*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.32, NO.2, Apr. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.2.201

# 다양한 데이터 전처리 기법과 데이터 오버샘플링을 적용한 GRU 모델 기반 이상 탐지 성능 비교*

유 승 태,[1†] 김 강 석[2‡]
[1]아주대학교 대학원 지식정보공학과 (대학원생), [2]아주대학교 사이버보안학과 (교수)

# Comparison of Anomaly Detection Performance Based on GRU Model Applying Various Data Preprocessing Techniques and Data Oversampling*

Seung-Tae Yoo,[1†] Kangseok Kim[2‡]
[1]Dept. of Knowledge Information Engineering (Graduate student),
[2]Dept. of Cyber Security, Ajou University (Professor)

## 요 약

최근 사이버보안 패러다임의 변화에 따라, 인공지능 구현 기술인 기계학습과 딥러닝 기법을 적용한 이상탐지 방법의 연구가 증가하고 있다. 본 연구에서는 공개 데이터셋인 NGIDS-DS(Next Generation IDS Dataset)를 이용하여 GRU(Gated Recurrent Unit) 신경망 기반 침입 탐지 모델의 이상(anomaly) 탐지 성능을 향상시킬 수 있는 데이터 전처리 기술에 관한 비교 연구를 수행하였다. 또한 정상 데이터와 공격 데이터 비율에 따른 클래스 불균형 문제를 해결하기 위해 DCGAN(Deep Convolutional Generative Adversarial Networks)을 적용한 오버샘플링 기법 등을 사용하여 오버샘플링 비율에 따른 탐지 성능을 비교 및 분석하였다. 실험 결과, 시스템 콜(system call) 특성과 프로세스 실행패스 특성에 Doc2Vec 알고리즘을 사용하여 전처리한 방법이 좋은 성능을 보였고, 오버샘플링별 성능의 경우 DCGAN을 사용하였을 때, 향상된 탐지 성능을 보였다.

## ABSTRACT

According to the recent change in the cybersecurity paradigm, research on anomaly detection methods using machine learning and deep learning techniques, which are AI implementation technologies, is increasing. In this study, a comparative study on data preprocessing techniques that can improve the anomaly detection performance of a GRU (Gated Recurrent Unit) neural network-based intrusion detection model using NGIDS-DS (Next Generation IDS Dataset), an open dataset, was conducted. In addition, in order to solve the class imbalance problem according to the ratio of normal data and attack data, the detection performance according to the oversampling ratio was compared and analyzed using the oversampling technique applied with DCGAN (Deep Convolutional Generative Adversarial Networks). As a result of the experiment, the method preprocessed using the Doc2Vec algorithm for system call feature and process execution path feature showed good performance, and in the case of oversampling performance, when DCGAN was used, improved detection performance was shown.

**Keywords:** Anomaly Detection, Preprocessing, Oversampling, GRU, DCGAN

Received(01. 17. 2022), Modified(03. 25. 2022), Accepted(03. 25. 2022)
* 본 연구는 한국연구재단의 지원을 받아 수행 하였습니다 (No. NRF-2019R1F1A1059036).
† 주저자, yst430@ajou.ac.kr
‡ 교신저자, kangskim@ajou.ac.kr(Corresponding author)

## I. 서 론

침입탐지시스템(IDS, Intrusion Detection System)은 외부에서 호스트로 들어오는 데이터 중 일반적인 접근이 아닌 공격을 위한 접근을 탐지하는 역할을 한다. 기존의 방화벽이라는 일차적인 방어시스템이 있지만, 사전에 정의된 패턴 탐지 및 규정 등을 사용하여 작동하기 때문에 한계가 있다. 또한, 방화벽은 들어오는 신호의 양이 버퍼의 용량을 초과하면 검사하지 않아 뚫리거나, 감지하지 못하는 경우가 발생한다[1]. 방화벽의 이러한 단점을 보완하기 위하여 침입탐지시스템을 사용한다. 일반적으로 침입탐지시스템은 HIDS(Host-based Intrusion Detection System)와 NIDS(Network based Intrusion Detection System)로 나눌 수 있다. NIDS는 네트워크상의 패킷 패턴을 분석 및 탐지하고, HIDS는 호스트로 접근한 데이터의 패턴을 분석 및 탐지한다[2]. 본 연구에서는 딥러닝 기반 HIDS를 제안한다. 최근 사이버보안 패러다임의 변화에 따라, 인공지능 구현 기술인 기계학습과 딥러닝 기법을 적용한 침입탐지 방법의 연구가 증가하고 있으며, 딥러닝 기반 침입탐지 모델을 구축한 연구들이 우수한 성능을 보이고 있다. 이때 딥러닝 모델에 적용하려면 일정한 길이의 특성(feature)들이 필요하다. Kyoto 2006+ 데이터 셋을 사용한 실험의 경우와 같이 일정한 길이의 특성을 가지고 있다면 딥러닝 모델에 적용하는 데 어려움이 없었다[3]. 하지만 이 실험에서 사용하는 공개 데이터셋인 NGIDS-DS(Next-Generation IDS Dataset)의 경우 시스템 콜 특성을 기존의 연구와 같이 가공하면 데이터의 길이가 일정하지 않는 문제점이 생긴다. 기존의 연구들은 이러한 문제를 빈도수를 통한 전처리를 통하여 문제를 해결한다. 하지만 이 실험에서는 자연어를 처리하듯이 전처리하여 이 문제를 해결하였고, 세 가지 전처리방법을 사용하여 실험을 진행하였다. 또한, NGIDS 데이터셋은 정상 데이터와 공격 데이터의 비율이 불균형을 이루고 있다. 이와 같은 클래스 불균형 문제를 해결하기 위하여 SMOTE(Synthetic Minority Oversampling TEchnique)[4]와 DCGAN(Deep Convolutional Generative Adversarial Networks)[5]을 사용하여 오버샘플링(Oversampling)을 수행하였다. 전처리 후, 일반적인 특성을 가진 데이터는 DNN (Deep Neural Network) 모델을 사용하였고, 순차적인 특징을 가진 데이터는 GRU(Gated Recurrent Units) 모델을 사용하여 분류학습을 진행하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구에 대하여 알아보고, 3장에서는 제안하는 연구 방법에 대하여 기술한다. 4장에서는 실험방법을 기술하고 5장에서 실험 결과와 분석 내용을 설명한다. 마지막으로 6장에서는 본 연구의 결론을 다룬다.

## II. 관련 연구

### 2.1 IDS

침입탐지시스템의 성능을 향상시키기 위한 연구가 지속적으로 진행되고 있으며, 최근에는 기계학습 및 딥러닝 모델을 적용한 연구가 좋은 성능을 보이고 있다. [6]의 연구는 KDD CUP 99 데이터 셋을 사용하여 딥러닝 알고리즘과 기계학습 알고리즘에 기반한 이상 탐지 모델들의 성능을 비교한 연구이다. 기계학습 모델의 경우 선형회귀, 나이브 베이즈 분류(Naïve Bayes Classification), K-최근접 이웃(K-Nearest Neighbor), 결정트리(Decision Tree), 랜덤 포레스트(Random Forest)와 같은 알고리즘을 사용하였고, 딥러닝의 경우 DNN 알고리즘을 사용하여 은닉층의 개수, 활성화 함수, 에폭(epoch)과 같은 하이퍼파라미터 (Hyperparameter)를 조정하면서 실험하였으며, 각각의 하이퍼파라미터에 대하여 성능과의 상관성을 연구하였다. 실험 결과 DNN을 통하여 정확도 93.8%의 성능을 보였다. [7]의 논문은 NSL-KDD 데이터 셋을 사용하여 기계학습 알고리즘과 논문에서 제시한 딥러닝 기반의 RNN-IDS 알고리즘의 분류 성능을 비교하였다. 기계학습 알고리즘은 나이브 베이즈, 랜덤 포레스트 등과 같은 알고리즘을 사용하였고, RNN-IDS의 경우 은닉 노드(hidden node)와 학습률(learning rate)을 조정하면서 실험을 진행하였고, 은닉 노드 수를 80개, 학습률을 0.5로 설정한 RNN-IDS 알고리즘을 사용하여 정확도 97.09%의 성능을 보였다. [8]의 논문은 KDD CUP 99 데이터 셋을 사용하여 딥러닝 모델에 적용하여 침입탐지 모델 실험을 진행하였다. DNN과 LSTM(Long Short-Term Memory) 알고리즘을 사용하여 분류를 진행하였고, SMOTE로 오버샘플링을 수행하였다. 실험결과 LSTM을 사용한 분류모델이 가장 좋

은 성능을 보였다.

[9]의 논문은 KDD CUP 99 데이터 셋을 사용하여 기존의 머신러닝 알고리즘인 랜덤 포레스트, SVM(Support Vector Machine)등과 논문에서 제안한 딥러닝 알고리즘인 S-NN(Shallow Neural Network Model)과 D-ONN(Deep-Optimized Neural Network Model)의 성능을 비교하였다. 실험결과 기존의 DLSTM(Distributed Long Short-Term Memory)을 사용한 실험이 F1-score 99.5로 가장 좋은 성능을 보였다. 앞에서 언급된 기존 관련연구에서 사용되어진 데이터가 본 연구에서 사용되어진 데이터와 달라 직접적인 비교가 어려우나 딥러닝 기반침입탐지시스템 연구사례를 확인할 수 있었다.

NGIDS-DS를 사용하여 침입탐지시스템을 연구한 논문으로써 [10]의 논문은 Word2Vec, TF-IDF(Term Frequency - Inverse Document Frequency) 그리고 Doc2Vec과 같은 자연어 처리 방법들을 조합하여 여러 전처리 과정을 수행한 뒤, 기계학습 알고리즘인 앙상블을 통하여 분류를 진행하였다. Word2Vec 알고리즘과 TF-IDF를 사용한 전처리방법에서 AUC(Area Uner Curve) 100%의 성능을 보였다. 본 연구와 유사하나, [10]의 논문에서는 일정하지 않은 길이의 특성을 빈도수를 사용하여 일정한 길이로 전처리하였고, 기계학습 알고리즘을 사용했다는 점에서 차이가 있으며, 또한 정상과 공격 데이터 사이에 클래스 불균형(9:1) 문제를 고려하지 않아 AUC 성능에 영향을 미치는 것으로 보이며, 따라서 본 연구에서는 클래스 불균형 문제를 해결하기 위하여 데이터의 오버샘플링을 고려하였다.

### 2.2 Doc2Vec

Doc2Vec(Document Embedding with Paragraph Vectors)은 Word2Vec의 확장된 개념으로 문장(document) 자체를 임베딩하는 알고리즘이다. Paragraph Matrix에 각각의 문장의 고유한 벡터를 랜덤으로 설정한다. 그 후 학습을 진행하고 학습이 완료되면 문장의 벡터값 이외의 파라미터는 고정한다. 문장의 고유한 벡터값을 문장에 따라 업데이트하여 임베딩을 진행한다[11]. 본 연구에서는 전처리과정에서 발생하는 행(row)의 길이가 일정하지 않은 문제를 해결하기 위하여 Doc2Vec 기법을 사용하였다.

### 2.3 DCGAN

DCGAN은 CNN(Convolutional Neural Network)을 기반으로 만들어진 GAN(Generative Adversarial Network) 알고리즘이다. CNN으로 생성자와 판별자를 구성하고, 이미지를 학습시킨 뒤, 생성자에 노이즈(noise)를 입력하여 기존의 데이터와 유사한 이미지를 생성하는 알고리즘이다. 생성자는 이미지를 점점 키워가며 가짜 데이터를 발생시키고, 판별자는 이미지를 점점 줄여가며 판별해낸다. 이와 같이 DCGAN은 2차원 데이터를 오버샘플링하는 알고리즘이다. 본 연구에서는 DCGAN을 변형시켜 1차원 데이터를 오버샘플링하는데 적용하였다.

## III. 연구 방법론

### 3.1 데이터셋 및 특성 선택

NGIDS-DS[12]는 호스트 로그 데이터셋으로 총 90,054,160개 중 7개 범주의 공격 클래스로 구성된 1,262,426개의 데이터가 있으며 속성으로 로그 발생시간, 프로세스 ID, 시스템 콜(system call), 이벤트 ID, 프로세스 실행패스(execution path), 클래스 레이블 등으로 구성되어 있다. NGIDS 데이터셋을 구성하는 속성들 중 정상과 공격을 판별하는데 유용하리라 판단되는 시스템 콜과 프로세스 실행패스를 분류 기반 탐지모델의 특성으로 선택하였다. 기존 연구에서는 프로세스 실행패스 특성을 사용하지 않았기 때문에 이 특성을 사용할 것인지에 대하여 판단하기 위하여 Cramer V 계수[13]를 사용하여 상관분석을 진행하였다. 시스템콜 특성과 종속변수(정상, 공격) 간의 Cramer V 계수가 0.022, 시스템콜 특성에 실행패스 특성을 추가 후 종속변수(정상, 공격) 간의 Cramer V 계수가 0.57로 실행패스 특성 또한 정상 데이터와 공격 데이터의 분류에 영향을 준다고 예상할 수 있다. 그렇기에 시스템 콜 특성과 패스 특성을 함께 사용한 경우와 그렇지 않은 경우로 구별하여 실험을 진행하였다.

### 3.2 데이터 전처리 및 분류 모델

NGIDS-DS는 순차적인 속성을 가지는 로그들을 저장한 데이터로써, 딥러닝 기반 분류 학습에 적용하기 위해서는 각 샘플이 일정한 길이가 되도록 전처리하는 것이 필요하다. 먼저 로그 데이터 수집 시간의 최소단위인 1초를 기준으로 각각 하나의 행(샘플)으로 가공하였으며, 또한 각 행을 일정한 길이로 변환하기 위하여 세 가지 전처리방법을 사용하였다. 첫 번째는 샘플들을 일정한 길이로 슬라이싱(slicing)하고, 나머지는 제로 패딩(zero padding)을 하였으며, 슬라이싱 길이를 100과 400으로 실험을 진행하였다. 일정한 길이로 가공한 후 정규화를 통하여 특성의 스케일을 조정한 다음 임베딩(embedding)을 하여 각 데이터의 특성을 분산표현(distributed representation)으로 나타내도록 하였다. 이렇게 전처리된 데이터는 순차 데이터 학습에 특화된 GRU(Gated Recurrent Unit) 알고리즘을 사용하여 분류 모델을 구축하였다.

두 번째는 Doc2Vec 알고리즘을 사용하여 일정한 길이로 전처리하였다. Doc2Vec 알고리즘은 한 문장의 특징을 추출하여 일정한 벡터 길이(vector size)로 나타내는 임베딩 알고리즘이다. 임베딩 변환 후에 정규화를 진행하였고, Doc2Vec을 사용하여 전처리 된 데이터는 일반적인 특성을 가진 데이터로 변환되기 때문에 DNN(Deep Neural Network) 알고리즘을 활용하여 분류학습을 진행하였다. 첫 번째와 두 번째에서 제시된 모델의 학습 속도는 두 실험에서 전처리한 데이터의 차원 수가 상이하여 큰 차이가 있을 것으로 예상한다. 첫 번째 실험의 경우 임베딩을 통하여 2차원에서 3차원의 벡터로 변환되고, 반면에 두 번째 실험의 경우 Doc2Vec을 통하여 벡터의 길이가 바뀔 뿐 차원의 수는 유지되기 때문이다. 즉, 첫 번째 실험이 두 번째 실험보다 더 많은 학습시간을 필요로 할 것이다. 위 내용을 바탕으로 학습시간이 적게 소요되는 두 번째 실험의 특성에 실행패스 특성을 추가하여 세 번째 실험을 설계하였다. Fig. 1은 본 연구에서 제안하는 연구 방법의 흐름도(workflow)이다. Table 1은 본 연구의 실험 구성표이다.

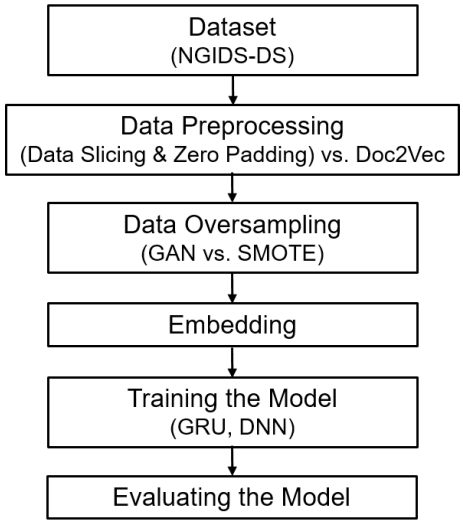


Fig. 1. A simplified workflow for proposed anomaly detection methodology

Table 1. Experimental Configuration

| | Preprocessing methods | Classification algorithms |
|---|---|---|
| #1 | Slicing | GRU |
| #2 | Doc2Vec | DNN |
| #3 | Doc2Vec + Path | DNN |

### 3.3 오버샘플링

사용된 로그 데이터는 공격 데이터와 정상 데이터의 비율이 1:90으로 공격벡터가 적다. 따라서 클래스 불균형 문제로 딥러닝 모델이 학습을 제대로 수행하기 어렵기 때문에 공격 데이터에 오버샘플링을 적용하여, 공격 데이터와 정상 데이터의 샘플수가 일정 비율이 되도록 공격 데이터를 생성하였다. 공격 데이터와 정상 데이터의 비율은 0.3, 0.5, 0.7, 1.0으로 선정하였고, 각각의 경우에 대한 성능을 비교하였다. 분류를 진행할 때는 이진 분류를 진행할 예정이나, 오버샘플링을 진행할 때는 각 7개의 공격 데이터 수의 비율에 비례하여 오버샘플링을 하였다. 두 가지 오버샘플링 방법을 사용하였다. 첫 번째는 DCGAN을 사용한 오버샘플링이다. DCGAN은 2차원 Convolution Layer와 배치정규화 계층의 조합으로 생성자와 판별자를 구축하는 모델로 본 연구에서는 1차원 데이터를 오버샘플링 하는데 사용하였다. 두 번째는 SMOTE를 사용한 오버샘플링이다. SMOTE는 기존의 데이터 중 랜덤으로 두 개를 선정하고, 그 중간에 데이터를 생성한다. 이것을 반복하여 비슷한 데이터를 생성한다. DCGAN의 성능과

비교하기 위하여 사용하였다.

## IV. 실험방법

### 4.1 실험환경

실험환경은 Table 2와 같다.

Table 2. Experimental environment

| Specification | |
|---|---|
| OS | Windows 10 |
| Framework | Anaconda |
| CPU | Intel(R) Core(TM) i7-1065G7 CPU @ 1.30GHz |
| Ram | 16GB |
| Library | TensorFlow 2.8.0 |

### 4.2 데이터 전처리

딥러닝 기반 이상 탐지 모델에 입력으로 주입하기 위한 시퀀스 데이터 전처리 방법으로 슬라이싱과 제로 패딩을 적용한 방법과 Doc2Vec을 적용한 방법을 사용하였다. 슬라이싱의 길이로는 100일 때와 400일 때를 비교하였다. 순차 데이터를 1초 기준으로 하나의 시퀀스(샘플)를 가지는 데이터로 가공하였을 때, 공격 데이터 시퀀스 길이를 기준으로 시퀀스의 분포가 길이 100과 길이 400 이하에 많이 분포되어, 슬라이싱 길이를 100과 400으로 선정 후, 임베딩을 적용하여 행은 샘플 수, 열은 시퀀스 길이, 깊이는 임베딩 벡터 길이로 구성된 3차원 데이터로 가공했다. Doc2Vec을 적용한 전처리 방법에서는, 각각 다른 길이의 특성을 Doc2Vec 알고리즘을 사용하여 일정 길이의 벡터로 변환하였다. 이때 벡터의 길이를 100, 200, 300으로 적용하여 길이에 따른 변화를 실험하였다. 그 이유는 Doc2Vec을 적용하기 전 데이터들의 길이 빈도수가 주로 400 이하에 분포하기 때문이다. 이를 통하여 행은 샘플 수, 열은 벡터 길이로 구성된 2차원 데이터로 가공하였다.

### 4.3 오버샘플링

Fig. 2는 실험에 사용된 DCGAN의 생성자이다. 실험에서는 1차원 데이터를 학습시키기 위하여 2개의 1D Convolution layer를 배치하여 생성자(generator)와 판별자(discriminator)를 구축하였다. Convolution layer를 두 개의 층으로 구현한 이유는 DCGAN의 생성자는 Upsampling으로 노이즈의 데이터 형태가 2배씩 증가하기 때문이다. 즉, 노이즈의 길이는 층이 늘어날 때마다 최종적으로 생성하고자 하는 데이터 길이의 1/2 길이가 되어야 한다. 이때 생성하고자 하는 데이터의 길이가 100, 200, 300이므로 최대 구축할 수 있는 Convolution layer는 2개의 층이다. Fig. 3는 판별자이다. 생성자와 마찬가지로 2개의 1D Convolution layer를 사용하였고, Dropout Layer를 통하여 안정적으로 학습하도록 하였다. 그 후 Sigmoid 함수를 사용하여 분류를 진행하였다. SMOTE는 Imblearn 패키지에서 제공하는 SMOTE 함수를 사용하여 구현하였다.

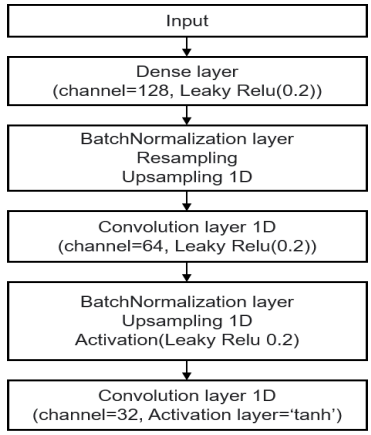


Fig. 2. DCGAN generator

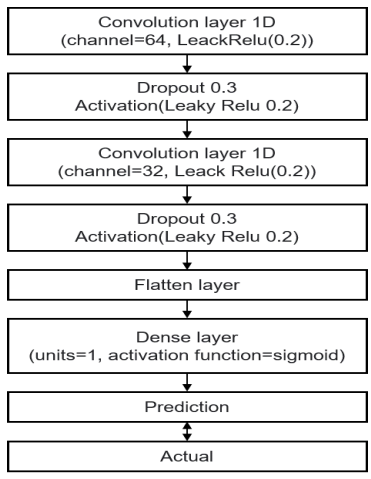


Fig. 3. DCGAN discriminator

### 4.4 분류 모델

본 실험에서는 딥러닝 모델을 구축하기 위하여 텐서플로우[14] 라이브러리를 활용하였다. Fig. 4는 첫 번째 실험을 위해 GRU를 사용하여 구축한 모델이다. 전처리된 데이터가 임베딩 레이어를 거쳐 분산 표현으로 변환된다. 그 후 두 개의 GRU layer를 거친다. layer 수의 경우 1~3개일 때를 비교하였고, 층이 1개인 경우 보다 2개와 3개인 경우가 더 좋은 성능을 보였으며, 2개일 때와 3개일 때는 거의 차이가 없어 2개로 선정하였다. 모델의 과적합(overfitting)을 대비하기 위하여 층마다 드롭아웃(dropout) layer를 배치하였으며 학습 시 무작위로 노드들을 드롭하기 위한 비율은 DCGAN의 고려사항 및 실험을 통하여 0.3 정도를 사용하였다. 마지막으로 Sigmoid를 활성화 함수로 하는 Dense layer를 배치하여 분류학습을 진행하였다. 에폭은 20, Optimizer는 Adam, 손실(loss) 함수는 이진 교차 엔트로피(Binary Cross Entropy)를 사용하였다.

Fig. 5는 두 번째와 세 번째 실험을 위하여 DNN을 사용한 모델이다. Doc2Vec을 적용하여 전처리 된 데이터가 입력으로 모델에 주입된 후 두 개의 Dense Layer를 거친다. 두 개로 선정한 이유는 첫 번째 실험과 같은 이유이다. 활성화 함수는 relu를 사용하였다. Dense Layer의 다음에 드롭아웃 Layer를 배치하여 과적합을 대비하였다. 그 후 Sigmoid를 활성화 함수로 하는 Dense Layer를 통하여 분류학습을 하였다. 에폭은 20, Optimizer는 Adam, 손실함수는 이진 교차 엔트로피를 사용하였다.

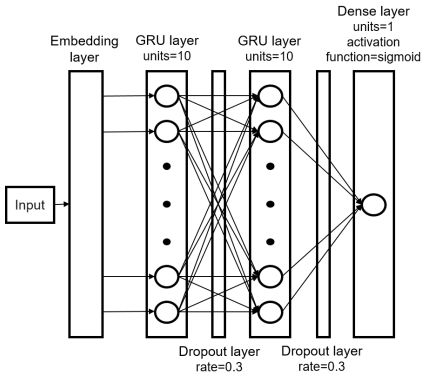


Fig. 4. GRU based classification model

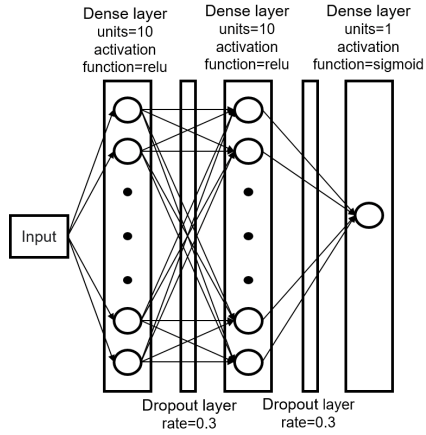


Fig. 5. DNN based classification model

### 4.5 성능 지표

훈련, 검증, 테스트 데이터셋의 적절한 비율을 위해 충분한 데이터셋이 있을때는 훈련 데이터를 많이 사용할 수도 있으나[15] 본 실험에서는 오버샘플링 후 전형적으로 사용되어지는 훈련:검증:테스트 데이터를 6:2:2 정도의 비율로 나누어 실험을 진행하였다. 성능 지표로 사용된 재현율(recall)은 실제 공격 데이터를 분류 모델이 공격으로 예측한 것의 비율이다. 침입탐지시스템의 경우 정상을 공격으로 오인했을 때 보다 공격을 정상으로 오인했을 때 더 치명적이므로 재현율을 통하여 성능을 측정하였다[3]. AUC는 ROC(Receiver Operating Characteristic) 곡선 아래 면적으로서, 분류 문제에서 클래스별로 그 수가 다를 때 정확도(accuracy)의 단점을 보완하여 성능을 평가하는 지표이다[16]. 다음 5장에서는 4장에서 언급한 방법으로 전처리된 데이터셋을 사용하여 얻은 실험 결과를 기술할 것이다.

## V. 실험 결과 및 분석

### 5.1 슬라이싱과 제로 패딩을 사용한 전처리 모델의 실험 결과

Fig. 6은 슬라이싱과 제로 패딩을 사용하여 전처리후 SMOTE와 DCGAN을 사용하여 오버샘플링한 데이터를 GRU 알고리즘을 적용한 모델에 주입하여 실험한 결과로써, 오버샘플링 방법, 오버샘플링

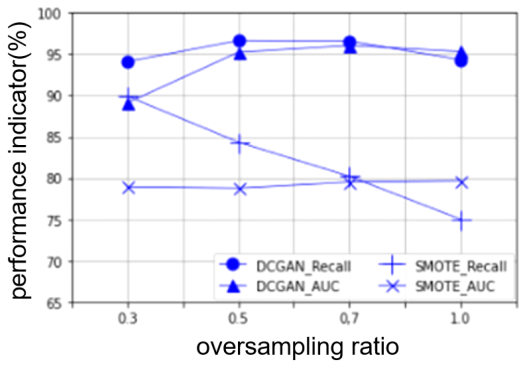


Fig. 6. Comparison of GRU classifier performance for each oversampling ratio with slicing and zero padding preprocessing method

비율 그리고 성능지표별로 평균을 내어 그래프로 나타낸 것이다. 가로축은 오버샘플링 비율, 세로축은 성능지표별 지표이다. 이를 통하여 재현율은 오버샘플링의 비율이 증가할수록 감소하는 것을 알 수 있다. 이것은 오버샘플링을 통하여 비교할 대상이 증가하면서 생기는 현상이라고 할 수 있다. 반면, AUC는 증가하거나 비슷한 수준으로 유지되는 것을 알 수 있다. 오버샘플링의 비율이 0.3에서 0.5로 증가할 때는 성능이 소폭 증가하지만, 그 후에는 거의 유지되는 것을 알 수 있다. 오버샘플링의 비율이 0.5 정도만 되어도 충분한 성능을 낸다고 판단할 수 있다. 오버샘플링 방법에 대하여 비교하였을 때, SMOTE는 약 80%의 성능을 보이지만, DCGAN은 약 93%의 성능을 보이며 DCGAN을 사용하였을 때, 더 좋은 성능을 보이는 것으로 나타났다.

Fig. 7은 슬라이싱 길이와 성능지표별로 평균을 낸 그래프이다. 가로축은 슬라이싱 길이, 세로축은 성능지표이다. 슬라이싱 길이가 100일 때보다 400일 때 더 좋은 성능을 보였다. 이것은 100일 때보다 400일 때 분류에 적절한 충분한 특징을 가지고 있다고 할 수 있다.

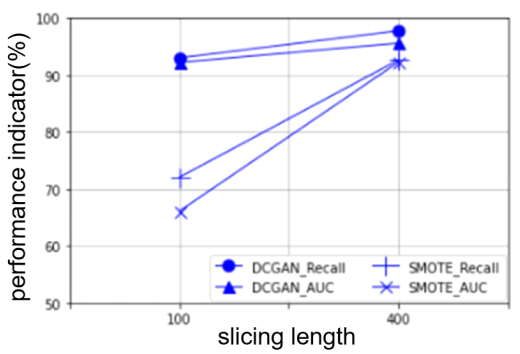


Fig. 7. Performance by slicing length with oversampling techniques

### 5.2 Doc2Vec을 사용한 전처리 모델의 실험 결과

Fig. 8은 Doc2Vec을 사용하여 전처리후 SMOTE와 DCGAN을 사용하여 오버샘플링 한 데이터를 DNN 알고리즘을 적용한 모델에 주입하여 실험한 결과로써, 오버샘플링 방법, 오버샘플링 비율 그리고 성능지표별로 평균을 낸 그래프이다. 가로축은 오버샘플링 비율, 세로축은 성능지표별 지표이다. 그래프의 추이는 5.1절의 실험과 크게 다르지 않은 것으로 확인되었다. 하지만, 오버샘플링 방법에 따른

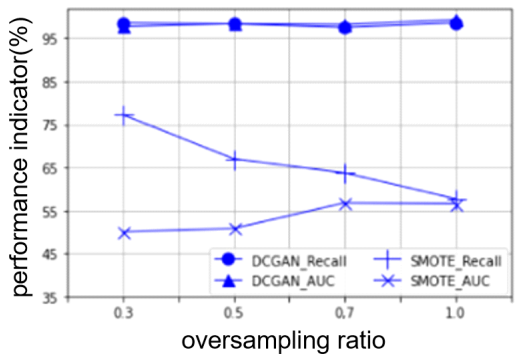


Fig. 8. Comparison of GRU classifier performance for each oversampling ratio with Doc2Vec preprocessing method

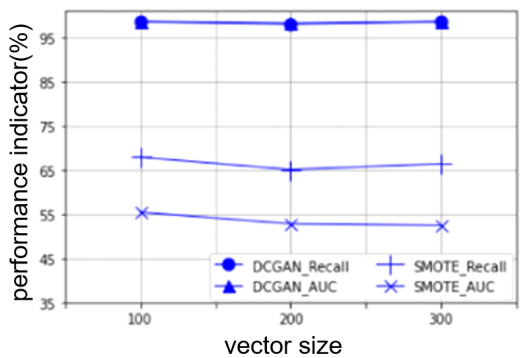


Fig. 9. Performance by vector size with oversampling techniques

성능의 차이가 나타났다. DCGAN을 통한 실험의 결과 평균 약 98%의 성능을 보였고, SMOTE를 통한 실험의 결과는 평균적으로 약 55%의 성능을 보였다. Fig. 9는 Doc2Vec에 의해서 변환된 벡터 길이에 따른 성능을 비교한 그래프이다. 그래프를 통하여 벡터 길이에 따른 성능의 변화 차이가 없음을 알 수 있다. 이를 통하여 벡터 길이가 100이어도 충분히 데이터의 특징을 일정한 길이로 나타낼 수 있음을 알 수 있다.

### 5.3 프로세스 실행패스 특성을 추가한 후, Doc2Vec을 사용한 전처리 모델의 실험 결과

Fig. 10의 결과는 전처리 방법과 분류 모델이 5.2절의 실험과 유사하나 패스 특성을 추가하여 전처리한 후 실험한 결과이다. 두 그래프(Fig. 8과 Fig. 10)는 패스 특성의 사용 여부에 따른 차이로 그래프 상의 추이는 비슷한 것으로 확인되었다. 하지만 평균적으로 패스 특성을 사용했을 때 DCGAN을 사용한 모델의 성능은 평균적으로 약 99% 이상이었고, SMOTE를 사용한 모델의 성능은 평균적으로 약 75%의 성능을 보였다. 위 내용을 바탕으로 패스 특성을 사용하지 않았을 때보다 사용하였을 때 성능이 향상되는 것을 확인할 수 있다.

Fig. 11은 Doc2Vec을 적용하여 변환된 벡터 길이에 따라 성능을 비교한 그래프이다. Fig. 9와 마찬가지로 벡터 길이에 따른 성능의 변화는 거의 없음을 알 수 있다. 하지만 성능이 약간 향상되었음을 알 수 있다. Fig. 12는 두 번째 실험의 결과와 세 번째

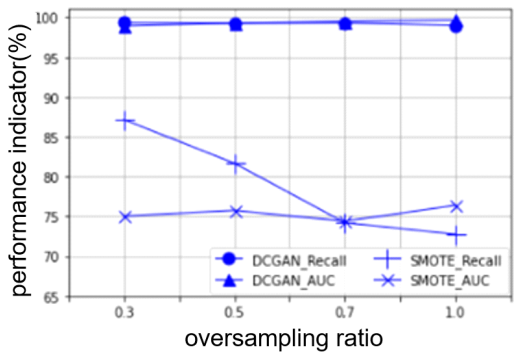


Fig. 10. Comparison of GRU classifier performance for each oversampling ratio with Doc2Vec preprocessing method after adding the process execution path feature

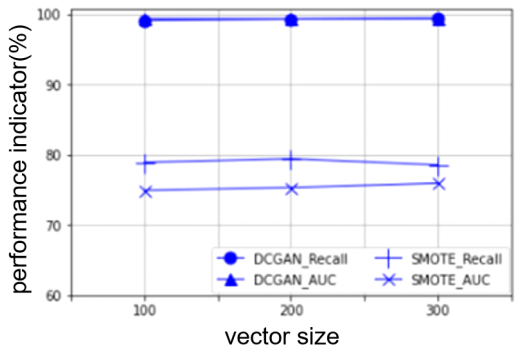


Fig. 11. Performance by vector size with oversampling techniques after adding the process execution path feature

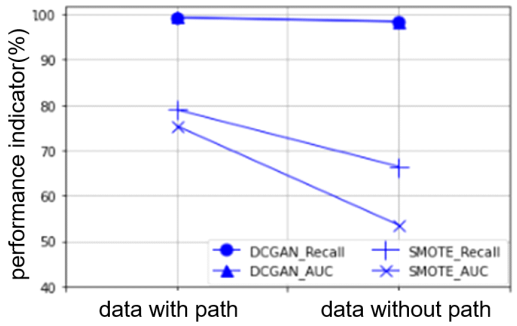


Fig. 12. Comparison of performance with or without process execution path feature

실험의 결과를 성능지표와 오버샘플링 방법별로 평균을 내어 나타낸 것이다. 패스 특성을 사용하였을 때 성능이 향상되었다는 것을 확인할 수 있다.

Table 3은 앞서 언급된 전처리방법과 오버샘플링 방법들의 조합으로 각각의 최대 성능을 나타낸 표이다. 가장 좋은 성능을 보인 조합은 실행패스 특성을 포함하고 Doc2Vec을 적용하여 전처리한 후 DCGAN으로 오버샘플링을 한 것으로 99.69%의 성능을 보였다.

딥러닝 기반 이상 탐지를 위하여 다양한 요소를 고려한 앞에서 언급된 실험 결과에 따라, 유용한 특성의 추출뿐만 아니라 시퀀스 특성을 가지는 데이터의 전처리 방법, 시퀀스 데이터 각각의 이벤트 원소들의 임베딩 벡터 방법, 일반적으로 이상(anomaly) 데이터가 부족한 경우에 대처하기 위한 처리 방법(비지도 혹은 준지도 학습 방법) 등을 고려한 학습 모델을 고려하는 것이 필요하다. 향후 본 실험에서 사용

Table 3. Maximum performance for each pre-processing method and oversampling method

| Preprocessing methods | Oversampling methods | AUC score |
|---|---|---|
| slicing | SMOTE | 92.48 |
| | DCGAN | 98.63 |
| Doc2Vec | SMOTE | 58.81 |
| | DCGAN | 99.33 |
| Doc2Vec +Path | SMOTE | 77.45 |
| | DCGAN | 99.69 |

한 요소들 외에도 다양한 요소들을 고려하여 탐지 성능의 개선을 위한 연구를 진행할 것이다.

## VI. 결 론

본 연구에서는 공개 데이터인 NGIDS 데이터 셋을 사용하여 이상 탐지(anomaly detection)를 위한 GRU 기반 딥러닝 모델을 구축하였다. 딥러닝 모델에 적용하기 위하여 사용된 전처리 방법으로 첫 번째는 슬라이싱과 제로 패딩을, 두 번째 방법은 Doc2Vec을, 세 번째 방법은 두 번째 방법에 데이터 특성을 추가하는 방법으로 수행하였다. 또한 클래스 불균형 문제를 해결하기 위하여 SMOTE와 DCGAN을 적용하여 오버샘플링을 하였다.

실험 결과에 따라 프로세스 실행 패스 특성을 추가로 사용하고, Doc2Vec을 적용하여 전처리한 후 DCGAN으로 오버샘플링 한 조합이 AUC 99.69%의 성능을 보였다. 이를 통하여 시스템 콜 데이터뿐만 아니라 프로세스 실행패스 특성 또한 이상 탐지 성능 향상에 유용하다는 것을 알 수 있었고, SMOTE보다 DCGAN이 더 좋은 성능을 보이는 것을 확인하였다. 그러나 GAN의 데이터 생성적 알고리즘을 활용하여 공격 벡터의 패턴을 가지는 입력 데이터와 유사한 생성적 합성된 표현을 얻는 것은 데이터의 증식뿐만 아니라 새로운 공격 데이터로써 (실제 공격 데이터 분포와 유사한 그러나 새로운 데이터 분포) 사용되어질 수 있으나, 생성된 데이터 셋이 실제 공격 데이터 샘플들을 대체할 수 있는지에 대한 검증 연구가 필요하다.

또한 향후 연구로 시퀀스 모델에 길이가 다른 입력 데이터를 주입하기 위한 마스킹 방법과 시퀀스 데이터 각각의 이벤트 원소들의 문맥을 고려한 임베딩 방법의 개발을 통해서, 이상탐지 모델의 성능을 향상시키기 위한 연구를 진행할 것이다.

## References


[1] Kyung-hyun Han and Seong-oun Hwang, "Development of firewall system for automated policy rule generation based on machine learning," Journal of The Institute of Internet, Broadcasting and Communication, 20(2), pp. 29-37, April 2020.

[2] K. Rahul-Vigneswaran, P. Poornachandran, and KP. Soman, "A compendium on network and host based intrusion detection systems," Proceedings of the 1st International Conference on Data Science, Machine Learning and Applications, pp. 23-30, May 2020.

[3] Yun-gyung Cheong, Ki-nam Park, Hyun-joo Kim, Jong-hyun Kim and Sang-won Hyun, "Machine learning based intrusion detection systems for class imbalanced datasets," Electronics and Telecommunications Research Institute, 27(6), pp. 1385-1395, Dec. 2017.

[4] S. Mishra, "Handling imbalanced data: SMOTE vs. random undersampling," International Research Journal of Engineering and Technology, vol. 4, no. 8, pp. 317-320, Aug. 2017.

[5] A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," International Conference on Learning Representations, pp. 1-16, Jan. 2016.

[6] Hyun Kwon, Seung-ho Bang and Ki-woong Park, "A design of deep neural network-based network intrusion detection system," Journal of KING Computing, 16(1), pp. 7-18, Feb. 2020.

[7] C. Yin, Y. Zhu, J. Fei and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," IEEE Access, vol. 5, pp. 21954-21961, Nov. 2017.

[8] Jae-hyun Seo, "A comparative study on the classification of the imbalanced intrusion detection dataset based on deep learning," Journal of Korean Institute of Intelligent System, 28(2), pp. 152-159, April 2018.

[9] M. Ramaiah, V. Chandrasekaran, V. Ravi and N. Kumar, "An intrusion detection system using optimized deep neural network architecture," Transactions on Emerging Telecommunications Technologies, vol. 32, no. 4, pp. 1-17, Feb. 2021.

[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, Dec. 2020.

[11] A. M. Dai, C. Olah, and Q. V. Le, "Document embedding with paragraph vectors," arXiv:1507.07998, 2015.

[12] W. Haider, J. Hu, J. Slay, B.P. Turnbull and Y. Xie, "Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling," Journal of Network and Computer Applications, vol. 87, no. 1, pp. 185-192, June 2017.

[13] H. Akoglu, "User's guide to correlation coefficients," Turkish Journal of Emergency Medicine, vol. 18, no. 3, pp. 91-93, Aug. 2018.

[14] N. Quang-Hung, H. Doan and N. Thoai, "Performance evaluation of distributed training in tensorflow 2," International Conference on Advanced Computing and Applications, pp. 155-159, Nov. 2020.

[15] A. Ng, "Sizeof dev and test sets (C3W1L06)," 2017. https://github.com/hithesh111/Hith100/blob/master/100Days/day035.ipynb

[16] R. A. Maxion and R. R. Roberts, "Proper Use of ROC Curves in Intrusion / Anomaly Detection," University of Newcastle upon Tyne, Computing Science Tyne, UK, p. 33, 2004.

## 〈저자소개〉

유 승 태 (Seung-Tae Yoo) 학생회원
2020년 2월: 아주대학교 산업공학과 졸업
2020년 3월~현재: 아주대학교 대학원 지식정보공학과 석사과정
〈관심분야〉 정보보호, 기계학습 및 딥러닝

김 강 석 (Kangseok Kim) 정회원
2007년 11월: Indiana University at Bloomington 컴퓨터 공학(박사)
2010년 9월~2016년 2월: 아주대학교 대학원 지식정보공학과 연구교수
2016년 3월~현재: 아주대학교 사이버보안학과 부교수
〈관심분야〉 빅데이터 응용보안, 기계학습 및 딥러닝