KIPS Trans. Softw. and Data Eng.
Vol.10, No.11 pp.449~456
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2021.10.11.449

Comparative Study of Anomaly Detection Accuracy of Intrusion Detection Systems Based on Various Data Preprocessing Techniques

Kyungseon Park† · Kangseok Kim††

ABSTRACT

An intrusion detection system is a technology that detects abnormal behaviors that violate security, and detects abnormal operations and prevents system attacks. Existing intrusion detection systems have been designed using statistical analysis or anomaly detection techniques for traffic patterns, but modern systems generate a variety of traffic different from existing systems due to rapidly growing technologies, so the existing methods have limitations. In order to overcome this limitation, study on intrusion detection methods applying various machine learning techniques is being actively conducted. In this study, a comparative study was conducted on data preprocessing techniques that can improve the accuracy of anomaly detection using NGIDS-DS (Next Generation IDS Database) generated by simulation equipment for traffic in various network environments. Padding and sliding window were used as data preprocessing, and an oversampling technique with Adversarial Auto-Encoder (AAE) was applied to solve the problem of imbalance between the normal data rate and the abnormal data rate. In addition, the performance improvement of detection accuracy was confirmed by using Skip-gram among the Word2Vec techniques that can extract feature vectors of preprocessed sequence data. PCA-SVM and GRU were used as models for comparative experiments, and the experimental results showed better performance when sliding window, skip-gram, AAE, and GRU were applied.

Keywords : Intrusion Detection, Sliding Window, Skip-gram, AAE, GRU

다양한 데이터 전처리 기법 기반 침입탐지 시스템의 이상탐지 정확도 비교 연구

박 경 선† · 김 강 석††

요 약

침입 탐지 시스템(IDS: Intrusion Detection System)은 보안을 침해하는 이상 행위를 탐지하는 기술로서 비정상적인 조작을 탐지하고 시스템 공격을 방지한다. 기존의 침입탐지 시스템은 트래픽 패턴을 통계 기반으로 분석하여 설계하였다. 그러나 급속도로 성장하는 기술에 의해 현대의 시스템은 다양한 트래픽을 생성하기 때문에 기존의 방법은 한계점이 명확해졌다. 이런 한계점을 극복하기 위해 다양한 기계학습 기법을 적용한 침입탐지 방법의 연구가 활발히 진행되고 있다. 본 논문에서는 다양한 네트워크 환경의 트래픽을 시뮬레이션 장비에서 생성한 NGIDS-DS(Next Generation IDS Dataset)를 이용하여 이상(Anomaly) 탐지 정확도를 높일 수 있는 데이터 전처리 기법에 관한 비교 연구를 진행하였다. 데이터 전처리로 패딩(Padding)과 슬라이딩 윈도우(Sliding Window)를 사용하였고, 정상 데이터 비율과 이상 데이터 비율의 불균형 문제를 해결하기 위해 AAE(Adversarial Auto-Encoder)를 적용한 오버샘플링 기법 등을 적용하였다. 또한, 전처리된 시퀀스 데이터의 특징벡터를 추출할 수 있는 Word2Vec 기법 중 Skip-gram을 이용하여 탐지 정확도의 성능 향상을 확인하였다. 비교실험을 위한 모델로는 PCA-SVM과 GRU를 사용하였고, 실험 결과는 슬라이딩 윈도우, Skip-gram, AAE, GRU를 적용하였을 때, 더 좋은 성능을 보였다.

키워드 : 침입탐지, 슬라이딩 윈도우, Skip-gram, AAE, GRU

1. 서 론

침입 탐지 시스템(IDS: Intrusion Detection System)은 컴퓨터 자원의 기밀성, 무결성, 가용성 등을 침해하는 행위들을 탐지하는 기술로서 보안 공격 및 유출 경로의 다양화 및 고도화에 대응할 수 있는 기술이다[1]. 이러한 침입 탐지 시스템은 IT 기술의 급격한 성장으로 정보의 공유가 활발해짐과 동시에 정보의 노출이 심각해지고 있는 현 시점에서 호스트나 네트워크의 보안을 위한 기술 중 하나로 대두되고 있다. 기존의 침입탐지 시스템은 네트워크상의 패킷 트래픽 패턴 및 호스트로 접근하는 데이터 패턴을 분석하여 탐지하였다. 그러나 급속도로 성장하는 지능화된 기술에 의해 현대의 시

※ 이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1F1A1059036).
† 비 회 원 : 아주대학교 지식정보공학과 석사과정
†† 정 회 원 : 아주대학교 사이버보안학과 부교수
Manuscript Received : September 15, 2021
First Revision : October 13, 2021
Accepted : October 17, 2021
* Corresponding Author : Kangseok Kim(kangskim@ajou.ac.kr)

스템은 다양한 트래픽을 생성하기 때문에 기존의 탐지 방법은 한계점이 명확해졌다. 이러한 한계점을 극복하기 위해 다양한 기계학습 및 딥러닝 기법을 적용한 침입 탐지 방법의 연구가 활발히 진행되고 있다.

본 연구에서는 다양한 네트워크 환경의 정상 트래픽과 공격 트래픽을 시뮬레이션 장비로 생성한 공개 데이터셋인 NGIDS-DS(Next Generation IDS Dataset)[2]를 사용하여 실험을 진행하였으며, IDS의 탐지 정확도 향상을 위해 시퀀스 데이터의 처리 방법에 대한 비교 실험을 진행하였다. 데이터 전처리 기법으로는 패딩과 슬라이딩 윈도우 기법을 적용하였고, 데이터 불균형 문제를 해결하기 위해서 오버샘플링(Oversampling)과 언더샘플링(Undersampling) 기법을 적용하였다. 관련 기술로는 SMOTE(Synthetic Minority Oversampling TEchnique)[3]와 RUS(Random Under Sampling) 그리고 AAE(Adversarial Auto-Encoder)[4]를 사용하여 실험을 진행하였다.

실험에 사용한 분류 모델은 첫 번째 방법으로 차원 축소를 위한 PCA(Principal Component Analysis)와 SVM(Support Vector Machine)을 결합한 PCA-SVM[5] 모델을 사용하였다. 두 번째 방법은 자연어 처리에서 주로 사용되는 순환신경망(RNN: Recurrent Neural Network)모델 중 GRU(Gated Recurrent Unit)[6]를 적용한 딥러닝 모델을 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서는 침입탐지 방법에 기계학습을 적용한 기존 관련 연구를 기술하였고, 3장에서는 본 연구에서 사용한 데이터셋과 탐지 정확도 향상을 위해 적용한 기술들과 관련하여 실험방법을 기술하였으며, 4장에서는 실험결과와 분석을 기술하였고, 5장에서는 결론 및 향후 연구에 대해 제시하였다.

2. 관련 연구

고전적인 기계학습 분류 모델부터 딥러닝과 GAN에 이르기까지 기계학습 분야는 급속도로 성장하고 있다. 이에 따라 다양한 기법들이 제시되고 있으며, 침입 탐지 모델의 정확도 향상을 위해 많은 연구들이 모델을 설계하고 다양한 기법을 적용한 실험을 진행하고 있다.

정윤경 외[7]는 SVM(Support Vector Machine)과 결정트리 등 다양한 기계 학습 기법을 이용하여 침입 탐지 모델을 설계하였고, 불균형 데이터의 경우 어떤 모델에서 가장 좋은 성능을 보이는지 실험하였다. 성능 측정 지표는 정밀도(Precision)와 재현율(Recall)을 동시에 고려하는 F_β-Score를 사용하여 성능 평가를 하였다.

이민욱[8]은 비순환형 학습모델인 kNN, Naive Bayesian, Random Forest와 LSTM(Long Short-Term Memory)을 기반으로한 SM-LSTM(Session Management based LSTM) 모델을 제안하며, 비순환형 모델에 비해 순환형 모델의 탐지 성능이 더 높은 것을 확인하였다.

Md Hasan Shahriar 외[9]는 KDD-99 데이터셋을 이용한 GAN 기반 오버샘플링을 적용하고, 원본 샘플과 함께 학습된 IDS, G-IDS를 제안하였으며 공격 감지 및 모델 안정화에서 더 나은 성능을 보인다는 것을 입증하였다.

Roberto Corizzo 외[10]는 ADFA-LD, NGIDS-DS 그리고 WWW2019 데이터셋을 이용하여 다양한 임베딩(Embedding) 방법으로 성능을 비교하고 단어 임베딩 방법이 IDS의 성능을 높일 수 있다는 가능성을 보였다.

민병준 외[11]는 데이터 불균형 문제의 해결과 대용량 네트워크 트래픽을 실시간으로 탐지하면서 학습 성능을 보장할 수 있는 HFS(Hybrid Feature Selection) 기법을 제안하였고, DNN(Deep Neural Network)과 결합하여 침입 탐지를 위한 HFS-DNN을 제시하였다.

이주화 외[12]는 CICIDS 2017 데이터셋을 이용하여 실험을 진행하였으며, GAN을 기반으로 데이터 불균형을 해결한 데이터와 불균형 문제를 다루지 않은 데이터에 대해 각각의 랜덤 포레스트 모델을 비교하여 GAN을 사용한 모델의 성능이 더 우수하다는 것을 입증하였다.

본 연구에서는 슬라이딩 윈도우, 임베딩, 오버샘플링 기법 등을 적용하여 전처리된 데이터와 함께 기계학습/딥러닝 모델의 탐지 정확도에 어떤 영향을 미치는지 실험을 진행하였다.

3. 연구 방법

본 실험에서 적용한 데이터 처리의 흐름은 Fig. 1과 같다. 미가공 데이터인 NGIDS-DS를 정렬한 후 맵핑(Mapping)한다. 그 후, 패딩, 슬라이딩 윈도우, 오버샘플링 그리고 Skip-gram을 적용하여 데이터를 전처리하고 PCA-SVM과 GRU 모델을 사용하였다. 학습된 모델의 성능 평가는 AUC와 F_1-Score를 사용하였다.

슬라이딩 윈도우를 적용한 데이터에 대해서는 임베딩을 적용하고 분류모델을 학습하였다. 패딩한 데이터에 임베딩을 적용한 모델의 결과가 Table 1과 같이 정확도 향상에 큰 영향을 주지 않기 때문에 Fig. 1과 같이 실험을 진행하였다.

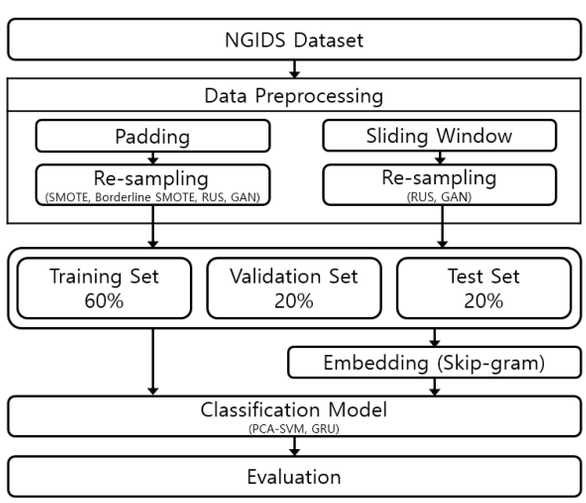

Fig. 1. Systematic Representation for the Proposed Anomaly Detection Methodology

Table 1. Skip-gram Applied Results

Embedding	AUC	Recall	Precision	F_1-Score
None	0.724	0.449	0.915	0.603
Skip-gram	0.748	0.496	0.994	0.662

3.1 데이터셋

실험에 사용된 NGIDS-DS[2]는 2017년에 공개된 데이터셋으로 ADFA(Australian Defence Force Academy)에서 진행되었던 프로젝트의 일부이며, 시뮬레이션 장비를 사용해서 생성한 리눅스 환경에서 수행되는 정상 및 비정상 호스트를 수집한 데이터셋이다. Fig. 2와 같이 일반적인 리눅스 서버에서 추출할 수 있는 미가공 데이터와 비슷한 형태를 갖고 있다.

NGIDS-DS의 호스트 로그는 총 90,054,160개이다. 속성은 date, time, pro_id, path, sys_call, event_id, attack_cat, attack_subcat, label이 있다. 이 중 트래픽의 로깅 시간을 알 수 있는 날짜(date)와 시간(time) 속성을 이용하여 데이터를 정렬하였다. 그 결과는 Fig. 3과 같으며, pro_id 속성으로 시퀀스를 정렬할 수 있다.

그리고 시스템콜(sys_call)과 시스템콜이 호출되는 응용프로그램의 위치를 나타내는 로컬 주소(path)를 입력 데이터로 사용하였다. data와 time 속성은 367,290개의 도메인 개수를 갖고 있어 367,290개의 샘플의 시퀀스 데이터로 처리하였다.

	A	B	C	D	E	F	G	H	I
1	11/03/2016	2:45:01	1830	/sbin/upstart-dbus-bridge	142	45354	normal	normal	0
2	11/03/2016	2:45:06	1804	/bin/dbus-daemon	256	45352	normal	normal	0
3	11/03/2016	2:45:06	2133	/usr/lib/i386-linux-gnu/gconf/gconfd-2	168	45372	normal	normal	0
4	11/03/2016	2:45:35	4528	/usr/bin/python3.4	3	39459	normal	normal	0
5	11/03/2016	2:45:44	1847	/usr/bin/ibus-daemon	102	37263	normal	normal	0
6	11/03/2016	2:45:44	1907	/usr/lib/ibus/ibus-ui-gtk3	168	37896	normal	normal	0
7	11/03/2016	2:45:44	1925	/usr/lib/ibus/ibus-engine-simple	168	37542	normal	normal	0
8	11/03/2016	2:45:44	4461	/usr/sbin/apache2	142	37647	normal	normal	0
9	11/03/2016	2:45:45	1081	/usr/bin/Xorg	102	37480	normal	normal	0
10	11/03/2016	2:45:11	3989	/sbin/auditd	256	45374	normal	normal	0
11	11/03/2016	2:45:06	2834	/usr/bin/update-notifier	142	45360	normal	normal	0
12	11/03/2016	2:45:01	1885	/usr/lib/unity/unity-panel-service	168	45353	normal	normal	0
13	11/03/2016	2:45:44	2363	/usr/bin/gnome-terminal	102	37266	normal	normal	0
14	11/03/2016	2:45:29	2106	/usr/lib/evolution/evolution-calendar-factory	168	45012	normal	normal	0
15	11/03/2016	2:45:01	1872	/usr/lib/unity/unity-panel-service	168	45355	normal	normal	0
16	11/03/2016	2:45:45	1081	/usr/bin/Xorg	102	37483	normal	normal	0
17	11/03/2016	2:45:37	1519	/usr/lib/rtkit/rtkit-daemon	168	37285	normal	normal	0
18	11/03/2016	2:45:29	2346	/usr/lib/telepathy/mission-control-5	168	45003	normal	normal	0

Fig. 2. Part of Host Log from NGIDS-DS

date_time	Features	attack cat	attack subcat	label
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:01	·····	0	0	0
2016-03-11 02:45:45	·····	1	1	1
2016-03-11 02:45:45	·····	1	1	1
2016-03-11 02:45:45	·····	1	1	1
2016-03-11 02:45:45	·····	1	1	1
2016-03-11 02:57:24	·····	2	13	1
2016-03-11 02:57:24	·····	2	13	1
2016-03-11 02:57:24	·····	2	13	1
2016-03-11 02:57:24	·····	2	13	1
2016-03-11 02:57:24	·····	2	13	1

Fig. 3. NGIDS-DS Sorted Chronologically

동일한 시간으로 분류된 각각의 시퀀스 데이터는 attack_cat, attack_subcat 그리고 label이 Fig. 3과 같이 동일한 값을 갖고 있다.

또한, path 속성은 100개, sys_call 속성은 122개의 도메인 개수를 갖고 두 속성의 쌍은 1,670개의 속성을 가지고 있다.

3.2 데이터 전처리

NGIDS 데이터에서 동일한 시간에 발생한 이벤트들의 속성(path, sys_call)들을 튜플 형태로 연결한 시퀀스 데이터 형태로 전처리하였다. 이를 패딩, 슬라이딩 윈도우를 사용해 시퀀스의 길이를 일치시켰으며, 정상적인 트래픽의 시퀀스에 비해 공격 트래픽의 시퀀스가 매우 적어 오버샘플링을 적용하여 데이터 클래스의 불균형을 완화시켰다. 또한, Skip-gram을 적용하여 시퀀스 의미를 표현한 특징 벡터를 추출하였다.

1) 패딩 기법

패딩은 가변적 길이를 갖는 시퀀스를 같은 길이로 맞춰 주기 위해 사용되는 데이터 전처리 기법이다. 기준 길이보다 긴 시퀀스는 기준 길이로 자르고, 기준 길이보다 짧은 시퀀스는 '0'을 채우는 제로패딩(Zero-padding)으로 길이를 일치시켰다. 이러한 패딩 기법에는 '0'을 앞에서부터 채우는 프리패딩(Pre-padding)과 뒤에서부터 채우는 포스트패딩(Post-padding)이 있다. 이는 뒷단의 입력이 중요한 RNN 모델에서는 프리패딩의 성능이 뛰어나기 때문에 프리패딩을 사용하여 패딩을 수행하였다[13].

패딩의 길이는 다음에 설명할 슬라이딩 윈도우 기법을 통해서 결정한 길이와 동일한 387로 실험하였다.

2) 슬라이딩 윈도우 기법

슬라이딩 윈도우를 사용하면 시계열 데이터를 효율적으로 분할할 수 있다. 슬라이딩 윈도우 기법을 적용한 데이터는 시계열 데이터의 함축적 맥락을 도출하거나 다양한 패턴을 도출할 수 있어 시계열 데이터를 분석하고 연구하는 과정에서 사용되는 기본 요소 중 하나로 사용되고 있다[14]. 도출된 데이터의 클래스 범주(Class Category)는 동일한 시간에 발생한 이벤트들의 라벨(Label) 속성을 사용하였다.

슬라이딩 윈도우에서 조절할 수 있는 파라미터는 Fig. 4와 같이 Window Size와 Shifting Size가 있다. 일반적으로 Shifting Size는 입력 데이터의 길이와 연관이 없고 작을수록 모델의 정확도가 향상되고 클수록 샘플의 수가 줄어들어 정확도가 저하될 수 있다. 이러한 이유로 Shifting Size는 8로 고정하고 Windows Size를 결정하기 위한 실험을 진행하였다.

Fig. 5와 같이 마지막 값 또는 시퀀스 중간에서 공격 시도의 흔적으로 공격 트래픽으로 결정될 수 있다. 임의로 Window Size를 결정하면 모집단 중 정상과 공격 트래픽에서 동일한 시퀀스를 나타낼 수 있기 때문에 Windows Size를 50부터 700까지 변경하면서 중복제거 이전과 이후의 비율을 확인하여 공격과 정상 트래픽에서 최대한 중복되지 않는 샘플을 추출하였다.

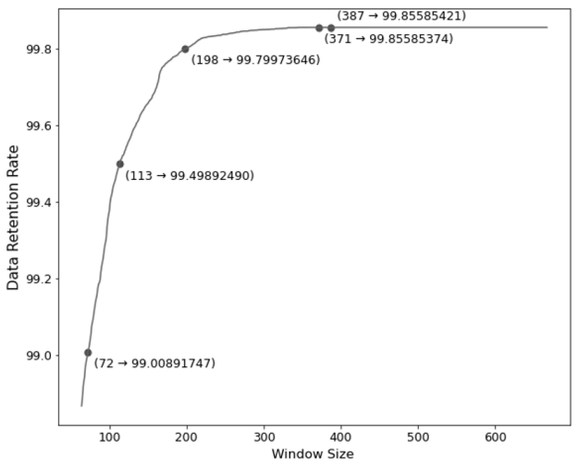

Fig. 4. Test for Determining Sliding Window Size

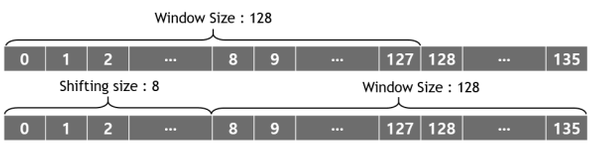

Fig. 5. Sliding Window

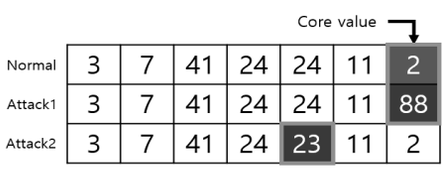

Fig. 6. Example of Determining Traffic Type

Table 2. Model Results by Window Size of Sliding Window

Window Size	AUC	Recall	Precision	F_1-Score
72	0.891	0.537	0.929	0.681
113	0.931	0.597	0.941	0.731
198	0.959	0.705	0.953	0.810
387	0.995	0.931	0.971	0.951

결과는 Fig. 6의 그래프와 같다. x축의 값은 Window Size이며 y축의 값은 슬라이딩 윈도우를 적용한 샘플의 개수에 대해 중복제거 이후의 샘플의 개수의 비율이다. 비율이 유의미하게 변화하는 값들을 Window Size로 결정하여 Sliding Window를 적용한 분류 모델 실험을 진행하였다.

실험 결과는 Table 2와 같다. Window Size는 387일 때 가장 좋은 정확도를 나타내고 있으며, 이후의 실험에 대해서도 Window Size를 387로 실험을 진행하였다.

3) 오버샘플링

시퀀스 데이터로 변환한 NGIDS-DS는 Fig. 7과 같이 클래스 불균형을 이루고 있다. 본 논문에서는 오버샘플링 기법으로 SMOTE, Borderline SMOTE 그리고 AAE(Adversarial Auto-Encoder)를 통해 비정상 트래픽 데이터를 생성한다. 그리고 언더 샘플링 기법으로는 RUS(Random under sampling)

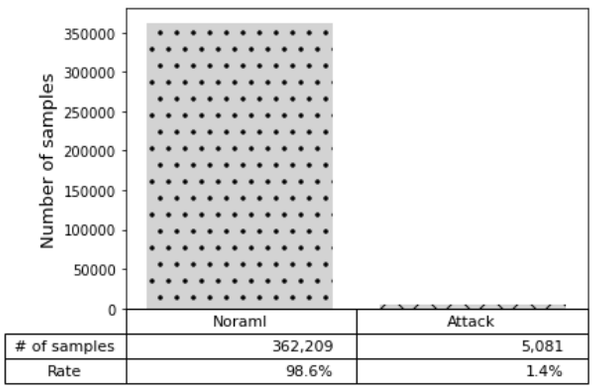

Fig. 7. Percentage of Data in the NGIDS-DS

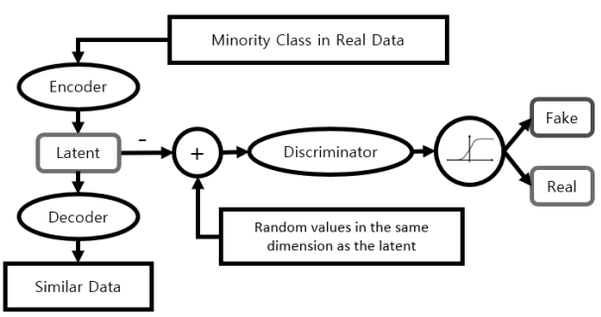

Fig. 8. Data Augmentation With AAE

를 사용하여 클래스 불균형 문제를 해소하였다.

본 실험에서 오버샘플링을 위해 사용한 AAE[4]는 Fig. 8과 같다. AAE는 변이형 오토인코더(VAE : Variational Auto-Encoder)와 GAN의 판별자(Discriminator)를 응용한 모델이다. 인코더(Encoder)와 디코더(Decoder)를 통해 생성된 잠재공간과 난수의 배열의 합을 판별자에 입력으로 주어 시그모이드(Sigmoid) 함수에 의해 진위를 판별한다. 그 이후, 학습이 완료된 AAE에 노이즈(Noise)를 입력하여 공격 데이터와 유사한 데이터를 생성한다.

4) Skip-gram

단어 임베딩 모델(Word2Vec)은 자연어처리에서 언어 모델링 및 기능 학습을 위해 일반적으로 채택되는 기술이다[10]. 단어와 문장을 자동화된 분석 도구에서 활용할 수 있는 특징 벡터로 맵핑하는 것이 일반적이다.

전처리된 시퀀스 데이터를 특징 벡터들(Feature Vector or Embedding Vector)로 변환하기 위해 Word2Vec 기법 중 Skip-gram을 사용하였다. Skip-gram은 같은 문장의 다른 단어를 기반으로 한 단어의 분류를 극대화한다[15]. 이를 통해 시퀀스를 구성하는 속성(path, sys_call) 튜플들의 상관관계를 반영한 특징 벡터를 추출할 수 있다.

3.3 학습 모델

본 논문에서는 PCA-SVM과 GRU 학습 모델을 사용하여 비교 실험하였다.

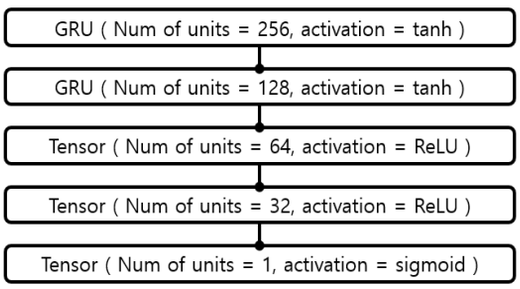

Fig. 9. Structure of GRU Classification Model

1) PCA-SVM

PCA는 다양한 분야의 패턴 인식에서 대표적으로 차원 축소를 위해 이용되는 다변량 통계 분석 방법이다. 주어진 데이터를 분산이 최대가 되는 축으로 변환하는 것이다. 이 새로운 차원에서의 데이터 벡터들을 주성분(Principal Components)이라고 한다. 분산이 작은 성분을 제거함으로써 데이터의 차원을 줄이는 동시에 데이터에 포함되어 있던 잡음(noise)를 제거할 수 있다[16]. SVM[17]은 Vapnik에 의해 개발된 기계학습 알고리즘으로 지도 학습(Supervised learning)에 이용된다. 커널기법을 이용하여 선형 또는 비선형으로 초평면이 만들어지는 고차원 특성으로 맵핑하게 된다[15]. PCA-SVM[16]은 PCA를 이용하여 데이터의 특징 벡터를 추출한 데이터에 SVM을 적용한 분류 기법이다.

2) GRU

2014년 GRU를 제시한 조경현 외[6]는 복잡했던 LSTM 구조를 단순화했다. GRU는 LSTM에 비해 단순화 된 만큼 학습 속도가 빠르고, LSTM과 비슷한 성능을 보인다고 알려져 있다. 실제 성능은 Gruber Nicole 외[19]의 실험을 통해 LSTM이 더 많은 데이터를 갖고 있을 때, 더 좋은 성능을 보인다고 하였다. 하지만 본 실험에서는 더 적은 파라미터를 통해 자원의 부담을 줄이기 위해 GRU를 사용한 모델을 Fig. 9와 같이 설계하였다.

GRU 층의 겹이 많아질수록 모델의 성능이 향상되는 것은 아니기 때문에 보편적으로 사용되는 2-layer GRU를 사용하였으며 이진 분류 모델을 위해 완전 연결 계층의 마지막 층의 활성함수는 sigmoid 함수를 사용하였다. 그리고 모델의 과대적합을 방지하기 위해 Dropout과 조기 종료(Early Stopping)을 적용해 실험을 진행하였다. Dropout의 비율은 0.5로 각 계층 사이에 배치하였다. Early Stopping은 학습 데이터의 Loss를 기준으로 7회 반복되면 학습을 멈추게 설정하였다. 이는 비교적 높은 수치이지만 목표로 설정한 Epoch에 최대한 도달하면서 불필요한 학습과 과대적합을 방지하기 위함이다.

4. 실 험

4.1 실험환경

본 실험에서 사용한 환경은 Table 3과 같다.

Table 3. Experimental Environment

OS	Windows 10
CPU	Intel i7-8750H
RAM	16G
GPU	GTX1060 6GB
Framework	Tensorflow 2.3

4.2 정확도 평가지표

본 실험의 평가지표로 AUC와 F_1-Score를 사용하였다. AUC는 이진 분류에 가장 적합한 지표이지만, 과도한 클래스 불균형을 갖고 있는 경우에는 정확한 지표가 될 수 없을 수도 있다. 그래서 재현율(Recall)과 정밀도(Precision)의 조화평균인 F_1-Score도 평가지표로 이용하여 모델 성능 평가의 객관성을 확보하고자 하였다.

4.3 패딩을 사용한 모델 실험

NGIDS-DS의 Host log를 정렬 및 맵핑을 거쳐 일정한 길이로 패딩한 데이터에 대해 불균형 문제를 해결하는 방법들을 적용하고 모델을 학습하였다.

Table 4는 3장에서 기술한 PCA-SVM을 적용한 모델의 결과이다. 첫 번째 모델은 데이터 불균형에 대해 리샘플링을 적용하지 않았으며, 두 번째 모델은 RUS와 Borderline SMOTE를 순차적으로 적용하여 데이터의 비율을 0.3으로 조정한 모델의 결과이다. 리샘플링의 비율은 정상 트래픽에 대한 공격 트래픽의 비율을 의미한다. 본 실험에서 사용된 리샘플링 비율의 표기는 모두 이 방법과 같다.

Table 4에 의하면 패딩을 사용한 데이터에 대한 PCA-SVM은 정상적인 트래픽과 공격 트래픽의 비율에 의해 정확도가 큰 영향을 받는 것을 확인할 수 있었다.

Table 5는 패딩을 사용한 샘플을 SMOTE, Borderline SMOTE, AAE, RUS를 사용하여 불균형 문제를 해결하고 GRU 모델을 적용한 결과이다. 리샘플링 방법과 정상 데이터와 공격 데이터의 리샘플링 비율 그리고 AAE 모델을 적용한 리샘플링이 모델 성능과 학습 시간에 어떤 영향을 끼치는지 비교를 위해 여러 모델의 성능을 측정하였다.

Fig. 10은 Table 5의 결과를 총 샘플수를 기준으로 정렬하고, 모델의 학습시간을 표현한 막대그래프와 정확도를 확인할 수 있는 AUC와 F_1-Score를 꺾은선 그래프로 표현한 그래프다.

RUS를 적용한 모델은 샘플의 총 개수가 줄어 모델의 학습 시간은 감소하였지만 정확도가 크게 저하되는 것을 확인할 수 있었다. 반면에 오버샘플링 기법을 적용한 모델은 정확도가 크게 상승되며, 특히 AAE 기반 오버샘플링을 적용한 모델

Table 4. PCA-SVM Results with Padding

Re-sampling(Rate)	AUC	Recall	Precsion	F_1-Score
None	0.623	0.252	0.910	0.395
RUS+Borderline(0.3)	0.813	0.718	0.758	0.737

Table 5. GRU Model Results with Padding

Re-sampling(Rate)	AUC	Recall	Precsion	F_1-Score
None	0.724	0.449	0.915	0.603
SMOTE(0.3)	0.948	0.929	0.896	0.912
SMOTE(0.5)	0.964	0.923	0.923	0.945
SMOTE(0.7)	0.969	0.979	0.943	0.961
SMOTE(0.9)	0.970	0.981	0.954	0.968
Borderline(0.3)	0.961	0.951	0.909	0.930
Borderline(0.5)	0.965	0.959	0.944	0.952
Borderline(0.7)	0.978	0.986	0.957	0.972
Borderline(0.9)	0.977	0.985	0.967	0.976
RUS(0.04)	0.578	0.158	0.752	0.261
RUS(0.06)	0.552	0.108	0.694	0.186
RUS(0.08)	0.605	0.215	0.800	0.339
RUS(0.1)	0.598	0.200	0.799	0.320
RUS(0.2)	0.604	0.218	0.810	0.344
RUS(0.3)	0.606	0.232	0.780	0.357
AAE(0.3)	0.987	0.974	0.998	0.986

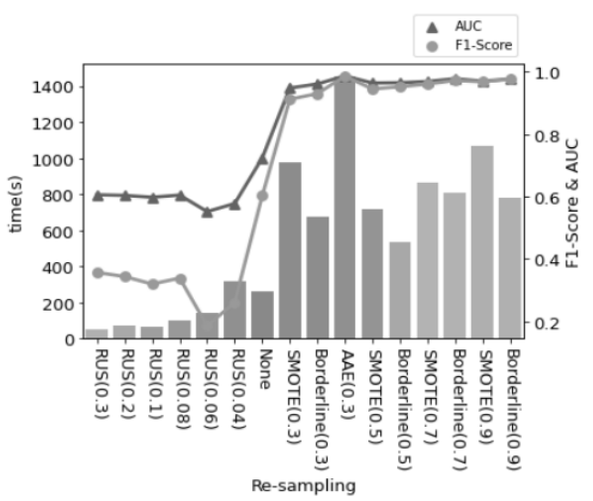

Fig. 10. GRU Classification Model Results with Padding

은 더 많은 학습시간이 소요되었지만, 다른 오버샘플링 기법보다 향상된 정확도를 얻을 수 있었다.

4.4 슬라이딩 윈도우를 사용한 모델 실험

슬라이딩 윈도우를 사용하게 되면, 패딩을 적용한 것과 달리 표본의 무의미한 데이터의 삽입을 줄일 수 있고 표본의 수도 증가시킬 수 있다. 슬라이딩 윈도우가 사용된 샘플에 대해 불균형 문제를 해결하고 skip-gram을 적용하여 모델을 학습하였다.

Table 6은 PCA-SVM로 분류한 모델의 결과이다. 첫 번째 모델은 데이터 불균형에 대해 리샘플링을 적용하지 않았으며, 두 번째 모델은 RUS와 Borderline SMOTE를 순차적으로 적용하여 데이터의 비율을 0.3으로 조정한 모델의 결과이다.

Table 6에 의하면 패딩을 사용한 샘플과 동일하게 PCA-SVM은 정상적인 트래픽과 공격 트래픽의 비율에 의해 정확도가 큰 영향을 받는 것을 확인할 수 있었다.

Table 6. PCA-SVM Results with Sliding Window

Re-sampling(Rate)	AUC	Recall	Precision	F_1-Score
None	0.678	0.359	0.778	0.491
RUS+Borderline(0.3)	0.833	0.700	0.861	0.772

Table 7. GRU Model Results with Sliding Window

Re-sampling(Rate)	AUC	Recall	Precision	F_1-Score
None	0.995	0.931	0.971	0.951
RUS(0.04)	0.995	0.934	0.953	0.943
RUS(0.06)	0.990	0.878	0.973	0.923
RUS(0.08)	0.995	0.910	0.975	0.942
RUS(0.1)	0.992	0.926	0.928	0.927
RUS(0.2)	0.989	0.915	0.948	0.931
RUS(0.3)	0.982	0.881	0.940	0.910
RUS+AAE(0.3)	0.999	0.989	0.997	0.993

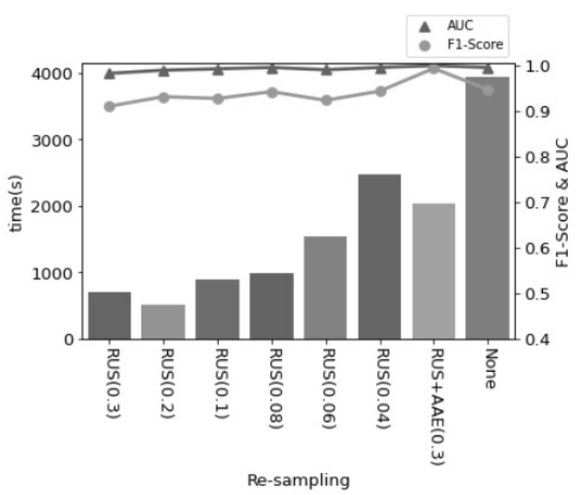

Fig. 11. GRU Classification Model Results with Sliding Window

Table 7은 슬라이딩 윈도우를 적용한 데이터에 대해 RUS와 AAE로 불균형 문제를 해결하고 GRU 모델로 분류한 결과이다.

Fig. 11은 Table 7의 결과를 총 샘플수를 기준으로 정렬하고, 모델의 학습시간을 표현한 막대그래프와 정확도를 확인할 수 있는 AUC와 F_1-Score를 꺾은선 그래프로 표현한 그래프다.

패딩을 사용한 경우와 달리, RUS를 적용한 경우에도 모델의 정확도가 크게 변동하지 않는 것을 확인할 수 있었다. 패딩과 skip-gram을 사용한 Table 1의 모델과 달리 슬라이딩 윈도우와 skip-gram을 사용한 샘플에는 무의미한 데이터가 줄어 정확도가 크게 향상되었다. AAE 기반 오버샘플링 기법과 RUS를 같이 적용한 모델의 경우 샘플의 수가 줄어 학습시간이 크게 감소한 반면 모델의 성능은 소폭 향상되었다.

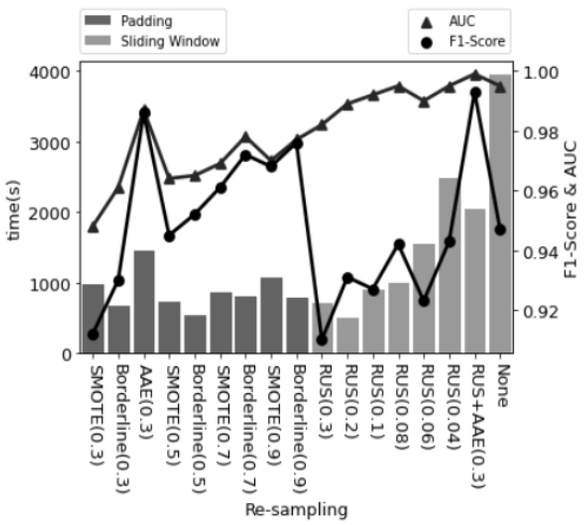

Fig. 12. GRU Classification Model Results

4.5 실험 결과 분석

Table 5와 Table 7의 결과에 의하면 PCA-SVM 모델의 결과가 클래스 불균형 데이터에 대해 좋지 않은 성능을 보였다. RUS와 Borderline SMOTE를 적용한 모델의 경우, 성능의 향상을 보였지만, GRU 모델에 비해 좋지 않은 성능을 보였다.

Fig. 12는 GRU 분류 모델 중 F_1-Score와 AUC가 0.9이상인 모델을 비교한 그래프다. RUS를 사용한 사례가 오버샘플링을 사용한 사례보다 학습시간이 더 적게 소요되지만 모델 성능이 저하되는 것을 확인할 수 있었다. 하지만 슬라이딩 윈도우와 임베딩 방법 중 하나인 skip-gram을 적용하면 성능 저하가 크게 발생하지 않았다.

또한 오버샘플링 방법으로 AAE를 적용하면 모델의 정확도를 향상시킬 수 있었으며, RUS와 동시에 적용하면 학습시간을 줄일 수 있다는 것을 확인하였다. 더하여 SMOTE, Borderline SMOTE를 적용한 모델보다 AAE를 적용한 모델이 더 향상된 정확도를 보였다.

Fig. 12를 통해 봤을 때, 패딩을 적용한 모델이 더 좋은 성능을 나타낸 것처럼 보이고 있다. 하지만 본 실험 환경의 한계로 인해 슬라이딩 윈도우를 사용한 샘플에 오버샘플링 기법을 적용하지 못한 결과이다. 패딩을 사용한 실험 결과로 보았을 때, 오버샘플링은 모델 정확도 향상에 효과적인 것을 확인할 수 있었으며 슬라이딩 윈도우를 사용한 실험에서 AAE를 적용한 오버샘플링을 통해 성능이 개선되는 것을 확인할 수 있었기 때문이다. 이런 점을 감안하더라도 본 실험에서는 슬라이딩 윈도우와 skip-gram 그리고 AAE 기반 오버샘플링을 적용한 모델의 정확도가 가장 개선된 성능을 보였다.

5. 결 론

본 논문에서는 한정된 자원을 이용하여 정확성 향상을 위한 모델의 효율적인 데이터 전처리 실험을 진행하였고, 이를 검증하기 위해 PCA-SVM 모델과 GRU 분류 모델의 비교를 진행하였다. 실제 미가공 데이터에서 접할 수 있는 데이터 클래스 불균형 문제에 대해 기존에 제시되었던 SMOTE, Borderline SMOTE, Random Under Sampling과 같은 리샘플링 알고리즘과 최근 머신러닝 분야에서 화두가 되고 있는 AAE을 사용한 오버샘플링을 적용해 정상 트래픽과 공격 트래픽의 표본의 비율이 균일하게 분포되도록 처리 후 학습한 모델들을 비교·평가하였다.

네트워크 트래픽에 대해 단순히 데이터를 자르는 패딩에 비해 슬라이딩 윈도우와 Skip-gram을 적용한 데이터를 학습한 분류 모델의 성능이 더 안정적이고 향상된 정확도를 보였다. 그리고 클래스 불균형 문제에 대해서는 SMOTE와 Borderline SMOTE 보다 AAE 기반 오버샘플링 기법을 적용하면 모델의 정확도를 향상시킬 수 있었다. 또한 RUS와 동시에 적용하면 학습 시간을 줄일 수 있다는 것을 실험을 통해 확인하였다.

향후 연구에는 도메인 적응(Domain Adaptation), 퓨샷 러닝(Few-shot Learning) 등과 같은 기계학습 분야에서 새롭게 화두가 되고 있는 기술들을 적용 및 실험을 통해 침입탐지 분야 기술의 정확성 향상을 위한 연구를 진행할 것이다.

References

[1] Y. Lee, "Design and analysis of multiple intrusion detection model," *Journal of The Korea Institute of Electronic Communication Sciences*, Vol.11, No.6, pp.619-626, 2016.

[2] W. Haider, J. Hua, J. Slaya, B. P. Turnbull, and Y. Xieb, "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling," *Journal of Network and Computer Applications*, Vol.87, No.1, pp.185-192, 2017. https://doi.org/10.1016/j.jnca.2017.03.018

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique," *Journal of Artificial Inteligence Research(JAIR)*, Vol.16, No.1, pp.321-357, 2002.

[4] A. Makhzani, J. Shlens, N. Jaitly, L. Goodfellow, and B. Frey, "Adversarial autoencoders," *International Conference on Learning Representations*, San Juan, Puerto Rico, 2016, http://arxiv.org/abs/1511.05644

[5] S. Kim and S. Park, "Multi-class classification of database workloads using PCA-SVM classifier," *Journal of KIISE: Database*, Vol.38, No.1, pp.1-8, 2011.

[6] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdabau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing EMNLP*, Doha, Qatar, pp.1724-1734, 2014.

[7] Y. Cheong, K. Park, H. Kim, J. Kim, and S. Hyun, "Machine learning based intrusion detection systems for class imbalanced datasets," *Journal of the Korea Institute of Information Security and Cryptology*, Vol.27, No.6, pp.1385-1395, 2017. https://doi.org/10.13089/JKIISC.2017.27.6.1385

[8] M. Lee, "LSTM model based on session management for network intrusion detection," *Journal of The Institute of Internet, Broadcasting and Communication*, Vol.20, No.3, pp.1-7, 2020. https://doi.org/10.7236/JIIBC.2020.20.3.1

[9] M. Shahriar and N. Haque, "G-IDS: Generative adversarial networks assisted intrusion detection system," *IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp.376-385, 2020. https://doi.org/10.1109/COMPSAC48688.2020.0-218

[10] R. Corizzo, E. Zdravevski, M. Russell, A. Vagliano, and N. Japkowicz, "Feature extraction based on word embedding models for intrusion detection in network traffic," *Journal of Surveillance, Security and Safety*, Vol.1, pp.140-150, 2020. https://doi.org/10.20517/jsss.2020.15

[11] B. Min, J. Ryu, D. Shin, and D. Shin, "Improved network intrusion detection model through hybrid feature selection and data balancing," *KIPS Transactions on Software and Data Engineering*, Vol.10, No.2, pp.65-72, 2021. https://doi.org/10.3745/KTSDE.2021.10.2.65

[12] J. Lee and K. Park, "GAN-based imbalanced data intrusion detection system," Personal and Ubiquitous Computing, Vol. 25, pp.121-128, 2021. https://doi.org/10.1007/s00779-019-01332-y

[13] D. M. Reddy and N. V. S. Reddy, "Effects of padding on LSTMs and CNNs," arXiv:1903.07288v1, 2019. https://arxiv.org/pdf/1903.07288.pdf

[14] D. Senthil and G. Suseendran, "Efficient time series data classification using sliding window technique based improved association rule mining with enhanced support vector machine," *International Journal of Engineering and Technology(UAE)*, Vol.7, No.2, 2018. https://doi.org/10.14419/ijet.v7i2.33.13890

[15] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient estimation of word representations in vector space," *International Conference on Learning Representations*, AZ, USA, pp.1-12, 2013. http://arxiv.org/abs/1301.3781

[16] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Maui, HI, USA, pp.586-591, 1991. https://doi.org/10.1109/CVPR.1991.139758

[17] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, Vol.20, No.3, pp.273-297, 1995. https://dx.doi.org/10.1007%2FBF00994018

[18] S. Jo, H. Sung, and B. Ahn, "A comparative study on the performance of SVM and an artificial neural network in intrusion detection," *Journal of Korea Academia-Industrial Cooperation Society*, Vol.17, No.2, pp.703-711, 2016. https://doi.org/10.5762/KAIS.2016.17.2.703

[19] G. Nicole and J. Alfred, "Are GRU cells more specific and LSTM cells more sensitive in motive classification of text?," *Frontiers in Artificial Intelligence*, Vol.3, 2020. https://doi.org/10.3389/frai.2020.00040

박 경 선

https://orcid.org/0000-0002-5661-2888
e-mail : gseon130@ajou.ac.kr
2019년 한밭대학교 정보통신공학과(학사)
2020년 ~ 현 재 아주대학교
지식정보공학과 석사과정
관심분야 : 기계학습, 정보보안

김 강 석

https://orcid.org/0000-0001-8950-7577
e-mail : kangskim@ajou.ac.kr
2007년 인디애나대학교 컴퓨터공학(박사)
2010년 ~ 2016년 아주대학교
지식정보공학과 연구교수
2016년 ~ 현 재 아주대학교
사이버보안학과 부교수
관심분야 : 빅데이터 응용보안, 기계학습 및 딥러닝