KIPS Trans. Softw. and Data Eng.
Vol.9, No.12 pp.411~418
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2020.9.12.411

A Pre-processing Study to Solve the Problem of Rare Class Classification of Network Traffic Data

Ryu Kyung Joon† · Shin DongIl†† · Shin DongKyoo†† · Park JeongChan††† · Kim JinGoog††††

ABSTRACT

In the field of information security, IDS(Intrusion Detection System) is normally classified in two different categories: signature-based IDS and anomaly-based IDS. Many studies in anomaly-based IDS have been conducted that analyze network traffic data generated in cyberspace by machine learning algorithms. In this paper, we studied pre-processing methods to overcome performance degradation problems cashed by rare classes. We experimented classification performance of a Machine Learning algorithm by reconstructing data set based on rare classes and semi rare classes. After reconstructing data into three different sets, wrapper and filter feature selection methods are applied continuously. Each data set is regularized by a quantile scaler. Depp neural network model is used for learning and validation. The evaluation results are compared by true positive values and false negative values. We acquired improved classification performances on all of three data sets.

Keywords : Machine Learning, Rare Class, Semi Rare Class, Pre-processing, Feature Selection

네트워크 트래픽 데이터의 희소 클래스 분류 문제 해결을 위한 전처리 연구

류 경 준† · 신 동 일†† · 신 동 규†† · 박 정 찬††† · 김 진 국††††

요 약

정보보안을 위한 IDS(Intrusion Detection Systems)는 통상적으로 서명기반(signature based) 침입탐지시스템과 이상기반(anomaly-based) 침입탐지시스템으로 분류한다. 이 중에서도 네트워크에서 발생하는 트래픽 데이터를 기계학습으로 분석하는 이상기반 IDS 연구가 활발하게 진행됐다. 본 논문에서는 공격 유형 학습에 사용되는 데이터에 존재하는 희소 클래스 문제로 인한 성능 저하를 해결하기 위한 전처리 방안에 대해 연구했다. 희소 클래스(Rare Class)와 준 희소 클래스(Semi Rare Class)를 기준으로 데이터를 재구성하여 기계학습의 분류 성능의 개선에 대하여 실험했다. 재구성된 3종의 데이터 세트에 대하여 Wrapper와 Filter 방식을 연이어 적용하는 하이브리드 특징 선택을 수행한 이후에 Quantile Scaler로 정규화를 처리하여 전처리를 완료한다. 준비된 데이터는 DNN(Deep Neural Network) 모델로 학습한 후 TP(True Positive)와 FN(False Negative)를 기준으로 분류 성능을 평가했다. 이 연구를 통해 3종류의 데이터 세트에서 분류 성능이 모두 개선되는 결과를 얻었다.

키워드 : 기계학습, 희소 클래스, 준 희소 클래스, 전처리, 특징 선택

1. 서 론

네트워크 침입 탐지의 목적은 악성 활동을 식별하고 모니터링하는 것이다. 현재 IDS는 크게 서명기반(signature based) IDS와 이상 기반(anomaly based) IDS 두 가지 유형으로 구분할 수 있다. 서명 기반 IDS는 이미 알려진 공격과 새로 유입되는 네트워크 트래픽을 비교 분석하고 분석 패턴 DB(DataBase)에 저장하는 형태로 다양한 패턴의 침입에 대해 탐지 할 수 있게 된다. 이상 기반 IDS는 기계학습(Machine Learning)을 통해 훈련 단계를 거쳐 학습된 패턴을 통해 식별하는 IDS 모델이다[1].

네트워크 트래픽 데이터를 분석하는 기계학습 연구에 있어서 가장 큰 문제점은 희소 클래스를 포함한 데이터 불균형(Data Imbalance)이다. 데이터 불균형 문제는 한 데이터 세트 내에서 유형별 샘플 수가 균형 잡혀있지 않는 것을 말한다. 전체 데이터 중 1% 이하의 비중을 차지하는 희소 클래스

※ 본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다(UD 190016ED).
※ 이 논문은 2020년 한국정보처리학회 춘계학술발표대회에서 “희소 클래스 분류 문제 해결을 위한 전처리 연구”의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : 세종대학교 컴퓨터공학과 석사과정
†† 종신회원 : 세종대학교 컴퓨터공학과 교수
††† 비 회 원 : 국방과학연구소 책임연구원
†††† 비 회 원 : 국방과학연구소 선임연구원
Manuscript Received : July 22, 2020
First Revision : November 3, 2020
Accepted : November 19, 2020
* Corresponding Author : Ryu Kyung Joon(rkj6663@sju.ac.kr)

(rare class)가 전체 분류기의 성능을 저하시키는 가장 큰 문제로 작용하고 있다. 이를 해결하기 위해 인공적으로 유사한 환경을 구축하여 데이터를 생성할 수도 있지만, 그 역시도 인적 자원이나 물적 자원과 같은 물리적인 한계에 부딪혀 어려움을 겪는 경우가 많다[2, 3].

Seo [2]의 연구에서 희소 클래스를 포함한 데이터 불균형 문제를 해결하기 위해 데이터 재구성 방법인 SMOTE (Synthetic Minority Oversampling Technique)를 사용하여 성능을 개선했다.

본 논문에서는 네트워크 정상 트래픽과 비정상 트래픽으로 구성된 데이터 세트의 희소 클래스(Rare Class)에 해당하는 공격 트래픽 데이터의 분류 성능을 높이기 위한 연구에 초점을 두고 데이터 구성 및 데이터 전처리와 딥러닝 분류 알고리즘을 통한 성능 개선 실험을 진행했다.

2. 관련 연구

2.1 네트워크 침입 탐지 데이터 세트

NSL-KDD 데이터 세트는 39개의 공격 유형을 일반적으로 4개(DoS, U2R, R2L, Probe)의 공격 유형 그룹과 1개의 정상 클래스로 전처리 작업[4]을 진행하고 학습을 진행한다. 또한, CSE-CIC-IDS 2018 데이터 세트는 15개의 공격 유형으로 이루어져 있고, DoS나 Web Attack과 같이 같은 계열의 클래스는 병합하는 등의 전처리 작업을 수행한 연구가 있다. CSE-CIC-IDS 2018 데이터 세트는 80개의 특징을 가지고 있으며, 다른 데이터 세트에 대해 비교적 최신 공격 데이터를 반영한 데이터 세트[5]로 네트워크 침입 탐지 시스템의 최신 연구 동향에 알맞은 데이터로 사용되는 추세다.

2.2 다양한 전처리 연구

ABDULAHEEM[3]의 연구에서는 신경망 알고리즘을 통해 분류하기 전에 데이터 세트에서 학습의 효율이나 성능을 저해하는 특징을 중요도(Importance)와 상관관계(Correlation)를 통해 분석하고 데이터의 분류 성능을 크게 개선했다. 정규화(Normalization)는 학습에 있어 필수적인 사전 작업이다. 방법에 따라 분류기 성능에도 영향을 미친다[6]. 정규화 방법은 Min-Max Scaler, Standard Scaler, Quantile Scaler가 대표적이다.

Equation (1)은 Min-Max Scaler 공식의 한 예이다.

$$f_{sc} = \frac{f - f_{\min}}{f_{\max} - f_{\min}} \tag{1}$$

Equation (2)은 Standard Scaler 공식의 한 예이다.

$$fz\_scre = \frac{f - \mu f}{\sigma f} \tag{2}$$

Equation (3)은 Quantile Scaler 공식의 한 예이다.

$$h\ observation = q(n+1) \tag{3}$$

Equation (1), (2)에서 f는 데이터의 특징(Feature)값을 의미한다. 서로 다른 단위를 통일시켜주는 정규화 방법이 학습 성능에 미치는 영향과 우수한 정규화 방법을 알아보기 위해 ABDULAHEEM[2]은 Equation (1), (2), (3)의 3가지 정규화 방법에 대해서 실험을 진행했다.

Equation (1)은 사용자가 지정한 특정 범위에서 원본 데이터에 대해 선형 변환을 수행하기 위한 수식이다. 이 방법의 장점은 데이터값의 모든 관계를 정확하게 보존하는 방법으로, 데이터에 어떠한 잠재적 편향도 도입하지 않는 방법이다[7]. Equation (2)는 평균을 제거하고 단위 분산으로 스케일링하여 서로 다른 특징을 표준화하는 방법의 기본 형태인 수식이다[8]. Equation (3)은 특정 누적 분포 함수 추정값을 사용하고 원본값을 균일분포에 매핑시켜서 값을 얻는 방식이며, Quantile 함수로 원하는 출력 분포에 매핑한다[9]. 특징 사이의 관계가 심장의 조직과 간의 조직을 비교하는 것과 같은 차이를 갖는 경우엔 사용하기 적절하지 않고, 특징을 균일하거나 정규 분포를 따르도록 변환하는 방법이다[10]. 3가지 정규화 방법으로 실험한 결과. Equation (3)의 Quantile Scaler로 정규화를 수행한 데이터의 분류 성능이 희소 클래스 분류에서 가장 우수하게 나타났다[3].

Devan [11]은 잘 알려진 네트워크 트래픽 데이터인 NSL-KDD 데이터를 사용하고 XGBoost 알고리즘으로 Feature Selection을 수행한 후, DNN 알고리즘을 통해 IDS 모델을 구현하고 성능을 평가했다.

Qazi [12]은 네트워크 트래픽 데이터인 KDD CUP99의 데이터 불균형 문제로 인한 성능 저하를 해결하기 위해 SMOTE (Synthetic Minority Oversampling Technique) 기법을 사용했다.

3. 데이터 소개 및 데이터 전처리

3.1 데이터 세트 재구성

CSE-CIC-IDS 2018 데이터 세트의 유형별 구성은 가장 많은 정상 데이터(BENIGN)가 전체 데이터 세트의 80%이고, 가장 적은 공격 데이터(Heartbleed)는 0.0004%로 데이터의 불균형 문제를 가진 데이터 세트이다[5]. 이처럼 데이터의 불균형이 심각한 경우, 샘플이 적은 클래스의 데이터들이 샘플 수가 많은 클래스의 데이터에 억눌려서 데이터가 제대로 학습되지 못하게 되어 전체적인 성능에 악영향을 끼친다.

본 논문에서는 1,500개 미만의 샘플을 가진 클래스의 데이터를 희소 클래스(Rare Class)라고 정의하고[2], 10,000개 미만의 샘플을 준 희소 클래스(Semi-Rare Class)로 구분하여 해당 클래스에 대한 분류 성능을 개선하기 위해 〈Table 1〉과

Table 1. The Number of Classes and Samples in the Reconstructed Data Set

Distinction	Class Name	Instance
Origin Data (Set A) 15 Class	BENIGN	2687419
	PortScan	317860
	DDoS	256054
	DoS Hulk	231073
	DoS GoldenEye	10293
	FTP-Patator	7938
	SSH-Patator	5897
	DoS slowloris	5796
	DoS Slowhttptest	5499
	Bot	3932
	Web Attack Brute Force	1507
	Web Attack XSS	652
	Infiltration	36
	Web Attack Sql Injection	21
	Heartbleed	11
Merge Data (Set B) 10 Class	BENIGN	2687419
	PortScan	317860
	DDoS	256054
	DoS [Hulk+GoldenEye+slowloris+Slowhttptest]	252661
	FTP-Patator	7938
	SSH-Patator	5897
	Bot	3932
	Web Attack [BruteForce+XSS+Sql Injection]	2180
	Infiltration	36
	Heartbleed	11
Delete Data (Set C) 12 Class	BENIGN	2687419
	PortScan	317860
	DDoS	256054
	DoS Hulk	231073
	DoS GoldenEye	10293
	FTP-Patator	7938
	SSH-Patator	5897
	DoS slowloris	5796
	DoS Slowhttptest	5499
	Bot	3932
	Web Attack Brute Force	1507
	Web Attack XSS	652
Merge&Delete Data (Set D) 8 Class	BENIGN	2687419
	PortScan	317860
	DDoS	256054
	DoS	252661
	FTP-Patator	7938
	SSH-Patator	5897
	Bot	3932
	Web Attack	2180

같이 각각의 희소 클래스에 초점을 맞추고 유사 공격 데이터 합치는 방식으로 데이터를 재구성했다. 희소 클래스와 준 희소 클래스를 기준으로 클래스 병합 및 제거를 통해 4개의 데이터 세트를 구성했다.

첫 번째, 병합 데이터(Set B)는 'DoS' 계열과 'Web Attack' 계열로 그룹화하여 총 10개의 클래스로 구성했다. 두 번째, 삭제 데이터(Set C)는 원본 샘플이 적어 정상적으로 분류되지 않고, 다른 클래스에 잡음으로 작용해서 분류 성능 저하의 원인으로 의심되는 3개의 희소 클래스(Infiltration, Hearbleed, Web Attack Sql Injection)를 제거한 데이터 세트로, 총 12개의 클래스로 구성했다. 세 번째, 병합&삭제 데이터(Set D)는 첫 번째와 두 번째 개념을 결합한 방법으로, 'DoS' 계열과 'Web Attack' 계열을 그룹화하고, 나머지 희소 클래스인 'Infiltration'과 'Heartbleed'를 제거하여 총 8개의 클래스로 구성했다.

3.2 데이터 전처리

본 연구에서는 UNB(University of New brunswick)에서 고안한 CSE-CIC-IDS 2018 데이터 세트를 사용했다[13]. CSE-CIC-IDS 2018 데이터 세트는 '금요일 오전, 오후', '월요일 오전, 오후', '목요일', '수요일'로 구성되었고 모든 요일의 공격 데이터는 같은 특성의 각기 다른 공격 유형으로 구성되었다.

Zhou[5]의 연구에서는 전처리 단계에서 데이터 분할(Train set&Test set), 결측값 제거, 데이터형 변환, 특징 선택(Feature Selection), 데이터 정규화(Normalization)의 과정을 수행했다. 본 연구에서도 이와 같은 구조를 따라 기계학습 연구를 수행하고자 했다.

1) 데이터 정제(Data Clean)

데이터 정제는 먼저, 학습 성능의 일반화 또는 기계학습에서 중요하게 다루는 과적합(Overfitting)을 방지하기 위해 수행했다. CSE-CIC-IDS 2018 데이터 세트는 적지 않은 데이터양을 가지고 있다고 판단하여, 충분한 테스트 셋을 주기 위해 훈련 세트와 테스트 세트를 7:3의 비율로 분할 했다. Null 값을 포함한 레코드는 삭제하고 inf 값은 데이터 세트 내의 최댓값으로 변경하였으며, 데이터형은 모두 float type으로 변환했다.

2) 특징 선택(Feature Selection)

특징 선택은 Importance와 Correlation을 분석한 후, Importance는 0.001 이하의 값을 가지는 Feature를 먼저 제거한다. 이후에 남은 Feature 중에서 상관관계가 0.95 이상인 Feature 짝을 찾아내어 이 중에 하나를 제거한다. Importance 측정에 기반한 Wrapper 방식은 실제 모델을 학습하여 Feature의 부분집합의 유용성을 측정하는 방식으로 계산 비용이 많이 들고 속도가 느리다는 단점이 있지만, 교차검증을 활용하여 Feature의 부분집합을 만들어 항상 최적의 부분집합을 선택할 수 있다는 장점이 있다. Corelation 분석에 기반한 Filter 방식은 종속변수와 상관관계에 의해 Feature의 관련성을 측정하는 방식으로 모델을 학습하지 않기 때문에 Wrapper 방식보다 속도가 빠르다는 장점이 있지

만, 통계 방법으로 Feature의 부분집합을 만들어 항상 최적의 Feature 부분집합을 선택하지 못한다. 따라서 본 연구에서는 Wrapper와 Filter의 2가지 방식을 순차적으로 적용하는 하이브리드 특징 선택 방법을 적용했다[14, 15].

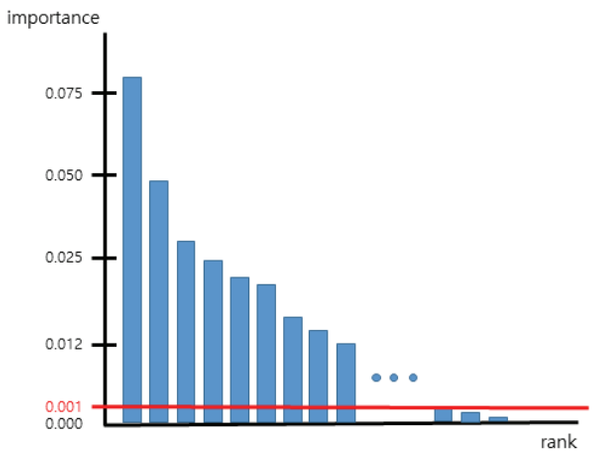


Fig. 1. Feature Importance Score using RandomForest for Set A, Set B, Set C and Set D

Fig. 1은 특징 선택을 위해 Wrapper 방식으로 RandomForest 모델을 활용해서 Feature의 중요도 점수를 산출하고 Feature Importance Score를 내림차순으로 정렬한 후, 빨간 실선으로 Threshold Value를 나타낸 그림이다. Threshold Value는 0.001로 설정했고, Threshold Value보다 큰 값의 Feature만을 선택하여 최적의 Feature 부분집합을 생성하게 된다.

	F1	F2	...	F77	F78
F1	1.00	0.56	...	0.96	0.12
F2	0.56	1.00	...	0.85	0.91
...	...	...	...	0.65	0.76
F77	0.96	0.85	0.65	1.00	0.95
F78	0.12	0.91	0.76	0.95	1.00

Fig. 2. Feature Correlation Score using Pearson for Set A, Set B, Set C and Set D

Fig. 2는 특징 선택을 위해 Filter 방식으로 Pearson 통계 방식을 사용하였고, Feature 간 상관관계를 분석하고 점수를 행렬로 나타낸 것이다. 두 Feature 상관관계 점수가 0.95 이상이면 두 Feature 중, 하나의 Feature만 존재해도 모델의 학습 성능에 영향을 주지 않는다고 판단했다. 이는 학습 효율 즉, 학습 시간을 개선할 수 있다고 판단하여 하나의 Feature만 선택했다.

3) 정규화(Normalization)

정규화(Normalization)는 데이터가 가진 Feature의 서로 다른 특징값의 단위 불일치 문제를 해결하기 위한 것이다. 이 문제는 기계학습을 사용하기 전, 데이터를 정규화 과정으로 해결해야 하는 문제이다. 본 연구에서 정규화 방법은 Equation (3)의 Quantile Scaler를 사용하여 해결했다[2].

4. 실 험

4.1 실험 환경

실험에 사용된 환경은 Table 2와 같다. Python 3.6.8 버전을 사용했고, tensorflow-gpu 1.14.0, keras 2.3.1, pandas 1.0.3, numpy 1.17.0, scikit-learn 0.22 버전을 사용해서 실험을 진행했다.

Table 2. Experimental Environment

Distinct	Hardware and Software Spec
OS	Windows 10 Pro
CPU	AMD Ryzen 7 3700X 8-Core Processor 3.59GHz
RAM	64GB
GPU	GeForce RTX 2070 SUPER 8GB
Language	Python 3.6.8
Library	tensorflow-gpu 1.14.0, keras 2.3.1, pandas 1.0.3, numpy 1.17.0, scikit-learn 0.22

4.2 실험 구조도

희소 클래스에 대한 분류 성능을 개선하기 위해 데이터 세트를 재구성하고 Fig. 3의 구조를 제안했다. Fig. 3은 데이터 구성, 전처리, 딥러닝 알고리즘을 사용한 분류 및 평가의 과

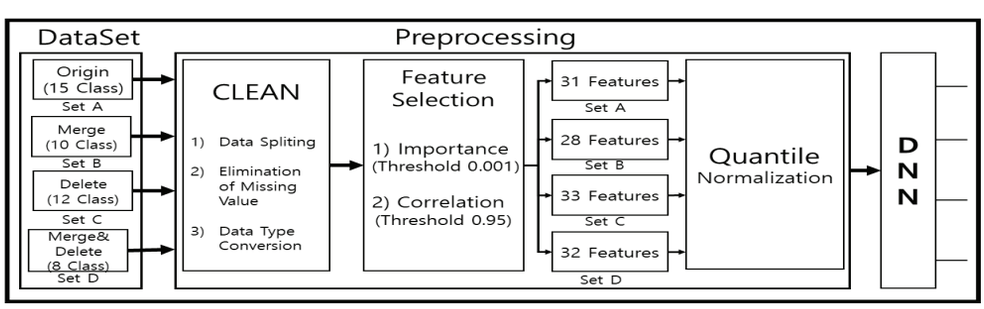


Fig. 3. Proposed Architecture for Solving Unbalanced Data Problems with Rare Classes

정을 설명한다. 원본 데이터 세트(Set A)와 재구성 데이터 세트(Set B, Set C, Set D)를 각각 'Preprocessing'의 데이터 CLEAN 과정에서 데이터 분할과 결측값 제거, 그리고 데이터 타입변환을 수행한다. 'Feature Selection' 과정에서는 10개의 추정기를 가진 RandomForest 알고리즘으로 산출한 중요도가 0.001 이하인 특징은 제거하고, 특징 쌍의 상관관계가 0.95 이상이면 하나의 특징을 제거하여 Feature Selection을 수행한다. Set A는 31 Features, Set B는 28 Features, Set C는 33 Features, Set D는 32 Feature를 얻게 된다. 새로운 Feature로 재구성된 데이터는 마지막 전처리 과정인 Quantile Normalization으로 데이터 정규화를 수행한 후, Table 3에 명시된 DNN(Deep Neural Network) Classifier를 통해 4개의 Subset에 대해 학습 및 분류를 수행한다.

Table 3. Hyper Parameter of DNN Classifier

	Hyper Parameter
DNN Classifier	Adam optimizer epoch 100 batch size 100 validation_split 0.2 Hidden Layer [1000, 500, 100]

4.3 성능지표

실험 결과의 성능지표는 TP(True Positive)를 사용한다. TP는 혼동행렬(Confusion Matrix)을 통해 알 수 있으며, 이는 단순한 정확도 측정보다 실제 정답을 정답으로 정확히 판별했는지, 각 데이터의 유형별로 알 수 있는 지표다. 해당 지표를 통해 앞서 구성한 4가지 데이터 세트의 희소 클래스 및 준 희소 클래스의 정확한 분류 성능을 측정하고 TP 개수에 따른 증감율을 백분율로 계산하여 원본 데이터에 대해 TP 개수가 얼마나 달라졌는지 나타냈다. 증가(▲)는 성능이 원본 데이터와 비교해 개선되었음을 나타내고, 감소(▼)는 원본 데이터와 비교해 저하되었음을 나타낸다.

4.4 실험 결과

Table 4는 Set A와 Set B의 성능을 Web Attack 계열과 DoS 계열을 포함한 모든 클래스에 대한 결과를 혼동행렬에서의 TP(True Positive) 증감율로 비교했다. BENIGN의 TP는 805020개에서 805605개로 0.07% 증가(▲), Bot은 591개에서 762개로 28.9% 증가(▲), DDoS는 76523개에서 76780개로 증가(▲), DoS 계열의 TP의 전체 합은 75,540개에서 75,522로 약 0.02%로 감소(▼), FTP-Patator는 2366개에서 2387개로 0.88% 증가(▲), Heartbleed는 4개에서 3개로 25% 감소(▼), Infiltration은 8개에서 9개로 12.5% 증가(▲), PortScan은 995585개에서 95090개로 0.51% 감소(▼), SSH-Patator는 1755개에서 1767개로 0.45% 감소(▼) Web Attack 계열은 410개에서 624개로 약 52% 증가(▲)했다. 재구성 데이터 Set B는 희소 클래스가 포함된 데이터 유형의 전체적인 TP 비율이 원본 데이터와 비교했을 때, 0.06% 증가(▲)했으므로 성능이 개선되었다고 할 수 있다.

Table 4. Comparison of Rare Class TP performance of Origin Data (Set A) and Merge Data (Set B)

Set A(Class)	TP	Set B(Class)	TP	Rate
BENIGN	805020	BENIGN	805605	▲0.07%
Bot	591	Bot	762	▲28.9%
DDoS	76523	DDoS	76780	▲0.33%
DoS GoldenEye	3067	DoS	75522	▼0.02%
DoS Hulk	69261			
DoS Slowhttptest	1532			
DoS slowloris	1680			
DoS Total	75540			
FTP-Patator	2366	FTP-Patator	2387	▲0.88%
Heartbleed	4	Heartbleed	3	▼25%
Infiltration	8	Infiltration	9	▲12.5%
PortScan	95585	PortScan	95090	▼0.51%
SSH-Patator	1775	SSH-Patator	1767	▼0.45%
Web Attack Brute Force	409	Web Attack	624	▲52%
Web Attack Sql	1			
Web Attack XSS	0			
Web Attack Total	410			
Total	1057822	Total	1058549	▲0.06%

Table 5는 Set A와 Set C의 성능을 비교한 내용이다. Set C는 Set A에서 잡음이라고 판단할 수 있을 만큼 적은 데이터 샘플을 가진 희소 클래스인 Heart bleed, Infiltration, Web Attack Sql Injection을 삭제해서 구성한 데이터 세트로 잡음을 제거했을 때, 준 희소 클래스라고 정의한 클래스들의 성능을 TP 증감율로 비교했다. Bot은 591개에서 836개으로 약 41% 증가(▲), DoS Slowhttptest는 1,532개에서 1,610개로 약 5% 증가(▲), DoS slowloris는 1,680개에서 1,713개로 약 1% 증가(▲), FTP-Patator는 1,532개에서 2,342개로 약 52% 증가(▲), SSH-Patator는 1,775개에서 1,789개로 약 0.7% 증가(▲), Web Attack Brute Force는 409개에서 439개로 약 7% 증가했고(▲), Web Attack XSS는 0개에서 3개로 증가(▲)했다.

Table 6은 Set A와 Set D의 성능을 비교한 내용이다. Set D는 단순히 Set B와 Set C의 아이디어를 결합해서 DoS 계열과 Web Attack 계열로 그룹화를 하고, 잡음으로 판단되는 클래스는 삭제한 데이터 세트이다. Bot은 591개에서 431개로 약 27% 감소(▼), DoS는 75,540개에서 75,268개로 약 0.3% 감소(▼), FTP-Patator는 1,532개에서 2,355개로 약

Table 5. Comparison of Rare Class TP Performance of Origin Data (Set A) and Delete Data (Set C)

Set A(Class)	TP	Set C(Class)	TP	Rate
Bot	591	Bot	836	▲41%
DoS Slowhttptest	1532	DoS Shttptest	1601	▲5%
DoS slowloris	1680	DoS slowloris	1713	▲1%
FTP-Patator	1532	FTP-Patator	2342	▲52%
SSH-Patator	1775	SSH-Patator	1789	▲0.7%
Web Attack Brute Force	409	Web Attack Brute Force	439	▲7%
Web Attack XSS	0	Web Attack XSS	3	▲

53%로 증가(▲), SSH-Patator는 2% 감소(▼), Web Attack 만 410개에서 630개로 약 53% 증가(▲)했다.

Table 6. Comparison of Rare Class TP Performance of Origin Data (Set A) and Merge & Delete Data (Set D)

Set A(Class)	TP	Set D(Class)	TP	Rate
Bot	591	Bot	431	▼27%
DoS GoldenEye	3067	DoS	75268	▼0.3%
DoS Hulk	69261			
DoS Slowhttptest	1532			
DoS slowloris	1680			
Total	75540			
FTP-Patator	1532	FTP-Patator	2355	▲53%
SSH-Patator	1775	SSH-Patator	1732	▼2.4%
Web Attack Brute Force	409	Web Attack	630	▲53%
Web Attack Sql	1			
Web Attack XSS	0			

4.5 상세 실험 결과(혼동행렬)

Fig. 4~7은 Set A, Set B, Set C, Set D를 DNN Classifier를 통해 분류한 실험 결과이다. 성능이 우수함을 TP로 평가했지만 앞서 혼동 행렬(Confusion Matrix)을 통해 False Negative, False Positive의 값을 통해 각 데이터 셋의 성능을 확인할 수 있었다.

5. 결 론

이 연구를 통해 많은 양의 데이터와 데이터의 특징 정보로부터 불필요한 정보를 제거하고 유의미한 정보만을 가지고 학습하여 희소 클래스에 대한 분류 성능을 개선하고자 했다.

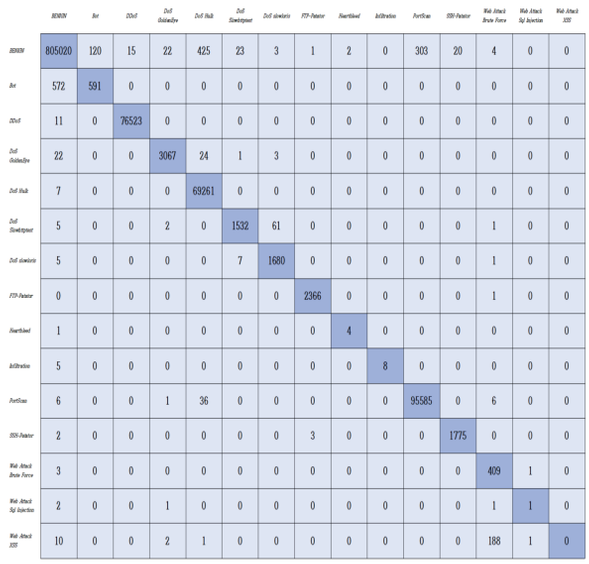


Fig. 4. Confusion Matrix of Set A

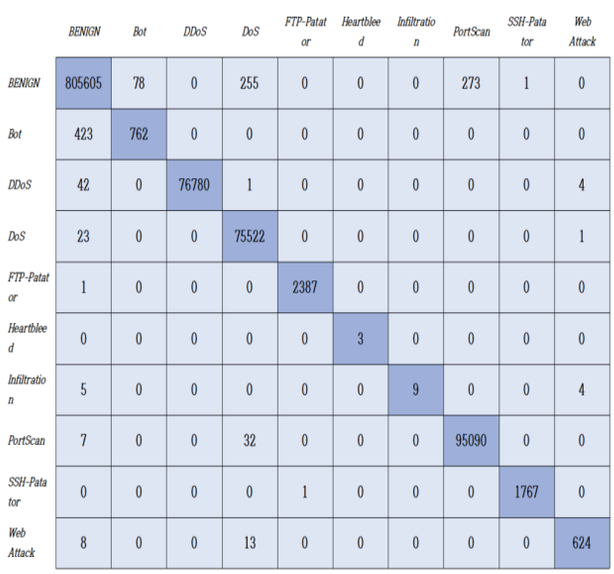
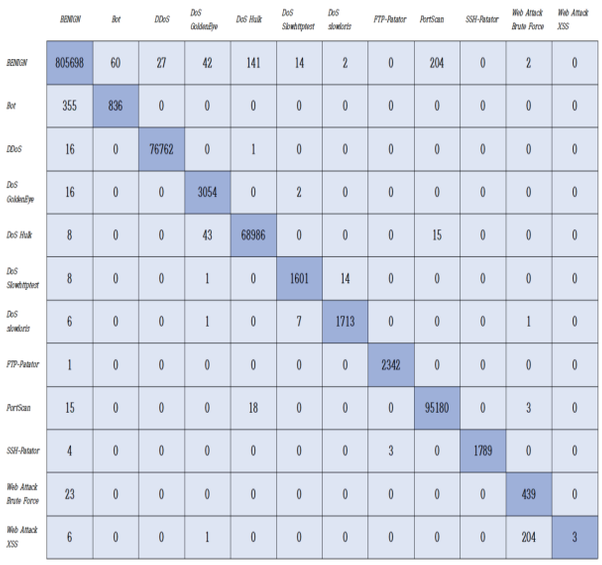


Fig. 5. Confusion Matrix of Set B

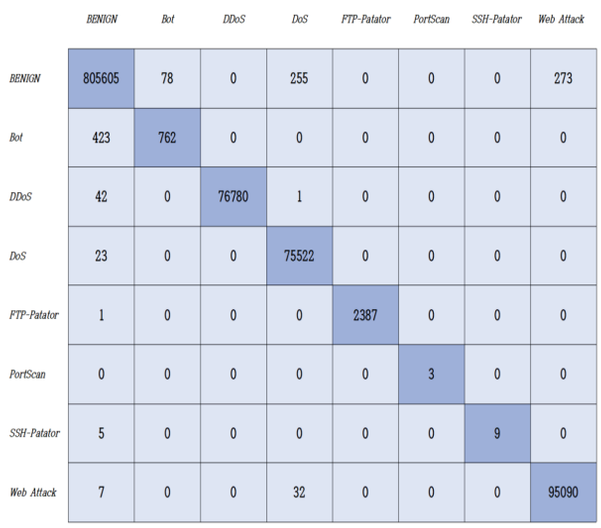
중요도와 상관관계를 통해 유의미한 특징을 선택하는 전처리 과정을 통해 데이터를 정제하고 이를 통해 분류 성능을 높이는 결과를 얻었다. 실험에서는 유사한 데이터 분류에 가장 널리 사용되는 딥러닝 모델인 DNN Classifier를 구현하여 새로 구성한 3가지 데이터 세트의 각 공격 유형별 분류를 수행했다. 그 결과, 3개의 데이터 세트 모두 개선된 모습을 보였고 특히, 잡음으로 분류될 가능성이 있는 희소 클래스를 제거하고 준 희소(Semi Rare) 클래스로 정의한 클래스의 성능을 개선하기 위해 구성한 데이터인 Delete Set(Set C)의 분류 성능이 가장 눈에 띄게 개선되었다.

하지만, 이들을 독립적인 데이터 셋으로 사용하기에는 삭제된 클래스도 있기에 적합하지 않다. 따라서, 향후 연구로



	BENIGN	Bot	DDoS	DoS GoldenEye	DoS Hulk	DoS Slowhttptest	DoS slowloris	FTP-Patator	PortScan	SSH-Patator	Web Attack Brute Force	Web Attack XSS
BENIGN	805698	60	27	42	141	14	2	0	204	0	2	0
Bot	355	836	0	0	0	0	0	0	0	0	0	0
DDoS	16	0	76762	0	1	0	0	0	0	0	0	0
DoS GoldenEye	16	0	0	3054	0	2	0	0	0	0	0	0
DoS Hulk	8	0	0	43	68986	0	0	0	15	0	0	0
DoS Slowhttptest	8	0	0	1	0	1601	14	0	0	0	0	0
DoS slowloris	6	0	0	1	0	7	1713	0	0	0	1	0
FTP-Patator	1	0	0	0	0	0	0	2342	0	0	0	0
PortScan	15	0	0	0	18	0	0	0	95180	0	3	0
SSH-Patator	4	0	0	0	0	0	0	3	0	1789	0	0
Web Attack Brute Force	23	0	0	0	0	0	0	0	0	0	439	0
Web Attack XSS	6	0	0	1	0	0	0	0	0	0	204	3

Fig. 6. Confusion Matrix of Set C



	BENIGN	Bot	DDoS	DoS	FTP-Patator	PortScan	SSH-Patator	Web Attack
BENIGN	805605	78	0	255	0	0	0	273
Bot	423	762	0	0	0	0	0	0
DDoS	42	0	76780	1	0	0	0	0
DoS	23	0	0	75522	0	0	0	0
FTP-Patator	1	0	0	0	2387	0	0	0
PortScan	0	0	0	0	0	3	0	0
SSH-Patator	5	0	0	0	0	0	9	0
Web Attack	7	0	0	32	0	0	0	95090

Fig. 7. Confusion Matrix of Set D

각 데이터 셋의 훈련을 통해 얻은 모델의 가중치를 이용하는 방법으로 개선된 모델 구현이 가능할 것이라 기대된다.

References

[1] V. Kanimozhi, and T. P. Jacob, "Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," In: *2019 International Conference on Communication and Signal Processing (ICCSP)*. IEEE, pp.0033-0036, 2019.

[2] J. H. Seo, "A comparative study on the classification of the imbalanced intrusion detection dataset based on deep learning," *Journal of Korean Institute of Intelligent Systems*, Vol.28, No.2, pp.152-159, 2018.

[3] M. H. Abdulraheem and N. B. Ibraheem, "A detailed analysis of new intrusion detection dataset," *Journal of Theoretical and Applied Information Technology*, Vol.97, No.17, 2019.

[4] B. Alsughayyir, A. M. Qamar, and R. Khan, "Developing a Network Attack Detection System Using Deep Learning," In: *2019 International Conference on Computer and Information Sciences (ICCIS)*. IEEE. pp.1-5, 2019.

[5] Q. Zhou and D. Pezaros, "Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection--An Analysis on CIC-AWS-2018 dataset," arXiv preprint arXiv: 1905.03685, 2019.

[6] B. K. Singh, K. Verma, and A. S. Thoke, "Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification," *International Journal of Computer Applications*, Vol.116, No.19, 2015.

[7] Z. Liu and W. Li, "A method of SVM with normalization in intrusion detection," *Procedia Environmental Sciences*, Vol.11, pp.256-262, 2011.

[8] scikit-learn.org [Internet], https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[9] S. C. Hicks and R. A. Irizarry, "When to use quantile normalization?," BioRxiv, 2014.

[10] scikit-learn.org [Internet], https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.quantile_transform.html

[11] P. Devan and N. Khare, "An efficient XGBoost-DNN-based classification model for network intrusion detection system," *Neural Computing and Applications*, 1-16, 2020.

[12] N. Qazi and K. Raza, "Effect of feature selection, SMOTE and under sampling on class imbalance classification," In: *2012 UKSim 14th International Conference on Computer Modelling and Simulation*. IEEE. pp.145-150, 2012.

[13] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," In: ICISSP. pp.108-116, 2018.

[14] J. M. Cadenas, M. C. Garrido, and R. MartiNez, "Feature subset selection filter–wrapper based on low quality data," *Expert Systems with Applications*, Vol.40, No.16, pp.6241-6252, 2013.

[15] H. Min and Wu. Fangfang, "Filter-wrapper hybrid method on feature selection," In: *2010 Second WRI Global Congress on Intelligent Systems*. IEEE. pp.98-101, 2010.

류 경 준

https://orcid.org/0000-0002-0714-2779
e-mail : rkj6663@sju.ac.kr
2019년 ~ 현 재 세종대학교 컴퓨터공학과 석사과정
관심분야 : 정보보안, 기계학습, 데이터 마이닝

신 동 일

https://orcid.org/0000-0002-8621-715X
e-mail : dshin@sejong.ac.kr
1988년 연세대학교 컴퓨터과학과(학사)
1993년 Washington State University 컴퓨터과학과(석사)
1997년 North Texas University 컴퓨터과학과(박사)
1998년 ~ 현 재 세종대학교 컴퓨터공학과 교수
관심분야 : 정보보안, 기계학습, 데이터 마이닝, 생체신호 데이터처리

신 동 규

https://orcid.org/0000-0002-2665-3339
e-mail : shindk@sejong.ac.kr
1986년 서울대학교 계산통계학과(학사)
1992년 Illinois Institute of Technology 컴퓨터과학과(석사)
1997년 Texas A&M University 컴퓨터과학과(박사)
1998년 ~ 현 재 세종대학교 컴퓨터공학과 교수
관심분야 : 정보보안, 기계학습, 유비쿼터스 컴퓨팅, 생체신호 데이터처리

박 정 찬

https://orcid.org/0000-0001-6192-0685
e-mail : jcpark@add.re.kr
1996년 광운대학교 컴퓨터공학과(석사)
2015년 고려대학교 사이버국방학과 (박사수료)
1996년 ~ 현 재 국방과학연구소 책임연구원
관심분야 : 디지털 포렌식, 정보보안

김 진 국

https://orcid.org/0000-0001-8508-5784
e-mail : jingoog78@gmail.com
2006년 KAIST 전기전자공학과(석사)
2010년 KAIST 전기전자공학과(박사)
2010년~2011년 SKB/SKT 네트워크 부문 매니저
2011년 ~ 현 재 국방과학연구소 선임연구원
관심분야 : 무선 통신, 임베디드 시스템 보안