042	0	47	129,106
	R2L	3	0	3	0	13	19
	Probe	13	0	38	0	32,116	32,167
	소계	587	0	129,724	0	32,716	16,3027

[표 5] 평가척도 비교 1

[Table 5] Comparisions 1 of Performance Measure

평가척도	본 연구						Kaggle 다층신경망
	Normal	DoS	Probe	U2R	R2L	평균	
정확도(Accuracy)	0.9997	0.9995	0.9985	0.9992	0.9999	0.9994	0.9966
정밀도(Precision)	0.9999	0.9999	0.9965	0.7519	0.5789	0.8654	0.5995
민감도(Sensitivity)	0.9998	0.9995	0.9958	0.9065	0.8462	0.9496	-
특이도(Specificity)	0.9844	0.9996	0.9992	0.9994	1.0000	0.9965	0.9622
FPR:False Positive Rate	0.0156	0.0004	0.0008	0.0006	0.0000	0.0035	0.0378
오류율(Error Rate)	0.0003	0.0005	0.0015	0.0008	0.0001	0.0006	0.0034

두 번째 방법에서는 10%데이터셋으로 훈련하고, 테스트셋중 22개 침입유형에 부합하는 234,299건으로 성능평가를 하였다. 이 평가결과를 [표 6]과 [표 7]에 나타냈다. 이에 대한 평가척도를 [표 8]에 LeNet-5는 침입유형별로 계산한 결과를 포함하여 표시하였으며 Kaggle의 다층신경망에 대해서는 평균치만 표시하였다. 해당 침입유형이 상대적으로 작은 U2R과 R2L에서는 정밀도가 낮은 것을 알 수 있다. 본 연구에서 LeNet-5로 생성한 모델과 Kaggle의 다층신경망을 비교한 결과 평균적으로 정확성은 0.9868과 0.9865로, 정밀도는 0.6852와 0.5975로 나타났다. 모든 평가척도에서 우리 모델이 더 우수함을 보여주고있다.

[표 6] LeNet-5의 corrected데이터셋을 이용한 평가결과

[Table 6] Evaluation Result of LeNet-5 using corrected dataset

구분		예측					
		정상	U2R	DoS	R2L	Probe	소계
실제	정상	1,988	0	5	0	384	2,377
	U2R	0	8	385	0	5,600	5,993
	DoS	11	0	164,200	0	1,086	165,297
	R2L	0	0	0	17	22	39
	Probe	166	6	60	4	60,357	60,593
	소계	2,165	14	164,650	21	67,449	234,299

[표 7] Kaggle 다층신경망의 corrected데이터셋을 이용한 평가결과

[Table 7] Evaluation Result of Kaggle DNN using corrected data set

구분		예측					
		정상	U2R	DoS	R2L	Probe	소계
실제	정상	1,987	0	27	0	453	2,377
	U2R	80	0	497	0	5,416	5,993
	DoS	81	0	164,171	0	1,045	165,297
	R2L	3	0	1	0	35	39
	Probe	181	0	70	0	60,342	60,593
	소계	2,242	0	164,776	0	67,291	234,299

[표 8] 평가척도 비교 2

[Table 8] Comparisions 2 of Performance Measure

평가척도	본 연구						Kaggle 다층신경망
	Normal	DoS	Probe	U2R	R2L	평균	
정확도(Accuracy)	0.9976	0.9934	0.9687	0.9744	0.9999	0.9868	0.9865
정밀도(Precision)	0.9992	0.9934	0.9961	0.0013	0.4359	0.6852	0.5795
민감도(Sensitivity)	0.9983	0.9973	0.8949	0.5714	0.8095	0.8543	-
특이도(Specificity)	0.9182	0.9842	0.9986	0.9745	0.9999	0.9751	0.9605
FPR:False Positive Rate	0.0818	0.0158	0.0014	0.0000	0.0000	0.0249	0.0395
오류율(Error Rate)	0.0024	0.0066	0.0313	0.0256	0.0001	0.0132	0.0135

5. 결론

KDD Cup 1999에 대하여 보안분야에서 그 동안 TASVM, 의사결정트리 알고리즘, 랜덤포레스트 알고리즘 등의 인공지능기법을 이용한 많은 분석이 이루어져왔었다. 또한 다층신경망을 이용한 분석도 이루어졌지만 합성곱신경망을 이용한 분석은 거의 없었다.

그리하여 본 연구에서는 이 KDD Cup 1999 데이터에 대하여 합성곱신경망중 LeNet-5를 이용하여 분석하여 이 결과를 Kaggle 사이트에 업로드 [9]된 다층신경망 결과와 비교하였다. KDD Cup 1999 데이터에서는 침입유형을 5개로 대분류한 바 이 기준으로 평가척도를 계산하여 비교하였다. 10%데이터셋을 훈련과 테스트로 나누어 이용한 방법에서는 LeNet-5로 생성한 모델과 Kaggle의 다층신경망을 비교한 결과 정확성은 0.9994와 0.9966로, 정밀도는 0.8654와 0.5995로 나타났다. 10%데이터셋으로 훈련을 하고 테스트셋으로 평가한 방법에서는 LeNet-5로 생성한 모델과 Kaggle의 다층신경망을 비교한 결과 평균적으로 정확성은 0.9868과 0.9865로, 정밀도는 0.6852와 0.5975로 나타났다. 두 방법 모두에서 여러가지 평가척도에서 본 연구모델이 더 우수함을 보여주었다.

향후 연구에서는 다른 형태의 인공신경망을 적용해서 더 비교분석할 필요가 있다.

References

[1] D. Dheeru, G. Casey, "KDD Cup 1999 Data Data Set", archive.ics.uci.edu, https://archive.ics.uci.edu/ml/datasets/KDD+Cup+1999+Data, (accessed November 2, 2020).

[2] P. Tang, R. Jiang, M. Zhao, "Feature Selection and Design of Intrusion Detection System Based on k-Means and Triangle Area Support Vector Machine", 2010 Second International Conference on Future Networks, January 22-24, 2010, Sanya Hainan China, pp. 144-148, doi: 10.1109/ICFN.2010.42.

[3] S. Devaraju, S. Ramakrishnan, "Performance analysis of intrusion detection system using various neural network classifiers", 2011 International Conference on Recent Trends in Information Technology, June 3-5, 2011, Tamil Nadu, India, pp. 1033-1038, doi: 10.1109/ICRTIT.2011.5972289.

[4] B. Shah, B. H. Trivedi, "Reducing Features of KDD CUP 1999 Dataset for Anomaly Detection Using Back Propagation Neural Network", 2015 Fifth International Conference on Advanced Computing & Communication Technologies, February 21-22, 2015, Haryana, India, pp. 247-251, doi: 10.1109/ACCT.2015.131.

[5] H. Shi, H. Xu, "An enhanced GHSOM for the intrusion detection", 11th International Conference on Wireless Communications, Networking and Mobile Computing, September 21-23, 2015, Shanghai, China, pp. 1-5, doi: 10.1049/cp.2015.0756.

[6] M. Li, "Application of CART decision tree combined with PCA algorithm in intrusion detection", 2017 8th IEEE International Conference on Software Engineering and Service Science, November 24-26, 2017, Beijing China, pp. 38-41, doi: 10.1109/ICSESS.2017.8342859.

[7] R. A. Alphonsovna, D. Shim, "An Application of Random Forest Algorithm to Network Intrusion Detection", Journal of Next-generation Convergence Information Services Technology, vol. 8, no. 2, June 2019, pp. 187-202, doi: 10.29056/jncist.2019.06.04.

[7] S. Behera, A. Pradhan, R. Dash, "Deep Neural Network Architecture for Anomaly Based Intrusion Detection System", 2018 5th International Conference on Signal Processing and Integrated Networks, February 22-23, 2018, Noida, India, pp. 270-274, doi: 10.1109/SPIN.2018.8474162.

[8] T. Roempluk, O. Surinta, "A Machine Learning Approach for Detecting Distributed Denial of Service Attacks", 2019 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering, January 30-February 2, 2019, Nan, Thailand, pp. 146-149, doi: 10.1109/ECTI-NCON.2019.8692243.

[9] Y. Byun, "Intrusion Detection System-accuracy", kaggle.com, https://www.kaggle.com/yungbyun/intrusion-detection-system-accuracy-99-9, (accessed November 11, 2020).

[10] S. Fook, "GAN for anomaly detection", kaggle.com, https://www.kaggle.com/sekfook97/gan-for-anomaly-detection/execution, (accessed November 15, 2020).

[11] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition", Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, November 1998, doi: 10.1109/5.726791.

[12] F. Chollet, "About Keras", keras.io, https://keras.io/api, (accessed November 8, 2020).

[13] R. Saravanan, P. Sujatha, “State of Art Techniques on Machine Learning Algorithms: A Perspective of Supervised Learning Approaches in Data Classification”, 2018 International Conference on Intelligent Computing and Control Systems, June 14-15, 2018, Madurai, India, pp. 945-949, doi: 10.1109/ICCONS.2018.8663155.

[14] W. McCulloch, W. Pitts, “Artificial neural network”, wikipedia.org, https://en.wikipedia.org/wiki/Artificial_neural_network, (accessed November 14, 2020).

[15] E. Fix, J. Hodges, “k-means clustering”, wikipedia.org, https://en.wikipedia.org/wiki/K-means_clustering, (accessed November 20, 2020).

[16] V. Vapnik, “Support-vector machine”, wikipedia.org, https://en.wikipedia.org/wiki/Support-vector_machine, (accessed November 14, 2020).

[17] R. Quinlan, “Decision tree”, wikipedia.org, https://en.wikipedia.org/wiki/Decision_tree, (accessed November 13, 2020).

[18] M. Tavallaee, E. Bagheri, W. Lu, A. Ghorbani “NSL-KDD dataset”, unb.ca, https://www.unb.ca/cic/datasets/nsl.html, (accessed November 30, 2020).

[19] E. Fix, J. Hodges, “k-nearest neighbors algorithm”, wikipedia.org, https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm, (accessed November 12, 2020).

[20] A. Goldbloom, B. Hamner, “Kaggle”, kaggle.com, “https://www.kaggle.com/competitions, (accessed November 15 2020).

[21] I. Goodfellow, “Generative adversarial network”, wikipedia.org, https://en.wikipedia.org/wiki/Generative_adversarial_network, (accessed December 5, 2020).