Journal of Digital Convergence
Vol. 16. No. 5, pp. 399-406, 2018

ISSN 1738-1916
https://doi.org/10.14400/JDC.2018.16.5.399

SVM을 이용한 네트워크 기반 침입탐지 시스템에서 새로운 침입탐지에 관한 연구

양은목[1], 서창호[2]
[1]숭실대학교 소프트웨어학부, [2]공주대학교 응용수학과

A Study on Intrusion Detection in Network Intrusion Detection System using SVM

Eun-mok YANG[1], Chang-Ho Seo[2]
[1]School of Software, Soongsil University
[2]Dept. of Applied Mathematics, Kongju National University

요 약 인공지능을 이용한 침입탐지 연구는 KDDCup99 데이터 세트를 사용하여 많은 연구가 이루어졌다. 이전 연구에서 SMO(SVM)알고리즘의 성능이 우수하다고 알려져 있다. 하지만 훈련에 사용되지 않은 새로운 침입유형의 침입탐지연구는 미비하다. 본 논문에서는 웨카(weka)의 SMO와 KDDCup99 훈련 데이터 세트인 kddcup.data.gz의 인스턴스를 이용하여 모델을 생성하였다. corrected.gz 파일의 인스턴스 중 기존 침입(292,300개)과 새로운 침입(18,729개)을 테스트하였다. 일반적으로 훈련에 사용되지 않은 침입 라벨은 테스트 되지 않기 때문에 새로운 침입라벨을 normal.로 변경하여 테스트하였다. 새로운 침입 18,729개의 인스턴스 중 1,827개는 침입으로 분류하였다. 새로운 침입으로 분류한 1,827개의 인스턴스는 buffer_overflow. 3개, neptune. 392개, portsweep. 164개, ipsweep. 9개, back. 511개, imap. 1개, satan. 개, 645 개, nmap. 102 개로 분류되었다.

주제어 : 웨카, 순차적최소최적화, KDDcup99, 침입탐지, 서포터벡터머신, 새로운 침입유형

Abstract Much research has been done using the KDDCup99 data set to study intrusion detection using artificial intelligence. Previous studies have shown that the performance of the SMO (SVM) algorithm is superior. However, intrusion detection studies of new intrusion types not used in training are insufficient. In this paper, a model was created using the instances of weka's SMO and KDDCup99 training data set, kddcup.data.gz. We tested existing instances(292,300) of the corrected.gz file and new intrusions(18,729). In general, intrusion labels not used in training are not tested, so new intrusion labels were changed to normal. Of the 18,729 new intrusions, 1,827 were classified as intrusions. 1,827 instances classified as new intrusions are buffer_overflow. Three, neptune. 392, portsweep. 164, ipsweep. 9, back. 511, imap. 1, satan. Dogs, 645, nmap. 102.

Key Words : Weka, SMO, KDDcup99, Intrusion Detection, SVM, New Intrusion Type

1. 서론

기계학습 알고리즘의 연구 및 개발 그리고 적용으로 인해 새로운 분류 방법 및 분류기가 개발되고 있다. 분류 기법으로는 통계적 방법, 퍼셉트론 기반 방법, 퍼지 기반 방법, 신경망 기반 방법 등 많은 분류 방법이 개발되었다

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government(MISP)(2016R1A4A1011761).
*Corresponding Author : Chang-Ho Seo(chseo@kongju.ac.kr)
Received February 23, 2018 Revised March 23, 2018
Accepted May 20, 2018 Published May 28, 2018

[1]. 현재 우리 사회는 4차산업혁명으로 변화가 이루어지고 있다. 이는 사물 인터넷, 빅데이터, 모바일 등 고도화된 지능정보기술을 기반으로 하는 인공지능화 시대로 접어들고 있다. 인공지능화 시대를 맞이하여 기존 침입에 대한 탐지 및 방지 기술도 중요하지만 새로운 침입에 대한 탐지 및 대응 능력이 더욱더 중요하다.

기계학습의 체계적 연구는 이전의 기계학습 연구를 보완하고 기계학습 긍정적인 측면을 부각한 연구가 있었고[2], 그 후 예제를 통한 학습은 인스턴스 기반 및 사례기반 추론 같은 방법으로 일반적인 지식을 습득하거나 구체적인 문제를 해결하는 방법으로 인공지능을 사용하였다[3]. 인공지능은 공학 및 의학 사회과학, 자연과학 등 학문의 모든 분야에서 다양하게 사용한다.

인공지능을 이용한 침입탐지 연구의 실험 데이터는 KDDCup99, NSL-KDD, Snort 등을 사용하여 사이킷런(Scikit-Learn), 웨카(Weak), 텐서플로(Tensorflow), 파이썬(Python), 자바(Java) 등을 사용하여 다양한 연구가 이루어지고 있다. 다양한 연구들에서 KDDCup99 데이터 세트와 웨카를 이용한 연구가 많이 진행되고 우수한 탐지 성능을 나타내고 있다[4-7].

기존 침입탐지 연구는 KDDCup99 데이터 세트의 침입을 4개의 카테고리(DoS, U2R, R2L, Probe)로 탐지결과를 나타내거나 알고리즘별로 성능을 비교하였다[8-11]. 또한, 전체 데이터 세트를 사용하여 훈련하는 것이 아니고 10% 데이터 세트로 훈련하여 침입을 탐지하였다.

본 논문에서는 10% 데이터 세트가 아닌 전체 데이터로 침입을 훈련하였고, 침입라벨별로 침입을 탐지하였다. 새로운 침입은 라벨을 normal.으로 변경하여 학습 모델이 새로운 침입에 대한 대응 여부를 분석하였다.

2장에서는 본 논문에서 사용한 웨카의 입력 옵션과 출력 및 결과 해석방법, SMO(Sequential Minimal Optimization) 알고리즘에 대해 설명하고, 3장에서는 KDDCup99 데이터 세트의 산술적인 통계분석을 하였다. 4장에서는 실험방법에 대하여 설명하고 5장에서는 훈련한 모델을 통해 기존의 침입과 새로운 침입의 탐지결과를 설명한다.

2. Weka 및 SVM

2.1 Weka

웨카((Waikato Environment for Knowledge Analysis)는 뉴질랜드 Waikato Hamilton 대학에서 만들어 배포하는 데이터마이닝 공개 프로그램이다.

웨카에서 서포터벡터머신(SVM) 분류자를 훈련하기 위해 John Platt의 순차적 최소 최적화 알고리즘으로 구현하였다. 이 구현은 누락 된 값을 전역적으로 대체하고 명목 속성을 2진 값으로 변환한다. 또한, 기본적으로 모든 속성을 정규화(표준화)한다. 다중 클래스 문제는 쌍으로 분류(1:1)를 사용하여 해결한다[12-14,18]. 적절한 확률을 얻으려면 모델에 맞는 옵션을 사용하여야 한다. 본 논문에서의 학습 옵션은 기본값으로 사용하였다.

웨카를 사용하여 학습하고 테스트 데이터를 통한 학습 결과는 다음과 같은 데이터가 표시된다[15,16].

- Kappa Statistic: 분류자가 분류한 공격과 실제 공격 사이의 일치도(agreement)이다. K=1이면 완벽하게 분류한 것이고, K=0이면 우연히 일치한 것이고, K값 높을수록 분류가 잘 되었다고 할 수 있다.

Table 1. Kappa Statistic

Kappa Value	Result
0.2 ≤ k	Slight(Poor)
0.2 < k ≤ 0.4	Fair
0.4 < k ≤ 0.6	Moderate
0.6 < k ≤ 0.8	Substantial
0.8 < k	Almost Perfect

- Mean Absolute Error(MAE): 에러들의 절대값의 평균으로 예측이 실제와 얼마나 가까운가를 계산한다.
- Root Mean Squared Error(RMSE): 에러값(관측값과 모델값의 차이)들의 제곱을 한 값을 평균으로 다시 제곱근으로 구한값이다. 따라서 RMSE & MAE 값이 최소일 때 예측과 정확도가 우수하다.

Table 2. Confusion Matrix

		True Condition	
		Intrusion	Normal
Predicted Condition	Intrusion	TP	FP
	Normal	FN	TN

- TP(True Positive): 인스턴스가 침입이고 침입으로 분류한 경우
- FN(False Negative): 인스턴스가 침입이고 정상으로 분류한 경우

- FP(False Positive): 인스턴스가 정상이고 침입으로 분류한 경우
- TN(True Negative): 인스턴스가 정상이고 정상으로 분류한 경우
- Recall: 실제 침입을 침입으로 판단

$$Recall = \frac{TP}{TP + FN}$$

- Precision: 예측한 침입에서 실제 침입으로 판단

$$\mathrm{P}recision = \frac{TP}{TP + FP}$$

- FPR(False Positive Rate): 실제 침입에서 정상으로 판단

$$FPR = \frac{FP}{FP + TN}$$

- F-Measure: Reacll-Precision 성능을 하나의 숫자포 표현하는 방법으로 Reacll-Precision의 조화평균

$$F-Measure = 2\frac{Precision \times Recall}{Precision + Recall}$$

- Accuracy: 침입과 정상을 올바르게 판단

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

3. KDDCUP99 데이터 세트

데이터 세트는 KDD-99 제5차 국제회의와 함께 개최된 제3회 국제 지식발견 및 데이터 마이닝 도구 공모전에 사용된 데이터이다. 공모전 과제는 네트워크 침입탐지기를 구축하기였다. 이 데이터는 군대 네트워크 환경에서 시뮬레이션 된 다양한 침입과 정상데이터를 포함하고 있다[17].

데이터는 침입유형이 라벨링 있는 데이터와 라벨이 없는 데이터로 구분되어 있다. 라벨이 있는 데이터는 감독학습 연구에 다양하게 사용되어왔다.

라벨이 있는 데이터 세트는 kddcup.data.gz, kddcup.data_10_percnet.gz, corrected.gz이 있다. kddcup.data.gz 은 전체 데이터 세트이고 kddcup.data_10_percnet.gz은 kddcup.data.gz의 10% 데이터 세트이다. corrected.gz은 기존의 침입유형과 새로운 침입유형을 포함한 데이터 세트로 훈련된 결과를 테스트하기 위한 데이터이다.

Table 3. Intrusion Categories and Intrusion label statistics for kddcup.data.gz file

label	count	rate	note
back.	2,203	0.045	
land.	21	0.000	
neptune.	1,072,017	21.885	
pod.	264	0.005	DoS
smurf.	2,807,886	57.322	
teardrop.	979	0.020	
Subtotal	3,883,370	79.278	
normal.	972,781	19.859	NOR
ipsweep.	12,481	0.255	
nmap.	2,316	0.047	
portsweep.	10,413	0.213	Probe
satan.	15,892	0.324	
Subtotal	41,102	0.839	
ftp_write.	8	0.000	
guess_passwd.	53	0.001	
imap.	12	0.000	
multihop.	7	0.000	
phf.	4	0.000	R2L
spy.	2	0.000	
warezclient.	1,020	0.021	
warezmaster.	20	0.000	
Subtotal	1,126	0.023	
buffer_overflow.	30	0.001	
loadmodule.	9	0.000	
perl.	3	0.000	U2R
rootkit.	10	0.000	
Subtotal	52	0.001	
Total	4,898,431	100.0	

Table 3은 kddcup.data.gz 파일의 침입 카테고리 및 침입라벨에 따른 데이터 인스턴스의 개수 및 비율이다. 인스턴스의 수가 총 4,898,431개이고 그 중 DoS가 3,88,370개로 79,3%이다. normal.의 인스턴스의 수가 972,781개로 19.8%로 구성되어 있으므로 대부분의 인스턴스가 normal.과 DoS 데이터로 구성되어 있다. Probe, R2L, U2L의 인스턴스는 42,280개로 0.863%로 다른 카테고리의 인스턴스와는 구성 비율에 많은 차이가 있다.

Table 4. Intrusion Categories and Intrusion label statistics for kddcup.data_10_percnet.gz file

label	count	rate	note
back.	2,203	0.446	DoS
land.	21	0.004	
neptune.	107,201	21.700	
pod.	264	0.053	
smurf.	280,790	56.838	
teardrop.	979	0.198	
Subtotal	391,458	79.239	
normal.	97,278	19.691	NOR
ipsweep.	1,247	0.252	Probe
nmap.	231	0.047	
portsweep.	1,040	0.211	
satan.	1,589	0.322	
Subtotal	4,107	0.831	
ftp_write.	8	0.002	R2L
guess_passwd.	53	0.011	
imap.	12	0.002	
multihop.	7	0.001	
phf.	4	0.001	
spy.	2	0.000	
warezclient.	1,020	0.206	
warezmaster.	20	0.004	
Subtotal	1,126	0.228	
buffer_overflow.	30	0.006	U2R
loadmodule.	9	0.002	
perl.	3	0.001	
rootkit.	10	0.002	
Subtotal	52	0.011	
Total	494,021	100.0	

Table 4은 kddcup.data_10_percnet.gz 파일의 침입 카테고리 및 침입라벨에 따른 데이터 인스턴스의 개수 및 비율이다. 총 494,021개의 데이터 중 DoS가 391,458개로 79.2%이고 normal. 데이터가 97,278개로 19.7%로 구성되어 있으므로 대부분의 데이터 인스턴스가 normal.과 DoS로 구성되어 있다. Probe, R2L, U2L 데이터는 5,285개로 1.1%로 적은 비율은 구성되어 있다. 하지만 kddcup.data.gz의 데이터 인스턴스 중 DoS, normal., Probe는 10%가량 줄였지만 R2L, U2L 데이터는 그대로 포함되어 있다.

Table 5. Intrusion Categories and Intrusion label statistics for corrected.gz file

label	count	rate	note
back.	1,098	0.353	DoS
land.	9	0.003	
neptune.	58,001	18.648	
pod.	87	0.028	
smurf.	164,091	52.757	
teardrop.	12	0.004	
Subtotal	223,298	71.793	
normal.	60,593	19.481	NOR
ipsweep.	306	0.098	Probe
nmap.	84	0.027	
portsweep.	354	0.114	
satan.	1,633	0.525	
Subtotal	2,377	0.764	
ftp_write.	3	0.001	R2L
guess_passwd.	4,367	1.404	
imap.	1	0.000	
multihop.	18	0.006	
phf.	2	0.001	
warezmaster.	1,602	0.515	
Subtotal	5,993	1.927	
buffer_overflow.	22	0.007	U2R
loadmodule.	2	0.001	
perl.	2	0.001	
rootkit.	13	0.004	
Subtotal	39	0.013	
apache2.	794	0.255	NEW Attack Label
httptunnel.	158	0.051	
mailbomb.	5,000	1.608	
mscan.	1,053	0.339	
named.	17	0.005	
processtable.	759	0.244	
ps.	16	0.005	
saint.	736	0.237	
sendmail.	17	0.005	
snmpgetattack.	7,741	2.489	
snmpguess.	2,406	0.774	
sqlattack.	2	0.001	
udpstorm.	2	0.001	
worm.	2	0.001	
xlock.	9	0.003	
xsnoop.	4	0.001	
xterm.	13	0.004	
Subtotal	18,729	6.022	
	311,029	100.000	Total

Table 5는 corrected.gz 파일의 침입 카테고리 및 침입라벨에 따른 데이터 인스턴스의 개수 및 비율이다. 총 311,029개의 데이터 중 DoS가 223,298개로 71.8%이고 normal. 데이터가 60,593개로 19.5%로 구성되어 있다. 대부분의 데이터 인스턴스가 normal.과 DoS로 구성되어 있다. Probe, R2L, U2L 데이터는 2,803개로 2.7%로 매우 적은 부분으로 구성되어 있다. 하지만 corrected.gz 파일

에는 기존의 데이터에는 포함되지 않은 새로운 침입라벨 apache2., httptunnel. 등 17개의 새로운 침입 인스턴스(18,729개, 6.0%)를 포함하고 있다.

4. 실험방법

웨카(Version 3.8.2)로 실험을 진행하였다. 전처리 단계로 KDDCUP99 데이터에는 칼럼명 없이 실제 데이터만 존재하기 때문에 데이터에 42개의 칼럼명을 추가하였다. kddcup.data.gz 데이터 인스턴스의 용량이 크기 때문에 MySQL을 사용하여 칼럼명을 추가하고 웨카의 입력으로 사용하기 위해 확장자가 csv 파일로 저장하였다.

csv 파일을 웨카 입력으로 사용하면 숫자는 Numeric으로 인식하기 때문에 Nominal로 변경해야 한다. 그래서 Preprocess탭에서 Filter의 unsupervised.attribute를 사용하여 7, 12, 21, 22번째 칼럼 land, logged_in, is_host_login, is_guest_login을 Numeric에서 Nominal으로 변환하였다. Classify탭에서는 학습할 알고리즘과 옵션을 선택하게 되어 있다. 알고리즘은 smo를 선택하고 옵션(Cross-Validation 10 Folds)은 기본으로 사용하였다. 학습은 kddcup.data.gz 데이터 인스턴스를 사용하여 훈련하였고, 테스트는 새로운 침입이 포함된 corrected.gz 데이터를 사용하였다. 웨카를 사용하여 corrected.gz 데이터 인스턴스를 테스트할 경우 훈련에 존재하는 침입라벨은 탐지가 되지만, 새로운 침입라벨은 새로운 침입이므로 훈련할 사용된 라벨로는 탐지할 수 없으므로 탐지가 되지 않았다. 새로운 침입라벨은 침입이 아닌 정상데이터 인스턴스로 가정하여 라벨을 normal.로 변경한 후 테스트를 하였다.

5. 실험결과

Fig. 1은 corrected.gz 파일 인스턴스의 테스트 결과 요약이다. 311,209개의 인스턴스 중 284,727건은 올바르게 탐지하였고, 7,573건은 올바르지 않게 탐지하였다. 또한, 18,729건은 새로운 침입이므로 무시되었다.

Fig. 2는 Fig. 1에서 무시된 침입 인스턴스의 라벨을 normal.으로 변경한 다음 테스트 결과이다. 18,729건의 새로운 침입이 탐지되었다. 침입으로 탐지된 인스턴스가 16,902개이고, 다르게 분류된 것은 1,827개이다. 이것은 침입라벨을 normal.으로 변경하였기 때문에 침입으로 탐지된 것이 1,827개이고 normal.으로 탐지된 것이 16,902개이다. 새로운 침입의 탐지율은 9.7549%이다.

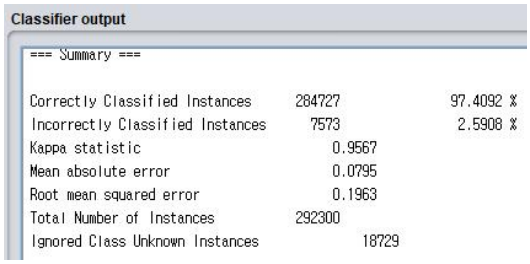

Fig. 1. corrected.gz Supplied Test Summary

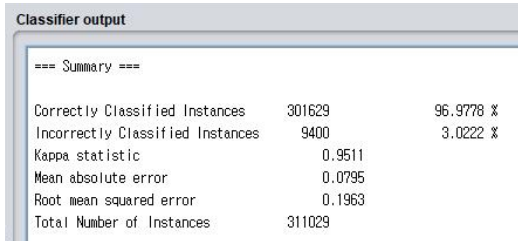

Fig. 2. corrected.gz Supplied Test Summary (new intrusion->normal.)

Table 6. Normal. and Intrusion Matrix

		Actual Class		Total
		Intrusion	Normal	
Predicated Class	Intrusion	225,123	945	226,068
	Normal	6,584	59,648	66,232
Total		231,707	60,593	292,300

Table 7. Normal. and Intrusion Matrix (new intrusion -> normal.)

		Actual Class		Total
		Intrusion	Normal	
Predicated Class	Intrusion	225,123	2,772	227,895
	Normal	6,584	76,550	83,134
Total		231,707	79,322	311,029

Fig. 3은 Fig. 1의 Confusion Matrix이다. a는 normal.이고 normal.을 normal로 탐지한 인스턴스는 59,648건이고, normal.을 침입으로 탐지한 인스턴스는 945개이다. Fig. 3.을 침입과 정상데이터로 정리한 것이 Table. 6. 이다.

Fig. 4(new intrusion->normal.)을 정리한 것이 Table 7이다. normal. 인스턴스가 60,539개에서 79,322개로 18,729개 증가한 것을 알 수 있다. 이는 새로운 침입 인스턴스를 noraml.로 변경하였기 때문이다. 침입탐지 후 결과 작성 시에는 새로운 침입은 normal.이 아니고 침입 인스턴스이므로 Table 8과 같이 수정하였다. Table 8의 침입 인스턴스가 Table. 6.보다 18,729개 증가한 250,436개임을 알 수 있다.

Table 8. Normal. and Intrusion Matrix (normal -> new intrusion.)

		Actual Class		Total
		Intrusion	Normal	
Predicated Class	Intrusion	226,950	945	250,436
	Normal	23,486	59,648	60,593
Total		227,895	83,134	311,029

Table 9는 Table 6과 Table 8의 결과로 침입탐지 시스템의 성능을 평가하기 위한 결과이다. 실제 침입 중 침입

Classifier output

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- classified as
59648	1	0	0	2	2	0	15	37	71	2	0	0	4	0	806	0	4	0	0	1	0	0	a = normal.
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = buffer_overflow.
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = loadmodule.
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = perl.
10	0	0	0	57977	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	e = neptune.
0	0	0	0	0	164091	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = smurf.
4367	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g = guess_passwd.
6	0	0	0	0	0	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h = pod.
2	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	i = teardrop.
2	0	0	0	5	0	0	0	0	343	1	0	0	0	0	3	0	0	0	0	0	0	0	j = portsweep.
2	0	0	0	0	6	0	0	0	0	297	0	0	0	0	0	0	1	0	0	0	0	0	k = ipsweep.
0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	l = land.
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = ftp_write.
433	0	0	0	0	0	0	0	0	0	0	0	0	665	0	0	0	0	0	0	0	0	0	n = back.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = imap.
98	0	0	0	5	0	0	0	0	7	0	0	0	0	0	1523	0	0	0	0	0	0	0	p = satan.
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	q = phf.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	r = nmap.
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	s = multihop.
1600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	t = warezmaster.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u = warezclient.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	v = spy.
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	w = rootkit.

Fig. 3. corrected.gz Supplied Test Confusion Matrix

Classifier output

=== Confusion Matrix ===

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	<-- classified as
76550	4	0	0	394	2	0	15	37	235	11	0	0	515	1	1451	0	106	0	0	1	0	0	a = normal.
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b = buffer_overflow.
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	c = loadmodule.
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	d = perl.
10	0	0	0	57977	0	0	0	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	e = neptune.
0	0	0	0	0	164091	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	f = smurf.
4367	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	g = guess_passwd.
6	0	0	0	0	0	0	81	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	h = pod.
2	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	i = teardrop.
2	0	0	0	5	0	0	0	0	343	1	0	0	0	0	3	0	0	0	0	0	0	0	j = portsweep.
2	0	0	0	0	6	0	0	0	0	297	0	0	0	0	0	0	1	0	0	0	0	0	k = ipsweep.
0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	l = land.
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	m = ftp_write.
433	0	0	0	0	0	0	0	0	0	0	0	0	665	0	0	0	0	0	0	0	0	0	n = back.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	o = imap.
98	0	0	0	5	0	0	0	0	7	0	0	0	0	0	1523	0	0	0	0	0	0	0	p = satan.
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	q = phf.
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	83	0	0	0	0	0	r = nmap.
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	s = multihop.
1600	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	t = warezmaster.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	u = warezclient.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	v = spy.
13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	w = rootkit.

Fig. 4. corrected.gz Supplied Test Confusion Matrix(new intrusion -> normal.)

을 얼마나 탐지하였는가를 평가하기 위한 지표 Recall은 기존 침입만 탐지하였을 경우 0.9716으로 나타났지만 새로운 침입을 포함하여 탐지하였을 경우 0.9062로 나타났다. 침입과 정상을 올바르게 탐지한 Accuracy는 0.9742에서 0.9215로 낮아졌다.

Table 9. Recall, Precision, FPR, Accuracy, F-Meansure of Classifiers

	corrected.gz	corrected.gz (new instrusion ->normal)
Recall	0.9716	0.9062
Precision	0.9958	0.9959
FPR	0.0156	0.0156
F-Measure	0.9836	0.9489
Accuracy	0.9742	0.9215

Table 10은 새로운 침입이 탐지된 라벨과 인스턴스의 개수이다. buffer_overflow. 3개, neptune. 392개, portsweep. 164개, ipsweep. 9개, back. 511개, imap. 1개, satan. 645개, nmap. 102개로 탐지되었다.

Table 10. Count of New Intrusion Detection

label	count
buffer_overflow.	3
neptune.	392
portsweep.	164
ipsweep.	9
back.	511
imap.	1
satan.	645
nmap.	102
Total	1,827

6. 결론

침입탐지에 관한 연구는 KDDCUP'99 데이터 세트와 웨카, 텐서플로, 사이킷런 등 다양한 방법으로 진행되었다. 이전 연구는 침입 카테고리별로 탐지결과를 제시하였고 corrected.gz 파일의 새로운 침입에 대한 탐지결과에 대한 논문은 미비하였다. 본 논문에서는 corrected.gz 파일의 침입 카테고리가 아닌 침입라벨로 결과를 제시하였고, 새로운 침입에 대한 탐지결과를 제시하였다. 새로운 침입의 라벨은 훈련 시 존재하지 않는 침입라벨이므로 탐지할 수 없었다. 새로운 침입이기 때문에 라벨을 정상 인스턴스인 normal.로 변경하여 새로운 침입에 대한 탐지를 시도하였다.

corrected.gz 파일의 기존 침입에 대한 탐지결과 Accuracy는 0.9742로 나타났지만, 새로운 침입을 포함한 corrected.gz 파일 전체 인스턴스에 대한 탐지결과는 0.9062로 나타났다. 새로운 침입 18,729개의 인스턴스 중 1,827개의 인스턴스는 침입으로 탐지하였고 16,902개의 인스턴스는 normal.로 탐지하였다.

REFERENCES

[1] Yugal kumar & G. Sahoo, (2012). Analysis of Parametric & Non Parametric Classifiers for Classification Technique using WEKA, *IJITCS, 4(7),* 43-49. DOI: 10.5815/ijitcs.2012.07.06

[2] DUTTON, D. & CONROY, G. (1997). A review of machine learning. *The Knowledge Engineering Review,12(4),* 341-367. DOI: 10.1017/S026988899700101X

[3] De Mantaras & Armengol E. (1998). Machine learning from example: Inductive and Lazy methods, *Data & Knowledge Engineering, 25,* 99-123. DOd: 10.1016/S0169-023X(97)00053-0

[4] Jing, L. & Bin, W. (2016, December). *Network Intrusion Detection Method Based on Relevance Deep Learning.* In Intelligent Transportation, Big Data & Smart City (ICITBS), 2016 International Conference on (pp. 237-240). IEEE. DOI: 10.1109/icitbs.2016.132

[5] Rani, N. & Purwar, R. K. (2017). Performance Analysis of various classifiers using Benchmark Datasets in Weka tools. *International Journal of Engineering Trends and Technology (IJETT), 47(5),* May. DOI: 10.14445/22315381/IJETT-V47P247

[6] Garg, T. & Khurana, S. S. (2014, May). *Comparison of classification techniques for intrusion detection dataset using WEKA.* In Recent Advances and Innovations in Engineering (ICRAIE), pp. 1-5. IEEE. DOI: 10.1109/ICRAIE.2014.6909184

[7] Ouyang, Z., Zhou, M., Wang, T. & Wu, Q. (2009, November). *Mining concept-drifting and noisy data streams using ensemble classifiers. In Artificial Intelligence and Computational Intelligence.* AICI'09. International Conference on (Vol. 4, pp. 360-364). IEEE. DOI: 10.1109/AICI.2009.153

[8] Ertam, F., & Yaman, O. (2017, September). *Intrusion detection in computer networks via machine learning algorithms.* In Artificial Intelligence and Data Processing Symposium (IDAP), 2017 International (pp. 1-4). IEEE. DOI: 10.1109/IDAP.2017.8090165

[9] Kabir, M. R., Onik, A. R., & Samad, T. (2017). A Network Intrusion Detection Framework based on Bayesian Network using Wrapper Approach. *International Journal of Computer Applications, 166(4).* DOI: 10.5120/ijca2017913992

[10] Garg, T., & Khurana, S. S. (2014, May). *Comparison of classification techniques for intrusion detection dataset using WEKA.* In Recent Advances and Innovations in Engineering (ICRAIE), 2014 (pp. 1-5). IEEE. DOI: 10.1109/ICRAIE.2014.6909184

[11] Modi, M. U., & Jain, A. (2015). *A survey of IDS classification using KDD CUP 99 dataset in WEKA. Int. J. Sci. Eng. Res, 6(11),* 947-954.

[12] Zeng, Z. Q., Yu, H. B., Xu, H. R., Xie, Y. Q., & Gao, J. (2008, November). *Fast training support vector machines using parallel sequential minimal optimization.* In Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on (Vol. 1, pp. 997-1001). IEEE DOI: 10.1109/iske.2008.4731075

[13] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy (2001). *Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation, 13(3),* 637-649. DOI: 10.1162/089976601300014493

[14] Trevor Hastie, Robert Tibshirani. (1998). *Classification by Pairwise Coupling.* In: Advances in Neural Information Processing Systems. DOI: 10.1214/aos/1028144844

[15] Srivastava, S. (2014). Weka: a tool for data preprocessing, classification, ensemble, clustering and association rule mining. *International Journal of Computer Applications, 88(10).* DOI: 10.5120/15389-3809

[16] E. M. Yang, H. J. Lee & C. H. Seo. (2017). Comparison of Detection Performance of Intrusion Detection System Using Fuzzy and Artificial Neural Network. *Journal of Digital Convergence, 15(6),* 391-398. DOI: 10.14400/JDC.2017.15.6.391

[17] http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[18] https://www.cs.waikato.ac.nz/~ml/weka/

양 은 목(Yang, Eun Mok) [정회원]

- 2000년 2월 : 한밭대학교 전자계산학과(학사)
- 2002년 2월 : 공주대학교 전자계산학과(석사)
- 2016년 8월 : 공주대학교 수학과(박사)
- 관심분야 : 정보보안, 통계, AI, 빅데이터, 등
- E-Mail : emyang@kongju.ac.kr

서 창 호(Seo, Chang Ho) [정회원]

- 1990년 2월 : 고려대학교 수학과(학사)
- 1996년 8월 : 고려대학교 수학과(박사)
- 1996년 8월 ~ 2000년 2월 : ETRI 선임연구원, 팀장
- 2000년 3월~ 현재 : 공주대학교 응용수학과 교수
- 관심분야 : 암호알고리즘, PKI, 무선 인터넷 보안 등
- E-Mail : chseo@kongju.ac.kr