JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 3, pp. 583-591, Mar. 2023

Check for updates

효율적인 네트워크 이상 탐지를 위한 차원축소 및 오버샘플링 방법 비교

윤 지 은[1] · 김 강 석[2*]
[1]아주대학교 지식정보공학과 석사과정
[2*]아주대학교 사이버보안학과 교수

Comparison of Dimensional Reduction and Oversampling Methods for Efficient Network Anomaly Detection

Ji-Eun Yoon[1] · Kangseok Kim[2*]

[1]Master's Course, Department of Knowledge Information Engineering, Ajou University, Suwon 16499, Korea
[2*]Professor, Department of Cyber Security, Ajou University, Suwon 16499, Korea

[요 약]

본 논문에서는 효율적인 네트워크 이상 탐지를 위한 차원 축소 및 오버 샘플링 방법을 비교한다. 다양한 분류 알고리즘들에 오버 샘플링과 차원 축소가 어떠한 영향을 미치는지 분석한다. 오버 샘플링의 평가는 분류 성능 평가 지표들을 사용하여야 하며, 차원 축소 영향의 평가는 단위 샘플 당 처리 속도를 지표로 사용한다. 실험 결과 차원 축소로 가장 눈에 띄게 처리 시간이 줄어든 모델은 KNN과 SVM이었다. 하지만 2차원으로 축소했을 땐 오히려 증가하였다. 오버 샘플링을 적용했을 땐 소수 클래스인 U2R과 R2L의 재현율과 F1 점수가 전반적으로 상승하여, 오버 샘플링은 소수의 공격 클래스 탐지에 유의미한 영향을 준다는 것을 확인하였다.

[Abstract]

In this paper, dimensionality reduction and oversampling methods are compared for efficient network anomaly detection, analyzing the effects of oversampling and dimensionality reduction on various classification algorithms. Oversampling was evaluated using classification performance evaluation indicators, and the dimensionality reduction effect was evaluated using the processing speed per unit sample as an indicator. As a result of the experiments, the models benefiting the most from dimensionality reduction were KNN and SVM displaying a significant reduction in processing time. However, when dimensionality was reduced to two dimensions, processing time increased. When oversampling was applied, it was confirmed that the recall and F1 scores of the minority classes U2R and R2L increased overall, confirming that oversampling had a significant effect on the detection of a minority class attack.

색인어 : 네트워크 이상 탐지, 기계 학습, 딥 러닝, 오버 샘플링, 차원 축소

Keyword : Network anomaly detection, Machine learning, Deep learning, Oversampling, Dimensionality reduction

http://dx.doi.org/10.9728/dcs.2023.24.3.583

Received 28 December 2022; **Revised** 03 February 2023
Accepted 07 February 2023

***Corresponding Author; Kangseok Kim**

Tel: +82-31-219-2496
E-mail: kangskim@ajou.ac.kr

Ⅰ. 서 론

컴퓨터 및 인터넷 기술의 발달로 네트워크에서 유통되는 정보의 양이 증가함에 따라 이를 대상으로 하는 공격 혹은 비정상 트래픽 역시 증가하고 있다. 특히 모바일 컴퓨팅이나 클라우드 컴퓨팅 기술 등의 발달로 네트워크에 저장되는 개인정보의 양이 많아지고 그 종류 또한 다양해지면서 이를 탈취하려는 공격도 증가하고 있다. 따라서 이러한 공격으로부터 정보를 보호하기 위한 네트워크 보안 기술이 중요해지고 있다[1].

네트워크 보안을 위한 방법으로는 침입 방지 시스템(IPS; Intrusion Prevention System), 망 분리, DMZ, NAT 등이 상용화되어 있다. 이 외에도 인공지능을 활용한 정보 보안 시스템의 상용화도 지속적으로 개발 중이다. 이러한 시스템 중 하나로 침입 탐지 시스템(IDS; Intrusion Detection System)이 있다. 침입 탐지 시스템은 다루는 대상에 따라 호스트 침입 탐지 시스템(HIDS; Host Intrusion Detection System)과 네트워크 침입 탐지 시스템(NIDS; Network Intrusion Detection System)으로 나뉜다. 네트워크 침입 탐지 방법 중에는 전문가가 공격 패턴을 파악하여 그에 해당하는 패턴을 가진 패킷의 접근을 차단하는 시그니처 탐지(Signature Detecion) 혹은 오용 탐지(Misuse Detection)라고 부르는 기술이 있다. 이 방법은 이미 알려진 공격에 대해서는 안정적인 탐지 성능을 보이지만 새로운 공격에는 취약하다는 단점이 있다. 따라서 최근에는 기계학습(Machine Learning) 기술을 활용하여 컴퓨터가 스스로 정상 패턴을 학습하고 그것에서 벗어나는 패턴을 공격으로 탐지하는 이상 탐지(Abnormal Detection) 기술이 활발히 연구되고 있다[2].

이상 탐지 연구는 주로 정상 데이터와 공격 데이터를 구분하는 이진 분류 방식으로 이루어지는데, 실제 시스템에서는 공격마다 다른 방어 메커니즘이 필요하므로 이진 분류만으로는 충분하지 않다[3]. 따라서 최근에는 다중 분류에 대한 연구도 이루어지고 있다. 하지만 여기에는 몇 가지 어려움이 있다. 네트워크 트래픽은 정상보다 공격 데이터가 현저하게 적어 이진 분류에서도 공격 패킷을 잡아내는 데 어려움이 있는데, 다중 분류의 경우 클래스 간 샘플 수의 차이가 더 크기 때문에 이러한 문제가 더 두드러질 수 있다. 그러므로 이러한 문제를 해결하기 위해 본 연구에서는 다중 분류를 실시하고, 전체적인 탐지율 외에도 샘플 수가 적은 클래스의 탐지율도 높일 방법을 연구할 것이다. 이를 위해 오버 샘플링을 실시하여 클래스별 샘플 수의 균형을 맞춘 데이터와 그렇지 않은 데이터가 탐지 모델의 성능에 어떠한 영향을 미치는지 알아볼 것이다. 마지막으로 차원 축소를 통해 탐지 시간을 단축할 방법을 모색할 것이다.

본 논문은 다음과 같이 구성된다. 제2장에서는 본 연구와 관련된 배경 지식과 관련 연구를 살펴본다. 제3장에서는 본 연구에서 사용된 네트워크 트래픽 데이터 세트와 본 연구에서 제안하는 네트워크 이상 탐지 모델을 설명한다. 제4장에서는 제안한 모델에 따라 실험한 결과를 분석하고 성능을 평가한다. 마지막으로 제5장에서는 결론 및 향후 연구 방향에 대해 논한다.

Ⅱ. 관련연구

이 장에서는 네트워크 이상 탐지의 기존 연구에 대해 간략히 설명한다.

2-1 네트워크 침입 탐지 시스템

침입 탐지 시스템은 정보 시스템이 공격에 대비하고 그것에 대응하도록 하는 시스템이다. 이 시스템은 다양한 시스템과 네트워크 소스에서 정보를 수집하여 분석한 후 공격을 탐지하기 위한 보안 활동들을 수행한다. 침입 탐지 시스템은 네트워크 침입 탐지 시스템, 네트워크 노드 침입 탐지 시스템(NNIDS; Network Node Intrusion Detection System), 호스트 침입 탐지 시스템으로 구분할 수 있다. 네트워크 침입 탐지 시스템은 전체 서브넷에서 전달되는 트래픽을 다루며, 네트워크 노드 침입 탐지 시스템은 네트워크에서 특정 호스트로 전달되는 트래픽을 분석한다. 마지막으로 호스트 침입 탐지 시스템은 단일 컴퓨터의 감사 기록, 사용자 정보 등을 바탕으로 침입을 탐지한다[4]. 이 중 본 연구에서 다룰 것은 네트워크 침입 탐지 시스템이다.

네트워크 침입 탐지 기법은 크게 시그니처 탐지 혹은 오용 탐지, 그리고 이상 탐지로 구분할 수 있다. 오용 탐지는 전문가가 분석한 특정 공격 패턴을 시스템에 입력하고, 이 패턴과 일치하는 패턴을 가진 패킷을 공격으로 판정하는 기법이다. 이러한 방식은 전문가가 직접 분석하여 판단해야 하고 사전에 파악된 공격만을 탐지할 수 있기 때문에 새로운 공격에 대한 실시간 탐지가 어렵다는 단점이 있다. 이러한 단점을 보완하기 위해 제안된 것이 이상 탐지이다. 이상 탐지는 정상적인 패턴을 컴퓨터가 스스로 학습하여 그 패턴에서 벗어나는 패킷을 침입으로 인지하는 방식이다. 최근 머신 러닝 및 딥러닝에 대한 연구가 활발해지며 관련 기술을 네트워크 이상 탐지 시스템에 적용하려는 연구 또한 활발히 이루어지고 있다[5].

이상 탐지 시스템 연구에서는 데이터 수집(Data Acquisition), 데이터 전처리(Data Pre-processing), 특성 선택 (Feature Selection), 분류(Classification) 모델 개발 등이 주제가 된다[6]. 데이터 수집에 관한 연구는 현실적인 데이터 수집 및 데이터 세트 개발을 위한 것이며, 데이터 전처리에 관한 연구는 주로 불균형한 데이터 세트(Imbalanced Data Set)의 균형을 맞춰 탐지 모델의 성능을 높이기 위한 것이다. 다음으로 특성 선택은 탐지 성능을 유지하면서도 컴퓨터 리소스를 최대한 적게 사용하기 위한 방향으로 연구가 진행되고 있다. 마지막으

로 분류 모델 연구는 다양한 머신 러닝 및 딥 러닝 분류 알고리즘을 적용하거나 몇 가지의 알고리즘을 결합한 하이브리드 모델을 개발하는 방향으로 진행되고 있다.

2-2 머신 러닝 기반 이상 탐지

1) 오버 샘플링(Oversampling)

클래스의 크기가 불균형한 데이터 세트는 분류 모델의 성능에도 영향을 준다. 특히 다중 분류 이상 탐지 모델에서는 특정 공격의 수가 다른 공격이나 정상 데이터에 비해 현저히 적을 경우 탐지 성능이 떨어질 수 있다. 이러한 문제를 해결하기 위해 몇 가지 방법이 제안되었다.

문헌 [6]은 딥러닝 기법 중 하나인 생성적 적대 신경망(GAN; Generative Adversarial Network)[7]을 이용하여 데이터 불균형 문제를 개선했다. 이 방법은 SCAE+SVM, VCDL, STL, S-NDAE 모델보다 모든 지표에서 전반적으로 더 좋은 결과를 보였다. CIC-DDoS 2019 데이터 세트의 다중 분류 결과는 정확도 98.53%, F-1 점수 99.17%로 좋은 성능을 보였다.

2) 특성 선택 (Feature Selection)

특성 선택 혹은 차원 축소는 모델에 사용되는 특성의 개수를 줄여 모델의 성능을 높이는 기술이다. 너무 많은 특성은 탐지 속도를 느리게 만들거나 모델의 성능을 떨어뜨릴 수 있다. 반대로 여러 특성들 중 분류에 유용한 특성들만을 추출하여 훈련하면 탐지 속도도 높이고 모델의 성능도 향상시킬 수 있다. 따라서 특성 선택에 대한 연구도 활발히 이루어지고 있다.

문헌 [8]은 NSL-KDD 데이터 세트의 41개의 특성들 중 관련도가 높은 5개의 특성을 선택하는 필터 방식을 사용했다. 분류 알고리즘으로는 XGBoost와 결정트리(Decision Tree), 랜덤 포레스트(RF; Random Forest)를 사용하였는데 XGBoost를 사용한 모델이 F1-score, 정밀도, 재현율 모두에서 가장 높은 점수를 보였다.

자연에서 영감을 얻은 방식도 있다. 문헌 [9]는 IDS의 성능을 높이기 위한 방법으로 생명체에게서 영감을 받은 메타휴리스틱 알고리즘(Bio-inspired Metaheuristic Algorithm)을 적용한 특성 선택을 제안하였다. 이 연구에서 제안한 차원축소 알고리즘은 PSO(Particle Swarm Optimization), MVO(Multi-verse optimizer), WOA(Whale Optimization Algorithm, FFA(Firefly Algorithm), BAT(Bat Algorithm)를 조합한 42개의 하이브리드 알고리즘이다. 이중 MFO-WOA, FFA-GWO 모델이 높은 정확도를 유지하면서도 44개의 특성을 15개로 줄이는 가장 좋은 결과를 보여줬다. 분류 모델로는 서포트 벡터 머신(SVM; Support Vector Machine), J48, RF를 사용하였다.

문헌 [10]은 적층적 오토인코더(Stacked Autoencoder)를 사용한 후 분류 모델은 SVM을 적용하는 SAE-SVM모델을 제안했다. 특성 추출의 효용성을 확인하기 위해 훈련 시간과 예측 시간을 평가 지표에 포함했다. 기존 연구에서 제안한 PCA-GMM 모델은 86.2%의 정확도와 40분의 예측 시간을 기록했지만, SAE-SVM모델은 90.2%의 정확도를 유지하면서 시간은 1분으로 단축하였다.

문헌 [11]은 DBN(Deep Belief Network)과 LSTM(Long Short-Term Memory)를 결합한 DBN-LSTM 모델을 제안했다. DBN은 준지도 학습으로 지도 학습과 비지도 학습의 조합으로 이루어져 있다. 비지도 학습인 RBM(Restricted Boltzmann Machine)으로 순차적으로 훈련한 다음 전체 시스템이 지도 학습 방식으로 세밀하게 조정하는 방식이다. 이렇게 얻어진 데이터를 LSTM에 보내 행동 모델링(Behavior Modeling)을 실시했다. 실험 결과 DNN이나 LSTM을 단독으로 사용한 것보다 DBN과 함께 사용하는 것이 더 좋은 결과를 보였다. 또한, DBN+DNN 조합보다 DBN+LSTM 조합이 더 좋은 성능을 보였다. 데이터의 불균형한 문제를 해결하지 않아 정상 데이터나 샘플 수가 많은 공격에서는 99% 이상의 높은 정확도와 정밀도, 재현율을 나타냈지만, 레코드 수가 적은 공격에서는 상대적으로 성능이 떨어지는 것으로 나타났다.

특성 선택 방법 중 하나로 주성분 분석(PCA; Principal Component Analysis)이 있다[12]. PCA는 가장 중요한 분산과 함께 오리지널 파라미터의 직교 선형 조합(orthogonal linear combinations)을 이용해 유의성을 기반으로 변수의 수를 줄이는 방식이다. 문헌 [13]은 실험을 통해 PCA가 침입탐지 속도를 높이고 메모리 공간과 CPU 시간 비용을 최소화할 수 있음을 보여주었다.

Ⅲ. 제안방법론

3-1 데이터 세트

네트워크 이상 탐지 시스템 연구에서는 실제 환경을 반영한 현실적이고 신뢰성 있는 데이터 세트가 중요하다. 특히 시그니처 탐지로는 잡아내지 못하는 새로운 공격 상황을 반영한 데이터 세트를 만드는 것이 연구자들의 주요 목표 중 하나이다. KDD-CUP'99 역시 이러한 노력의 결과로 만들어져 네트워크 이상 탐지 분야에서 가장 널리 사용되어왔다. 하지만 2009년 이 데이터 세트에 대한 분석을 실시한 결과 몇 가지 내재적인 문제점들이 발견되었다. 그중 하나는 중복된 레코드가 많다는 점이다. 분석 결과 훈련 세트와 테스트 세트에서 각각 78%, 75%의 중복된 레코드가 발견되었다[14]. 데이터 세트의 중복성은 훈련 모델의 편향을 야기할 수 있다.

NSL-KDD는 이러한 문제점을 개선하기 위해 중복된 레코드를 제거한 데이터 세트이다. 비록 실제 네트워크를 완벽히 대표할 수 없고 공격 클래스별 데이터 수가 불균형한 한계는 있지만, 여전히 네트워크 이상 탐지 연구에 널리 이용되는 벤

치마크 데이터 세트이다. 따라서 본 연구에서도 NSL-KDD를 대상으로 실험을 진행한다.

NSL-KDD 데이터 세트는 총 41개의 특성들과 함께 공격의 난이도와 종류(정상 포함)을 구분하는 레이블로 구성되어 있다. 본 연구에서는 난이도는 고려하지 않으며, 공격 종류 카테고리를 목푯값으로 한다.

protocol_type, service, flag는 문자로 된 범주형 데이터이므로 분류 모델이 처리할 수 없다. 따라서 숫자로 변환하고 원-핫 인코딩(One-hot Encoding)을 실시한다. 그런데 service와 flag 종류의 수는 훈련 세트와 테스트 세트에서 서로 다르다. 따라서 인코딩에 앞서 공통된 값을 제외한 값들은 모두 'others'로 바꾸어 두 데이터 세트의 특성 수를 일치시킨다.

한편 데이터 세트의 각 특성값들은 그 스케일이 서로 다르다. 따라서 Standard Scaler를 통해 정규화를 실시한다. 이때 원-핫 인코딩된 protocol_type, service, flag 값은 포함하고 difficulty_level과 attack_cat은 제외한다. NSL-KDD 데이터 세트에 포함된 공격유형은 4개의 클래스(DoS, Probe, R2L, U2R)로 나뉜다. 여기에 정상(Normal) 레이블까지 더해져 총 5개의 클래스를 이룬다.

본 연구에서 사용할 분류 모델은 데이터의 샘플들을 이 5개의 클래스로 분류하는 모델이다. 클래스 값이 문자열로 되어 있어 훈련에 앞서 숫자로 변환하는 작업을 한다. 결측치나 중복된 데이터는 없어 이에 대한 전처리는 필요하지 않다.

NSL-KDD 데이터 세트는 훈련 데이터와 테스트 데이터가 별도의 파일로 나뉘어 있다. 따라서 훈련 데이터와 테스트 데이터를 나누는 작업은 필요하지 않다. 검증을 위해 훈련 데이터의 20%를 검증 세트로 따로 떼어 두는 작업만 진행한다. 테스트 세트에는 훈련 세트에 없는 새로운 공격이 포함되어 있다.

3-2 오버 샘플링과 차원 축소

NSL-KDD는 그림 1, 2에서 나타나는 것처럼 공격 클래스별 레코드 수의 차이가 매우 크다. 이 경우 다중 분류를 실시할 때 수가 적은 Probe, U2R, R2L 공격에 대한 탐지력이 떨어진다는 문제가 있다. 본 연구에서는 다중 분류를 진행할 것이므로 오버 샘플링 기법인 SMOTE(Synthetic Minority Over-sampling Technique)[15]를 사용하여 훈련 데이터 세트의 공격 클래스별 샘플 수의 균형을 맞출 것이다.

SMOTE는 기존의 대체를 통한 오버 샘플링 방식 대신 합성(Synthetic) 샘플을 생성하여 소수 클래스가 오버 샘플링되는 방식을 채택한 기술이다[15]. 이 방식은 데이터 공간(Data Space)이 아닌 특성 공간(Feature Space)에서 작동함으로써 애플리케이션에 덜 특정한 방식으로 합성 예제를 생성한다. 소수 클래스는 각 소수 클래스 샘플을 취하고 k의 일부 또는 모두를 연결하는 선분을 따라 합성 예제를 도입하여 오버 샘플링 된다.

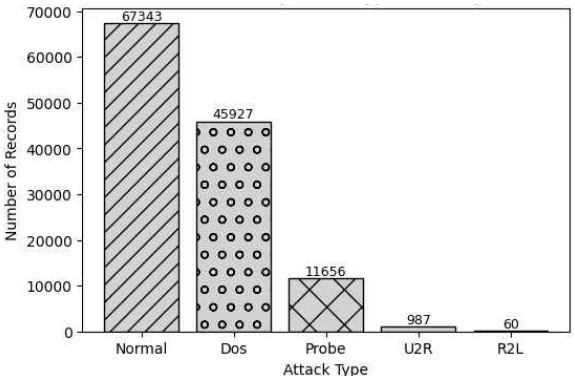

그림 1. 공격 유형별 레코드의 수 - 훈련 데이터 세트
Fig. 1. Number of records by attack type - Training dataset

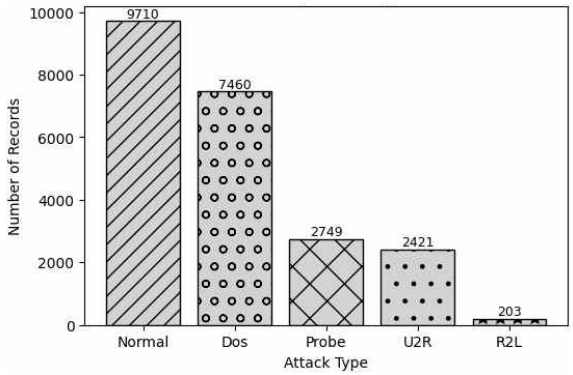

그림 2. 공격 유형별 레코드 수 - 테스트 데이터 세트
Fig. 2. Number of records by attack type - Test Dataset

오버 샘플링 후에는 오버 샘플링을 적용한 모델과 적용하지 않은 모델을 비교하여 불균형한 데이터가 모델의 성능에 어떠한 영향을 미치는지 분석한다.

차원 축소 방식으로는 주성분 분석(PCA)을 적용한다. 차원 축소를 적용한 모델과 적용하지 않은 모델의 성능을 비교하여 차원 축소 정도에 따라 분류 모델의 성능을 측정한다. 59개의 특성을 30개, 15개, 7개, 5개, 2개로 축소한 각 모델의 성능을 비교하여 차원 축소의 정도가 모델의 성능에 어떤 영향을 미치는지 살펴본다.

3-3 분류 모델

분류 모델로는 SVM, RF, XGBoost, KNN, 그리고 심층 신경망(DNN; Deep Neural Networks)과 합성곱 신경망(CNN; Convolutional Neural Networks)을 사용한다. 각 모델의 하이퍼파라미터는 표 1과 같이 설정하고, 차원의 수(2, 7, 15, 30 차원)와 오버 샘플링 여부만 다르게 하여 실험을 진행하였다.

표 1. 모델별 하이퍼파라미터
Table 1. Hyperparameters for each model

Model	Hyperparameter	Value
SVM	kernel gamma C	rbf 1 10
RF	n_estimators max_depth min_samples_leaf min_samples_split n_jobs	100 12 8 20 -1
XGBoost	-	-
KNN	n_neighbors	3
DNN	dense activation dropout rate activation loss optimizer epochs	300, 100, 5 Relu 0.2 softmax categorical_crossentropy nadam 30
CNN	units activation dropout rate activation loss optimizer epochs	64, 128, 256, 128, 64 Relu 0.5 softmax categorical_crossentropy nadam 30

3-4 실험 워크플로우

본 연구의 실험은 그림 3과 같이 구성된다. 데이터 전처리 후 데이터 분할 과정을 거쳐 SMOTE와 PCA를 적용한 데이터 세트를 준비한다. 그 후 각 데이터 세트에 분류 알고리즘을 적용한 후 각 모델의 성능을 평가한다.

Ⅳ. 실험 결과

4-1 차원축소 모델별 성능 분석

차원 축소를 적용한 모델들의 정확도를 비교하면 그림 4와 같다. SVM은 7차원으로 축소했을 때 정확도가 가장 높고, 2차원의 정확도가 가장 떨어졌다. 랜덤 포레스트 역시 7차원에서 정확도가 가장 높고, 2차원의 정확도가 가장 낮았다. KNN과 XGBoost는 차원을 축소할수록 정확도가 낮아지는 결과를 보였다. DNN는 30차원으로 줄였을 때는 오히려 정확도가 떨어졌다가 15차원에서 정확도가 가장 높아졌고 7차원과 2차원에서는 감소하였다. 마지막으로 CNN 모델은 30차원으로 축소했을 때 정확도가 좋아졌다가 점차 감소하였다. 전체적으로 CNN모델을 30차원으로 축소했을 때의 정확도가 가장 높은 것을 알 수 있다.

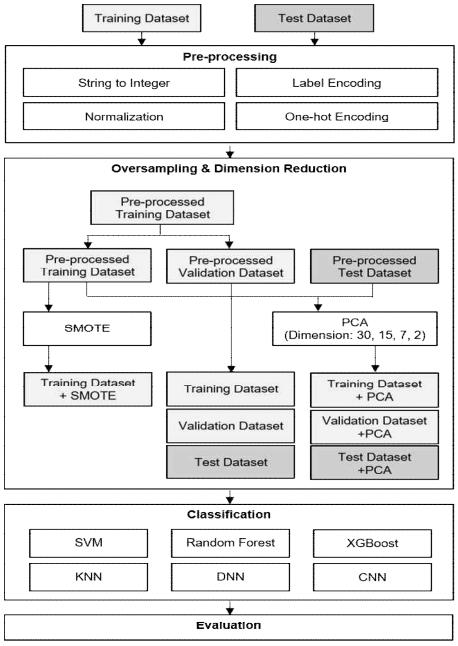

그림 3. 실험 워크플로우
Fig. 3. Experimental workflow

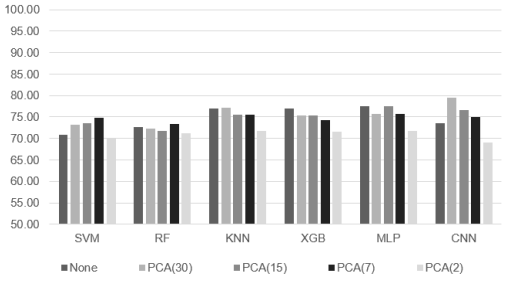

그림 4. 모델별 정확도
Fig. 4. Accuracy of models

다음 그림 5에서 볼 수 있듯 정밀도는 차원축소로 인한 성능의 향상이 미미하거나 더 낮아지는 결과를 보인다. 하지만 CNN의 경우 30차원에서 정밀도가 가장 높았고, 15차원, 7차원의 정밀도도 차원축소를 적용하지 않은 모델에 비해 더 높은 결과를 보인다. 모든 모델에서 2차원의 정밀도가 가장 낮았다. 또한 RF의 정밀도가 전반적으로 낮았다.

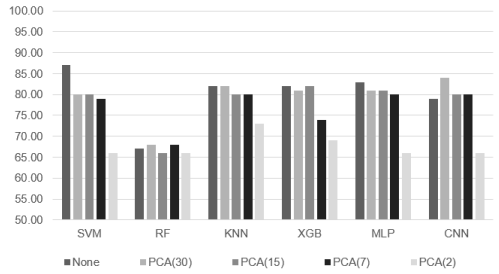

그림 5. 모델별 정밀도
Fig. 5. Precision of models

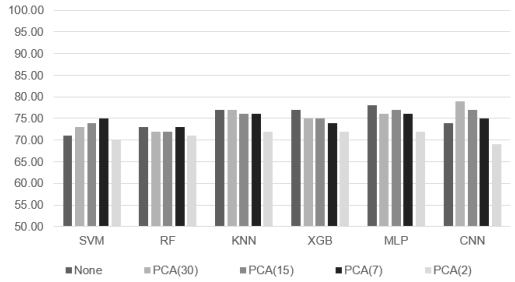

그림 6. 모델별 재현율
Fig. 6. Recall of models

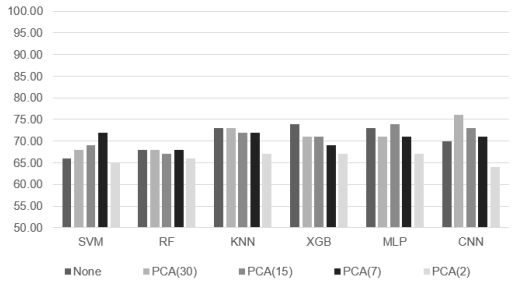

그림 7. 모델별 F1 점수
Fig. 7. F1-score of models

모델별 재현율을 비교한 그림 6을 보면 SVM의 경우 재현율은 차원이 축소될수록 높아졌지만, 2차원에선 감소하였다. RF는 모델 간 차이가 미미하였고 역시 2차원에서 가장 낮은 재현율을 보였다. 다음 KNN, XGB, DNN은 차원이 감소할수록 재현율이 낮아지는 추세를 보였다. CNN 모델은 차원 축소 모델들의 재현율이 차원 축소를 하지 않은 모델보다 높았으나 2차원 모델은 원래 모델보다 재현율이 떨어지는 결과를 보였다. 또한 전체적으로 차원이 감소될수록 재현율 또한 감소되었다.

그림 7을 보면 F1 점수는 재현율과 비슷한 양상을 보이는 것을 확인할 수 있다. CNN 30차원 모델의 F1 점수가 가장 높으며, 차원이 낮아질수록 점수도 낮아졌고 2차원에서 가장 낮았다.

표 2. 모델별 예측 시간(sec)
Table 2. Prediction time(sec) of models

Model	None	PCA(30)	PCA(15)	PCA(7)	PCA(2)
SVM	0.917	0.429	0.345	0.169	0.870
RF	0.011	0.005	0.013	0.006	0.005
KNN	1.979	1.547	0.172	0.048	0.031
XGB	0.016	0.018	0.024	0.017	0.016
DNN	0.237	0.060	0.045	0.119	0.062
CNN	0.107	0.059	0.122	0.131	0.088

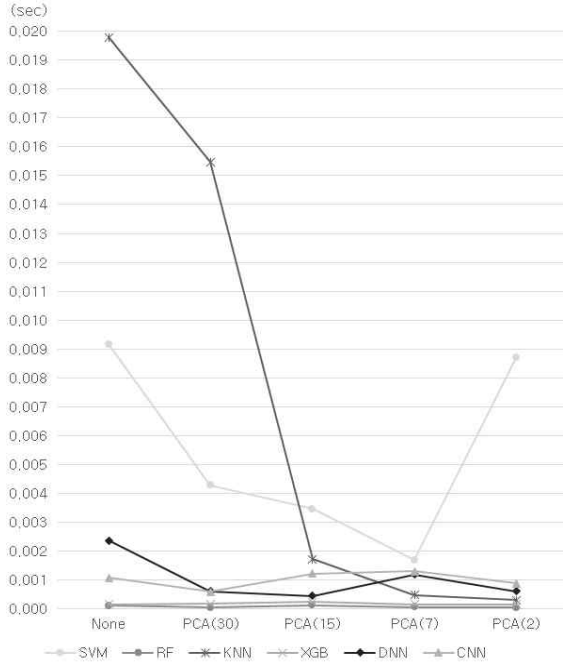

그림 8. 모델별 예측 시간
Fig. 8. Prediction time of models

각 모델의 예측 시간을 표와 그래프로 정리하면 표 2, 그림 8과 같다. 이때 시간 측정 기준은 1000개의 레코드(샘플)를 처리하는 데 걸리는 시간이며, 시간 단위는 초(sec)이다.

차원축소의 영향을 가장 크게 받은 것은 KNN과 SVM 모델이다. KNN의 경우 예측 시간이 가장 긴 1.979였으나 차원을 축소할수록 시간이 단축되어 0.031까지 감소되었다. SVM 역시 차원을 축소할수록 예측 시간이 단축되었으나 2차원에서는 오히려 시간이 증가하였다. RF와 XGB 모델은 수치의 변화가 미미하였다. DNN은 차원축소 이후 예측 시간이 감소하긴 했지만, 0.237에서 0.045까지 감소하긴 했지만, 7차원에서는 0.119로 증가하는 양상을 보였다. 마지막으로 CNN은 30차원으로 축소했을 땐 0.107에서 0.059까지 감소하였지만, 15차원과 7차원에선 0.122, 0.131로 차원축소 전보다 시간이 오히려 증가하였다.

4-2 오버 샘플링 모델별 성능 분석

표 3은 SVM 모델에 오버 샘플링을 적용하기 전후의 각 클래스별 정밀도와 재현율, F1 점수를 비교한 혼동 행렬이다. 표 3에서는 SVM은 오버 샘플링 후에도 큰 차이를 보이지 않았으나, 레코드 수가 가장 적은 R2L의 정확도와 재현율, f1-score가 0에서 0.04, 0.02, 0.03으로 미미하게나마 증가된 것을 확인할 수 있다.

표 4는 오버 샘플링 전후 정밀도와 재현율, F1 점수를 비교한 표이다. KNN의 경우 Normal, DoS, Probe는 탐지 성능이 더 떨어지거나 미미하게 증가하였지만, U2R과 R2L은 재현율과 f1-score 수치가 상당히 개선되었다.

표 3. 오버 샘플링 전후 혼동 행렬 - SVM
Table 3. Confusion matrix before/after oversampling - SVM

	SVM			SVM + SMOTE		
	precision	recall	f1-score	precision	recall	f1-score
Normal	0.60	0.99	0.75	0.60	0.99	0.75
Dos	0.99	0.71	0.83	0.99	0.71	0.83
Probe	0.92	0.43	0.58	0.90	0.43	0.59
U2R	0.90	0.01	0.03	0.91	0.01	0.03
R2L	0.00	0.00	0.00	0.04	0.02	0.03

표 4. 오버 샘플링 전후 혼동 행렬 - KNN
Table 4. Confusion matrix before/after oversampling - KNN

	KNN			KNN + SMOTE		
	precision	recall	f1-score	precision	recall	f1-score
Normal	0.67	0.97	0.79	0.71	0.94	0.81
Dos	0.96	0.81	0.88	0.94	0.71	0.81
Probe	0.85	0.75	0.80	0.60	0.71	0.65
U2R	0.96	0.03	0.06	0.89	0.33	0.48
R2L	0.36	0.11	0.17	0.27	0.28	0.27

표 5. 오버 샘플링 전후 혼동 행렬 - RF
Table 5. Confusion matrix before/after oversampling - RF

	RF			RF + SMOTE		
	precision	recall	f1-score	precision	recall	f1-score
Normal	0.62	0.97	0.76	0.71	0.93	0.80
Dos	0.95	0.72	0.82	0.91	0.81	0.86
Probe	0.87	0.62	0.72	0.83	0.76	0.79
U2R	0.00	0.00	0.00	0.83	0.07	0.12
R2L	0.00	0.00	0.00	0.06	0.19	0.09

표 5를 보면 RF의 경우 전반적으로 재현율과 F1점수의 수치가 개선되었고, 특히 U2R과 R2L은 오버샘플링 전에는 정확도와 재현율, F1점수가 0이었지만 오버 샘플링을 실시한 후에는 각 수치가 모두 상승한 것을 알 수 있다.

표 6은 오버 샘플링 한 데이터 세트를 XGBoost로 분류했을 때는 오버 샘플링 전과 수치상 큰 성능의 차이가 없다는 것을 보여준다. 표 7에서는 DNN 모델에 오버 샘플링을 적용했을 때는 U2R과 R2L의 성능이 다소 개선된 것을 확인할 수 있다. 오버 샘플링 전에는 전혀 탐지하지 못했는데 오버 샘플링 후에는 U2R은 재현율과 F1점수가 각각 0.06, 0.11로 상승하였고, R2L은 0.18과 0.12로 증가하였다.

표 6. 오버 샘플링 전후 혼동 행렬 - XGBoost
Table 6. Confusion matrix before/after oversampling - XGBoost

	XGBoost			XGBoost + SMOTE		
	precision	recall	f1-score	precision	recall	f1-score
Normal	0.67	0.97	0.79	0.71	0.93	0.80
Dos	0.96	0.83	0.89	0.91	0.81	0.86
Probe	0.85	0.60	0.71	0.83	0.76	0.79
U2R	0.99	0.10	0.19	0.90	0.07	0.12
R2L	0.09	0.03	0.06	0.06	0.19	0.09

표 7. 오버 샘플링 전후 혼동 행렬 - DNN
Table 7. Confusion Matrix Before/After Oversampling - DNN

	DNN			DNN + SMOTE		
	precision	recall	f1-score	precision	recall	f1-score
Normal	0.67	0.97	0.80	0.69	0.97	0.81
Dos	0.97	0.83	0.89	0.97	0.83	0.89
Probe	0.88	0.75	0.81	0.83	0.68	0.75
U2R	1.00	0.00	0.00	0.72	0.06	0.11
R2L	0.00	0.00	0.00	0.09	0.18	0.12

표 8. 오버 샘플링 전후 혼동 행렬 - CNN
Table 8. Confusion matrix before/after oversampling - CNN

	CNN			CNN + SMOTE		
	precision	recall	f1-score	precision	recall	f1-score
Normal	0.65	0.96	0.77	0.68	0.93	0.78
Dos	0.97	0.82	0.78	0.97	0.78	0.87
Probe	0.75	0.48	0.59	0.62	0.50	0.56
U2R	0.98	0.13	0.24	0.87	0.17	0.29
R2L	0.00	0.00	0.00	0.05	0.17	0.08

마지막으로 표 8은 CNN 모델의 오버 샘플링 전후 혼동 행렬을 나타낸 표이다. R2L의 정확도, 재현율, F1 점수가 모두 0 이었지만, 오버 샘플링 이후 각각 0.05, 0.17, 0.08로 증가하였다.

전반적으로 오버 샘플링을 실시하면 성능이 높아진다는 것을 알 수 있다. 특히 RF와 XGBoost가 다른 분류 모델에 비해 오버 샘플링 이후의 성능이 상대적으로 많이 높아졌다. 반면 SVM이나 KNN은 오버 샘플링 전과 후가 근소한 차이를 보이거나 오버 샘플링 후 성능이 더 낮아지는 결과를 보여줬다. 트리 기반의 RF와 XGBoost가 SMOTE의 긍정적인 영향을 더 많이 받는다는 것을 알 수 있다.

V. 결 론

본 연구에서는 보안 시스템이 공격의 유형까지 인지할 수 있도록 공격들이 5개의 클래스로 나뉘어 있는 NSL-KDD 데이터 세트를 사용하여 다중 분류를 실시했다. 이때 전체 데이터 세트에 대한 정확도뿐만 아니라 샘플 수가 적은 공격 클래스의 탐지율을 높이는 데 중점을 두었다. 또한 차원 축소를 활용하여 성능의 향상과 예측 시간의 단축을 모색했다.

먼저 PCA를 이용한 차원 축소가 모델의 성능에 어떠한 영향을 미치는지 살펴보았다. 실험 결과 차원이 작아진다고 해서 무조건 성능이 좋아지거나 예측 시간이 줄어드는 것이 아니며, 분류 모델마다 차원 정도에 따른 성능이나 속도 변화의 추이가 서로 다르다는 것을 확인하였다. 차원축소 결과 CNN이 가장 좋은 결과를 보였는데, PCA를 실시하기 전 모델의 정확도는 4번째로 높았으나 30차원으로 축소한 후에는 79.43%로 가장 좋은 성능을 보였다. 하지만 2차원으로 축소했을 때는 가장 낮은 68.99%를 보여 과도한 차원 축소는 성능을 오히려 감소시킨다는 것을 확인했다.

샘플 수가 적은 공격 클래스를 탐지하기 위해서 SMOTE로 오버 샘플링을 진행한 후 원래의 모델과 비교했다. 실험 결과 DNN, CNN은 정확도가 각각 2.01%, 0.64% 낮아졌지만, U2R과 R2L의 재현율과 F1 점수가 상승한 것을 확인하였다. KNN의 경우 정확도에서는 큰 변화가 없지만 U2R과 R2L의 재현율과 F1 점수가 크게 상승한 것을 확인할 수 있었다. 이상 탐지 모델의 목표가 소수 클래스까지 탐지하는 것이라면 이는 유의미한 결과일 것이다.

본 연구에서는 PCA 기반의 차원 축소와 SMOTE 기반의 오버 샘플링이 전반적으로 성능 및 예측 시간에 어떠한 영향을 주는지에 대한 연구를 진행하였다. 향후 연구에서는 딥러닝 방식의 오토인코더(Autoencoder)[16]와 같은 비지도 학습 기반 차원 축소 방법과 GAN 기반의 오버 샘플링 기법을 적용하여 탐지 성능 및 예측 시간에 미치는 영향을 연구할 것이다.

감사의 글

본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. NRF-2019R1F1A1059036).

참고문헌

[1] R. Boutaba, M. A. Salahuddin, N. Limam, S. Ayoubi, N. Shahriar, F. Estrada-Solano, and O. M. Caicedo, “A comprehensive survey on machine learning for networking: evolution applications and research opportunities,” *Journal of Internet Services and Applications*, Vol. 9, No. 1, June 2018. https://doi.org/10.1186/s13174-018-0087-2

[2] P. Mell, “Understanding Intrusion Detection Systems,” in *IS Management Handbook*, 8th ed. Auerbach Publications, pp. 409-418, June 2003. https://doi.org/10.1201/9781420031393

[3] H. Zhang, L. Huang, C. Wu, and Z Li, “An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset,” *Computer Networks*, Vol. 177, August 2020. https://doi.org/10.1016/j.comnet.2020.107315

[4] D. Rozenblum, “Understanding Intrusion Detection Systems,” *SANS White Paper*, August 2001. https://sansorg.egnyte.com/dl/RssXub01Q9

[5] G. Pang, C. Shen, L. Cao, and A. Van Den Hengel, “Deep Learning for Anomaly Detection: A Review,” *ACM Computing Surveys*, Vol. 54, No. 2, pp. 1-38, March 2022. https://doi.org/10.1145/3439950

[6] L. Nie, Y. Wu, X. Wang, L. Guo, G. Wang, X. Gao, and S. Li, “Intrusion Detection for Secure Social Internet of Things Based on Collaborative Edge Computing: A Generative Adversarial Network-Based Approach,” *IEEE Transactions on Computational Social Systems*, Vol. 9, No. 1, pp. 134-145, February 2022. https://doi.org/10.1109/TCSS.2021.3063538

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, ... and Y. Bengio, “Generative adversarial networks,” *Communications of the ACM*, Vol. 63, No. 11, pp. 139-144, November 2020. https://doi.org/10.1145/3422622

[8] Z. Ahmad, A. S. Khanm, C. W. Shiang, J. Abdullah, and F. Ahmaad, “Network intrusion detection system: A systematic study of machine learning and deep learning approaches,” *Transactions on Emerging Telecommunications Technologies*, Vol. 32, No. 1, e4150, January 2021. https://doi.org/10.1002/ett.4150

[9] O. Almomani, “A Hybrid Model Using Bio-inspired Metaheuristic Algorithms for Network Intrusion Detection System,” *Computers, Materials & Continua*, Vol. 68, No. 1, pp. 409-429, March 2021. https://doi.org/10.32604/cmc.2021.016113

[10] S. N. Mighan and M. Kahani, “A novel scalable intrusion detection system based on deep learning,” *International Journal of Information Security*, Vol. 20, pp. 387-403, June 2021. https://doi.org/10.1007/s10207-020-00508-5

[11] A. Chen, Y. Fu, X. Zheng, and G. Lu, “An efficient network behavior anomaly detection using a hybrid

DBN-LSTM network," *Computers & Security*, Vol. 114, 102600, March 2022. https://doi.org/10.1016/j.cose.2021.102600

[12] D. Granato, J. S. Santos, G. B. Eschera, B. L. Ferreira, and R. M. Maggio, "Use of principal component analysis (PCA) and hierarchical cluster analysis (HCA) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective," *Trends in Food Science Technology*, Vol. 72, pp. 83-90, February 2018. https://doi.org/10.1016/j.tifs.2017.12.006

[13] F. E. Heba, A. Darwish, A. E. Hassanien, and A. Abraham, "Principle components analysis and Support Vector Machine based Intrusion Detection System," in *10th International Conference on Intelligent Systems Design and Applications,* Cairo, Egypt, November/December 2010. https://doi.org/10.1109/ISDA.2010.5687239

[14] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *IEEE Symposium on Computational Intelligence for Security and Defense Applications*, Ottawa, ON, Canada, July 2009. https://doi.org/10.1109/CISDA.2009.5356528

[15] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE : Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, Vol. 16, pp. 321-357, June 2002. https://doi.org/10.1613/jair.953

[16] M. A. Kramer, "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*, Vol. 37, No. 2, pp. 233-243, February 1991. https://doi.org/10.1002/aic.690370209

윤지은(Ji-Eun Yoon)

2010년 : 한동대학교 공연 영상학, 국제지역학 학사

2019년 : 한국학중앙연구원 한국학대학원 인류학 석사

2020년～현 재: 아주대학교 대학원 지식정보공학과 석사과정

※관심분야 : 정보보안(Information Security), 기계학습(Machine Learning)

김강석(Kangseok Kim)

2007년 : Indiana University (at Bloomington) 컴퓨터공학과 (공학박사)

2010년～2016년: 아주대학교 대학원 지식정보공학과 연구교수

2016년～현 재: 아주대학교 사이버보안학과 부교수

※관심분야 : 정보보안(Information Security), 딥러닝 응용 보안(Applied Deep Learning for Security)