희소 클래스 분류 문제 해결을 위한 전처리 연구

류경준*, 신동규**, 신동일*
*세종대학교 컴퓨터공학과
**세종대학교 컴퓨터공학과
rkj6663@sju.ac.kr, shindk@sejong.ac.kr, dshin@sejong.ac.kr

A Study on Pre-processing for the Classification of Rare Classes

Kyungjoon Ryu*, Dongkyoo Shin**, Dongil Shin*
*Dept. of Computer Engineering, Sejong University
**Dept. of Computer Engineering, Sejong University

요 약

실생활의 사례를 바탕으로 생성된 여러 분야의 데이터셋을 기계학습(Machine Learning) 문제에 적용하고 있다. 정보보안 분야에서도 사이버 공간에서의 공격 트래픽 데이터를 기계학습으로 분석하는 많은 연구들이 진행 되어 왔다. 본 논문에서는 공격 데이터를 유형별로 정확히 분류할 때, 실생활 데이터에서 흔하게 발생하는 데이터 불균형 문제로 인한 분류 성능 저하에 대한 해결방안을 연구했다. 희소 클래스 관점에서 데이터를 재구성하고 기계학습에 악영향을 끼치는 특징들을 제거하고 DNN(Deep Neural Network) 모델을 사용해 분류 성능을 평가했다.

Machine-Learning, Deep-Learning, Rare Class, Feature Selection, Data Reconstruction

1. 서론

최근 실생활에서 나타나는 사례로 생성된 데이터를 기계학습(Machine Learning)으로 해결하기 위한 연구들이 등장하고 있다. 현재 IT산업에서는 4차 산업혁명 시대의 중요 자산으로써 데이터의 의미는 가중되고 있으며, 사회에 각 기관들과 세계의 여러 국가들은 정보보호, 정보수집과 정보 활용에 대한 연구를 진행하고 있다. 실생활 데이터를 이용한 기계학습에 있어서 가장 큰 문제점은 데이터 불균형(Data Imbalance)이다.

데이터 불균형 문제는 한 데이터 세트 내에서 유형별 샘플 수가 균형 잡혀있지 않는 것을 말한다. 이런 데이터 내에서 샘플의 수가 적은 유형을 가리켜 희소 클래스라고 한다. 이러한 문제가 생기는 이유는 실생활에서 정상적인 상황보다 비정상적인 상황이 훨씬 적게 발생하기 때문이다. 이를 해결하기 위해 인공적으로 데이터를 생성할 수도 있지만, 그 역시도 물리적인 한계에 부딪히는 경우가 많다[1, 2].

본 논문에서는 네트워크 정상 트래픽과 비정상 트래픽으로 구성된 데이터셋의 희소 클래스(Rare Class)에 해당하는 공격 트래픽 데이터의 분류 성능을 높이기 위한 연구에 초점을 두고, 실험을 진행했다.

2. 관련 연구

M.H.ABDULAHEEM[2]의 연구에서는 신경망 알고리즘을 통해 분류하기 전에 데이터셋에서 학습의 효율이나 성능을 저해하는 특징을 중요도(Importance)와 상관관계(Correlation)를 통해 분석하고 데이터의 분류 성능을 크게 개선했다. 학습에 있어서 가장 중요한 작업은 정규화(Normalization)이다. 정규화는 학습에 유의미한 결과를 가져다주는 효과도 있지만, 분류기 성능에도 영향을 미친다[3]. 정규화 방법에는 여러 방법이 있다. 대표적인 방법으로 'Min-Max Scaler', 'Standard Scaler'와 'Quantile Transform'이다. M.H.ABDULAHEEM[2]은 대표적인 3가지 정규화 방법에 대해서 실험을 진행했다. 3가지 정규화 방법 중 분류기 성능이 가장 우수하게 나타나는 방법은 'Quantile Transform'이라고 실험을 통해 증명했다.

3. 본론

3.1 데이터셋 재구성

CSE-CIC-IDS 2018 데이터셋의 유형별 구성은 가장 많은 정상 데이터(BENIGN)가 전체 데이터셋의 80%이고, 가장 적은 공격 데이터(Heartbleed)는 0.0004%로 데이터의 불균형이 문제를 가진 데이터

셋이다[4]. 이처럼 데이터의 불균형이 심각한 경우, 샘플이 적은 클래스의 데이터들이 샘플 수가 많은 클래스의 데이터에 억눌려서 데이터가 제대로 학습하지 못하는 경우가 발생한다. 본 논문에서는 1,500개 미만의 샘플을 가진 클래스의 데이터가 희소 클래스(Rare Class)라고 정의하고[2], 10,000개 미만의 샘플을 준희소 클래스(Semi-Rare Class)로 구분하고 해당 클래스에 대한 분류 성능을 개선하기 위해 <표 1>과 같이 각각의 희소 클래스에 초점을 맞추고 유사 공격데이터 합치는 방식으로 재구성하였다.

<표 1> 재구성한 데이터셋의 클래스와 샘플 개수

구분	클래스 이름	개수
원본 데이터 (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
병합 데이터 (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
삭제 데이터 (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
병합&삭제 데이터 (Set D) 8 Class	BENIGN	2687419
	PortScan	317860
	DDoS	256054
	DoS	252661
	FTP-Patator	7938
	SSH-Patator	5897
	Bot	3932
	Web Attack	2180

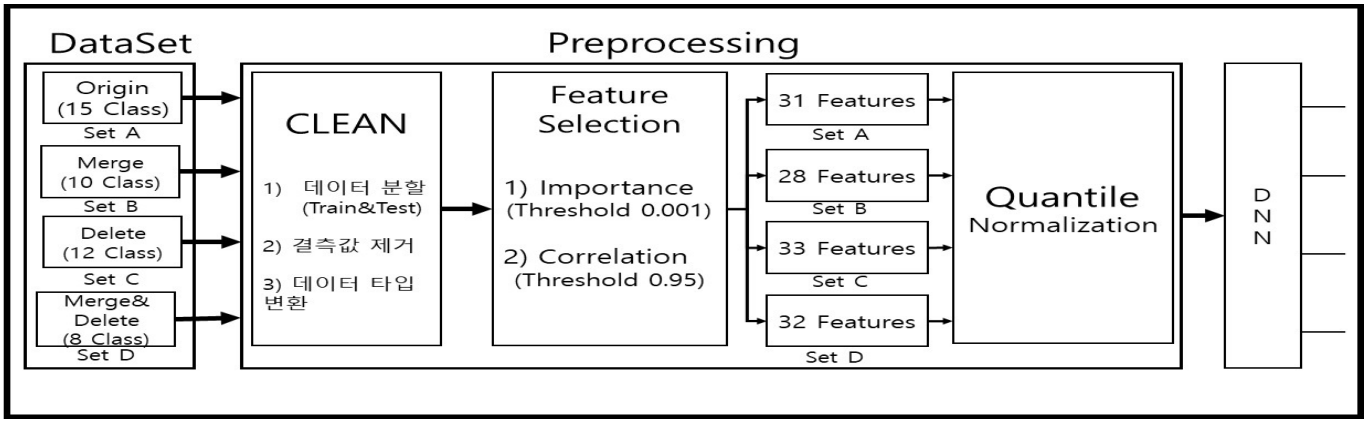
정상 데이터(BENIGN)를 제외한 공격 데이터에 대해서 희소 클래스를 기준으로 그룹화와 제거를 통해 4개의 데이터셋을 구성하였다. 병합 데이터(Set B)는 'DoS'계열과 'Web Attack'계열로 그룹화하고 'FTP-Patator'와 'SSH-Patator' 클래스는 희소 클래스로 정의하지 않아 그대로 두어서 총 10개의 클래스로 구성되었다. 삭제 데이터(Set C)는 원본 샘플의 개수가 작아 오히려 희소 클래스로 분류되지 않는 데이터에 잡음으로 들어가 잘못된 정확도가 될 것이라고 판단한 3개의 희소 클래스(Infiltration, Hearbleed, Web Attack Sql Injection)를 제거해서 준희소 클래스(Semi-Rare Class) 총 12개의 클래스로 구성되었고, 병합&삭제 데이터(Set D)는 앞에서 말한 2가지 방법을 결합하여 먼저 'DoS'계열과 'Web Attack' 계열을 그룹화하고 나머지 희소 클래스인 'Infiltration'과 'Heartbleed'를 제거해서 총 8개의 클래스로 구성했다.

3.2 데이터 전처리

본 논문에서 사용한 CSE-CIC-IDS 2018 데이터셋[5]은 '금요일 오전, 오후', '월요일 오전, 오후', '목요일', '수요일'로 구성되어있고 동일한 특성과 각기 다른 공격 유형에 대한 정보를 가지고 있다. Q.Zhou[4]의 연구에서는 데이터셋을 기계학습을 위한 전처리 단계에서 데이터 분할(Train set&Test set), 결측값 제거, 데이터형 변환, 특징선택(Feature Selection), 데이터 정규화(Normalization)를 수행한다. 데이터 분할은 Train set과 Test set을 7:3의 비율로 나눈다. 학습에 영향을 끼친 데이터를 테스트에 사용하면 정확도 성능은 오르지만, 제대로된 평가가 되지 않는 과적합(Overfitting) 문제로 이어질 수 있다. 결측값 제거에서는 Null 값을 포함한 레코드는 삭제하고 inf 값은 최대값으로 채워주며, 데이터형은 모두 float type으로 변환한다. 그 후에 특징선택은 Importance와 Correlation을 0-1사이의 값으로 표현하고, Importance는 0.001이하의 값을 가지는 Feature와 상관관계가 0.95이상인 Feature를 제거한다. Importance는 앙상블 방법인 RandomForest 알고리즘으로 분석한 후, 0.001이하의 값을 갖는 특징은 학습 성능(정확도)을 저하시키고 효율(시간)을 감소시킨다고 판단하여 제거하였고, Correlation은 두 개의 특징의 상관관계가 0.95이상일 경우, 한 특징만 있어도 학습 성능에 있어서 무방하다고 판단하고 학습 효율 향상을 위해 하나의 특징은 제거하였다. 마지막으로 Quantile Normalization을 적용하면 학습 모델에 들어가기까지의 전처리 과정이 끝난다.

4. 실험

희소 클래스에 대한 분류 성능을 개선하기 위해 데이터셋을 재구성하고 (그림 1)의 구조를 제안하였



(그림 1) 제안된 구조

다. 원본 데이터(Set A)과 재구성 데이터셋(Set B, Set C, Set D)은 각각 전처리 과정을 수행한다. 먼저 데이터 CLEAN 과정에서 데이터 분할과 결측값 제거, 그리고 데이터 타입변환을 수행하고 Feature Selection과정을 수행한다. 먼저, 10개의 추정기를 가진 RandomForest알고리즘으로 Threshold Value가 0.001이하인 특징은 제거하고, 특징 간에 상관관계를 따졌을 때, Threshold Value가 0.95이상인 경우에 제거하는 방식으로 각각 Feature Selection을 수행하며, 그 결과 Set A는 31 Features, Set B는 28 Features, Set C는 33 Features, Set D는 32 Feature를 얻는다. 새로운 Feature 구성으로 재구성된 데이터는 마지막 전처리 과정인 Quantile Normalization을 수행한 후에, <표 2>와 같이 구성된 DNN(Deep Neural Network) Classifier를 통해 각각 분류된다.

<표 2> DNN Classifier의 Hyper Parameter

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

<표 3>은 Set A와 Set B의 성능을 Web Attack 계열과 DoS계열의 혼동행렬에서의 TP(True Positive) 증감율로 비교하였다. DoS 계열의 TP의 전체 합은 75,540개에서 75,522로 약 0.0002% 감소했지만, Web Attack계열은 410개에서 624개로 약 0.52% 증가했다. 재구성 데이터 Set B는 희소 클래스에 대해 좋은 성능을 보였다.

<표 3> 원본 데이터(Set A)와 병합 데이터(Set B)의 그룹화 클래스 TP 성능 비교

Set A(Class)	TP	Set B(Class)	TP
DoS GoldenEye	3067	DoS	75522
DoS Hulk	69261		
DoS Shttptest	1532		
DoS slowloris	1680		
Total	75540		
Web Attack Brute Force	409	Web Attack	624
Web Attack Sql	1		
Web Attack XSS	0		
Total	410		

<표 4>는 Set A와 Set C의 성능을 비교한 내용이다. Set C는 Set A에서 잡음이라고 판단할 수 있을만큼 적은 데이터 샘플을 가진 희소 클래스인 Heart bleed, Infiltration, Web Attack Sql Injection를 삭제해서 구성한 데이터셋으로 잡음을 제거했을 때, 준희소 클래스라고 정의한 클래스들의 성능을 TP 증감율로 비교하였다. Bot은 591개에서 836개으로 약 0.41% 증가, DoS Slowhttptest는 1532개에서 1610개로 약 0.05% 증가, DoS slowloris는 1680개에서 1713개로 약 0.01% 증가, FTP-Patator는 1532개에서 2342개로 약 0.52% 증가, SSH-Patator는 1775개에서 1789개로 약 0.007%증가, Web Attack Brute Force는 409개에서 439개로 약 0.07%증가했고, Web Attack XSS는 0개에서 3개로 증가했다.

<표 4> 원본 데이터(Set A)와 삭제 데이터(Set C)의 준(Semi) 희소 클래스 TP 성능 비교

Set A(Class)	TP	Set C(Class)	TP
Bot	591	Bot	836
DoS Shttptest	1532	DoS Shttp	1601
DoS slowloris	1680	DoS slow	1713
FTP-Patator	1532	FTP-Patator	2342
SSH-Patator	1775	SSH-Patator	1789
Web Attack Brute Force	409	Web Attack Brute Force	439
Web Attack XSS	0	Web Attack XSS	3

<표 5>는 Set A와 Set D의 성능을 비교한 내용이다. Set D는 단순히 Set B와 Set C의 아이디어를 결합해서 DoS계열과 Web Attack계열로 그룹화를 하고, 잡음으로 판단되는 클래스는 삭제한 데이터셋이다. Bot은 591개에서 431개로 약 0.27%감소, DoS는 75540개에서 75268개로 약 0.003% 감소, FTP-Patator는 1532개에서 2355개로 약 0.53%로 증가, SSH-Patator는 0.02%감소, Web Attack만 410개에서 630개로 약 0.53% 증가했다.

<표 5> 원본 데이터(Set A)와 병합&삭제 데이터(Set D)의 희소 클래스 TP 성능 비교

Set A(Class)	TP	Set D(Class)	TP
Bot	591	Bot	431
DoS GoldenEye	3067	DoS	75268
DoS Hulk	69261		
DoS Shttptest	1532		
DoS slowloris	1680		
Total	75540		
FTP-Patator	1532	FTP-Patator	2355
SSH-Patator	1775	SSH-Patator	1732
Web Attack Brute Force	409	Web Attack	630
Web Attack Sql	1		
Web Attack XSS	0		

5. 결론

많은 정보를 포함하고 있는 네트워크 트래픽으로부터 불필요한 정보를 제거하고 유의미한 정보만을 가지고 학습하여 희소 클래스에 대한 분류 성능을 개선했다. 본 논문에서 제안했듯이, 특성 중요도와 특성 상관관계를 고려하여, 특징을 선택하고 희소 클래스에 대한 분류에 있어서 일반적인 기계학습 모델인 DNN Classifier로 원본 데이터셋과 분류 결과를 비교하였다.

향후, 우리가 살아가는 실생활 속 데이터의 불균형이 심한 경우에 본 논문의 연구를 인용하여 여러 도메인 환경에서 발생하는 데이터만으로도 더 좋은 성능을 끌어낼 수 있을 것이다.

Acknowledgement

본 연구는 방위사업청과 국방과학연구소의 지원으로 수행되었습니다 (UD190016ED).

참고문헌

[1] 서재현. (2018). 딥러닝 기반 불균형 침입탐지 데이터 분류에 관한 비교 연구. 한국지능시스템학회 논문지, 28(2), 152-159.

[2] ABDULRAHEEM, MOHAMMED HAMID; IBRAHEEM, NAJLA BADIE. A DETAILED ANALYSIS OF NEW INTRUSION DETECTION DATASET. Journal of Theoretical and Applied Information Technology, 2019, 97.17.

[3] SINGH, Bikesh Kumar; VERMA, Kesari; THOKE, A. S. Investigations on impact of feature normalization techniques on classifier's performance in breast tumor classification. International Journal of Computer Applications, 2015, 116.19.

[4] ZHOU, Qianru; PEZAROS, Dimitrios. Evaluation of Machine Learning Classifiers for Zero-Day Intrusion Detection--An Analysis on CIC-AWS-2018 dataset. arXiv preprint arXiv:1905.03685, 2019.

[5] SHARAFALDIN, Iman; LASHKARI, Arash Habibi; GHORBANI, Ali A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: ICISSP. 2018. p. 108-116.