Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.4, Aug. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.4.795

네트워크 비정상 탐지를 위한 속성 축소를 반영한 의사결정나무 기술

강 구 홍†‡
서원대학교 정보통신공학과

Decision Tree Techniques with Feature Reduction for Network Anomaly Detection

Koohong Kang†‡
Dept. of Information and Communications Eng., Seowon University

요 약

최근 알려지지 않은 공격에 대처하기 위한 네트워크 비정상(anomaly) 탐지 기술에 대한 관심이 한층 높아지고 있다. 이러한 기술 개발을 위해 데이터 마이닝(data mining), 기계학습(machine learning), 그리고 딥러닝(deep learning)등을 활용한 다양한 연구가 진행되고 있다. 본 논문에서는 분류(classification) 문제를 다루는 데이터 마이닝 기술 중 가장 전통적인 방법 중 하나인 의사결정나무(decision tree)를 이용하여 NSL-KDD 데이터 셋을 대상으로 네트워크 비정상 탐지 가능성을 보여준다. 의사결정나무의 과대적합(over-fitting) 단점을 해소하기 위해 카이-제곱(chi-square) 테스트를 통해 최적의 속성 선택(feature selection)을 수행하고, 선택된 13개의 속성을 사용한 의사결정나무 모델 환경에서 NSL-KDD 시험 데이터 셋 KDDTest+에 대해 84% 그리고 KDDTest-21에 대해 70%의 네트워크 비정상 검출 정확도를 보였다. 제시된 정확도는 기존 의사결정나무 모델 적용 시 이들 시험 데이터 셋을 대상으로 알려진 정확도 81% 그리고 64% 수준과 비교해 약 3% 그리고 6% 각각 향상된 결과다.

ABSTRACT

Recently, there is a growing interest in network anomaly detection technology to tackle unknown attacks. For this purpose, diverse studies using data mining, machine learning, and deep learning have been applied to detect network anomalies. In this paper, we evaluate the decision tree to see its feasibility for network anomaly detection on NSL-KDD data set, which is one of the most popular data mining techniques for classification. In order to handle the over-fitting problem of decision tree, we select 13 features from the original 41 features of the data set using chi-square test, and then model the decision tree using TensorFlow and Scik-Learn, yielding 84% and 70% of binary classification accuracies on the KDDTest+ and KDDTest-21 of NSL-KDD test data set. This result shows 3% and 6% improvements compared to the previous 81% and 64% of binary classification accuracies by decision tree technologies, respectively.

Keywords: Network Anomaly Detection, NSL-KDD Data Set, Decision Tree, Feature Selection

Received(04. 10. 2019), Modified(05. 29. 2019), Accepted(06. 17. 2019)

† 주저자, khkang@seowon.ac.kr
‡ 교신저자, khkang@seowon.ac.kr(Corresponding auther)

I. 서 론

사회, 경제, 그리고 개인의 일상생활이 인터넷 사용에 더욱 의존함에 따라서 다양한 유형의 공격자들로부터 네트워크를 안전하게 보호하는 것은 국가와 기업뿐만 아니라 우리 개인에게 조차도 첫 번째 관심사가 되었다. 침입탐지시스템(IDS: Intrusion Detection System)과 침입방지시스템(IPS: Intrusion Prevention System)은 인터넷에 연결된 컴퓨터 및 네트워크를 해커로부터 보호하기 위해 사용되는 가장 대표적인 보안장비다[1].

오늘날 사용되고 있는 대부분의 IDS/IPS는 악의적인 행위를 탐지할 수 있는 규칙(rule) 혹은 시그너처(signature)를 사전에 준비하고 이를 기준으로 공격을 탐지하는 규칙-기반(혹은 misuse) 탐지 기술이다. 따라서 공격 패턴이 사전에 잘 정의된 알려진 공격(known attacks)들에 대한 검출 성능(오탐율(false positive) 및 미탐율(false negative))은 매우 우수하다. 그러나 이러한 규칙-기반 IDS/IPS로는 알려지지 않은 취약점(unknown vulnerability)이나 혹은 아직 해당 익스플로잇(exploits)을 방어하기 위한 규칙이 마련되지 않은 "제로-데이 공격(zero-day attacks)" 혹은 "변종 공격(polymorphic malware)"을 검출하는 것은 불가능하다[1,2]. 이러한 문제점을 극복하기 위해 인터넷 보안 시장에서는 비정상(anomaly) 검출 기술에 많은 관심을 보이고 있다. 비정상 탐지 기술은 정상적인 행위에 대한 프로파일(profiles)을 확보하고 이를 근거로 공격행위로 인해 발생되는 비정상 행위를 탐지하는 기술이다. 따라서 공격에 대한 오탐 및 미탐 확률이 매우 높은 문제점을 가지고 있다[2].

비정상 기반 IDS/IPS의 가장 큰 단점인 낮은 탐지 정확도를 높이기 위해서 데이터 마이닝(data mining), 기계학습(machine learning), 그리고 딥러닝(deep learning) 기술들을 활용한 연구결과들이 지난 수년간 꾸준히 발표되고 있다. 본 논문에서는 데이터 마이닝 기법 중에서 가장 많이 사용되는 의사결정나무(DT: decision tree)를 사용하여 네트워크 비정상 검출 가능성을 제시하고자 한다. 물론, DT를 IDS에 적용한 기존 연구[3,4,5,6,7]들이 존재하지만, 본 논문에서는 이들 연구결과들의 문제점을 지적하고 성능을 개선하고자 한다.

네트워크 비정상 검출을 위한 DT를 모델링 하기 위해서는 모델을 학습시키고 평가할 훈련 및 시험 데이터 셋이 필요하다. 기존의 많은 연구들이 DARPA 98 Lincoln Lab 평가 데이터 셋(DARPA Set)을 훈련 및 시험 데이터 셋으로 사용해 왔다[7]. 이들 데이터 셋은 약간의 변경을 통해 제3회 국제 'Knowledge Discovery and Data Mining Tool Competition'의 데이터 셋(KDD Cup 1999로 명칭)으로 재사용되었으며[8], 최근에는 많은 중복된 레코드를 제거하여 NSL-KDD 데이터 셋으로 완성되었다[9]. 본 논문에서 사용할 NSL-KDD 데이터 셋은 훈련 데이터 셋 KDDTrain+와 KDDTrain-, 그리고 시험 데이터 셋 KDDTest+와 KDDTest-21로 최종 정리되었다.

Tavallaee et al.[7]는 C4.5 DT 알고리즘을 사용하여 KDDTest+에 대해 81.05%와 KDDTest-21에 대해 63.97%의 공격 탐지 정확도(accuracy)를 제공하였다. 한편, 데이터 마이닝 기술과 기계학습을 적용한 IDS 개발 과정에서 데이터 처리 속도와 탐지의 정확성을 향상시키기 위해 속성 선택(feature selection) 혹은 속성 축소(feature reduction)에 대한 연구도 진행되었다[10,11,12]. 특히 DT 모델의 경우, 훈련 데이터 셋에 과대적합(over-fitting)하는 경향이 있으며, 이로 인해 실제 데이터 셋에 대한 분류 성능이 감소하는 경향을 보인다. 속성 축소는 이러한 과대적합 문제점을 해결하기 위해 매우 유용한 접근법으로 사용되어 왔다. 본 논문에서는 사이킷런(Scikit Learn) 라이브러리[13]가 제공하는 카이-제곱 테스트를 사용해 DT의 최대 분류 정확도를 얻을 수 있는 13개 속성(NSL-KDD 데이터 셋은 41개의 속성을 제공)을 선택하였다. 또한 축소된 속성만을 사용하여 사이킷런이 제공하는 지니 불순도 (Gini impurity)[14]에 의한 DT 모델링을 통해 KDDTest+에 대해 84.04% 그리고 KDDTest-21에 대해 70.03%의 공격 검출 정확도를 제공한다.

서론에 이어, 제2장에서는 본 논문에서 사용하는 NSL-KDD 데이터 셋의 종류와 속성 값을 기술한다. 제3장에서는 본 연구에서 사용하는 DT와 이를 활용한 네트워크 비정상 검출을 위한 기존 연구 결과와 문제점들을 간략히 알아본다. 제4장에서는 텐서플로우(TensorFlow)[15,16]와 사이킷런을 이용해 속성 축소와 DT 모델링을 진행하는 과정을 설명하고, NSL-KDD 시험 데이터 셋을 대상으로 정확도

(accuracy), 정밀도(precision), 재현율(recall) 그리고 F1-점수(F1 score)를 알아봄으로써 DT 활용에 따른 네트워크 비정상 검출 가능성을 보인다. 마지막으로 제5장에서 결론 및 향후 연구 방향에 대해 기술하였다.

II. NSL-KDD 데이터 셋

기계학습 혹은 인공신경망을 이용한 딥러닝(deep learning) 기술개발에 있어, 적절한 훈련 및 시험 데이터 셋을 사용하는 것은 매우 중요하다. NSL-KDD 데이터 셋은 네트워크 비정상 검출 알고리즘 혹은 IDS/IPS 성능을 평가할 때 전 세계적으로 가장 많이 사용되는 것으로 알려져 있다[7,8,9]. 본 장에서는 NSL-KDD 훈련 및 시험 데이터 셋의 주요 특징, 즉 데이터 셋의 종류, 지도학습에 사용되는 공격 유형, 그리고 학습에 사용될 트래픽 속성 등을 기술한다.

2.1 데이터 셋의 종류

NSL-KDD 데이터 셋은 KDD Cup 1999 데이터 셋 내 중복되어 존재하는 레코드(records) (전체 레코드의 70%)를 제거한 새로운 데이터 셋이다. 이러한 중복된 레코드들은 학습 알고리즘이 자주 발생되는 레코드에 편향(bias)되어 훈련과정에서 자주 발생되지 않은 레코드들을 학습하지 못하게 한다. 뿐만 아니라, 시험 데이터 셋에 중복된 레코드들로 인해 실제 성능을 과대평가하게 된다[7]. 따라서 NSL-KDD 데이터 셋 이전 버전을 사용한 기존 연구결과들은 실제 성능을 과대평가한 결과가 된다.

NSL-KDD 데이터 셋은 두 가지 유형의 훈련 및 시험 데이터 셋을 제공한다. KDDTrain+는 공격 유형 레이블을 포함한 전체 훈련 데이터 셋이고 KDDTrain-는 KDDTrain+의 20%를 포함하는 훈련 데이터 셋이다. 본 논문에서는 전체 훈련 데이터 셋 KDDTrain+을 사용하였다. 한편, KDDTest+은 공격유형 레이블을 포함한 전체 시험 데이터 셋이고 KDDTest-21는 KDDTest+에서 난이도 수준 21을 제거한 시험 데이터 셋이다. 난이도 수준 21은 공격 검출이 가장 쉬운 레코드를 의미한다. 따라서 KDDTest-21 시험 데이터 셋을 사용하여 제안한 모델의 성능을 평가하면 전체 시험 레코드를 포함하고 있는 KDDTest+ 시험 데이터 셋을 사용한 성능 결과보다 낮아질 것이다.

Table 1. Details of normal and attack data of NSL-KDD data set

	KDD Train+	KDD Test+	KDD Test-21
Attacks	58,630	12,833	9,698
Normal	67,343	9,711	2,152
Total	125,973	22,544	11,850

NSL-KDD 훈련 데이터 셋 내에는 neptune, mscan 그리고 mailbomb 등과 같은 24개 공격 타입을 포함하고 있으며 시험 데이터 셋 내에는 추가적으로 14개의 새로운 공격 타입이 포함되어 있다. 따라서 네트워크 비정상 검출 성능을 확인하기 위해서는 반드시 시험 데이터 셋을 사용해야 한다. 즉 훈련 데이터의 일부를 시험 데이터로 재사용하면 이미 학습한 레코드에 대한 분류 결과임으로 당연히 높은 검출 성능을 보일 것이다. 그럼에도 불구하고, 몇 몇 기존 연구들은 훈련 데이터 셋의 일부를 이용해 자신의 모델을 평가하는 오류를 범하고 있다. 이들 데이터 셋에는 다음과 같이 크게 네 가지 종류의 공격 유형이 존재한다. (i) 서비스거부(DoS: Denial of Service) 공격: 컴퓨팅 혹은 네트워크 자원을 고갈시키는 공격, (ii) R2L(Remote to Local) 공격: 비밀번호 추측과 같은 원격 시스템으로부터의 비인증 접속 시도 공격 등, (iii) U2R(User to Root) 공격: 버퍼 오버 플로우(buffer overflow) 공격과 같은 로컬 수퍼유저 (root) 권한의 비인증 접속 공격 등, 그리고 (iv) 프로빙(Probing) 공격: 호스트 혹은 포트 스캔 공격이 있다. 다음 Table 1.은 각 데이터 셋 내 정상 및 공격 레코드 수를 보여준다.

2.2 레코드 속성 값

NSL-KDD 데이터 셋의 각 레코드는 범주형(categorical) 혹은 숫자형(numerical) 값을 가진 41개의 속성(feature 혹은 attribute)을 가진다. 이들 속성들은 본 논문에서 DT 모델의 입력 벡터 값으로 사용되며 다음과 같이 크게 세 가지 그룹으로 분류할 수 있다.

(i) 기본(Basic) 속성: TCP/IP 연결 정보로부터 직접 추출할 수 있는 속성. 예를 들어, Table 2.에서 연결지속시간(duration), 프로토콜 타입(protocol_type) 혹은 서비스 유형(service) 등이

기본 속성에 속한다.

(ii) 트래픽(Traffic) 속성: 윈도우 구간(window interval)을 기준으로 계산되는 속성. 이때, 윈도우 구간은 2초 시간 구간을 갖는 시간 기준 윈도우(time-based window)와 100개의 연결을 기준으로 하는 연결 기준 윈도우(connection-based window)가 있다. 한편 각 윈도우 내에 타겟 연결(해당 레코드가 정상인지 비정상인지를 분류하는 대상 연결)과 동일한 목적지 호스트를 갖는 연결들에 대한 각종 통계치를 계산한 "동일 호스트(same host)" 속성과 현재 연결과 동일한 서비스를 갖는 연결들에 대한 통계치를 계산한 "동일 서비스(same service)" 속성을 갖는다. 따라서 트래픽 속성은 4개의 조합 - TH: 시간기준 윈도우의 동일 호스트 속성, TS: 시간기준 윈도우의 동일 서비스 속성, CH: 연결 기준 윈도우의 동일 호스트 속성, CS: 연결 기준 윈도우의 동일 서비스 속성 - 을 고려하게 된다. 예를 들어, Table 2.에서 count는 지난 2초 동안에 동일한 목적지를 가지는 연결 수, srv_count는 지난 2초 동안에 동일한 서비스를 가지는 연결 수, dst_host_count는 과거 100개의 연결 중에 동일한 목적지를 가지는 연결 수, 그리고 dst_host_srv_count는 과거 100개의 연결 중에서 동일한 서비스를 가지는 연결 수를 각각 나타낸다.

(iii) 콘텐츠(Content) 속성:DoS 혹은 프로빙 공격과는 달리 R2L 과 U2R 공격은 연속적으로 빈번하게 발생되는 공격 패턴을 보이지 않는다. 즉 DoS 혹은 프로빙 공격은 짧은 시간에 많은 연결이 존재하는 반면, R2L 과 U2R 공격들은 패킷의 데이터 영역에 공격 정보가 존재한다. 이러한 공격을 검출하기 위해서는 데이터 영역의 의심스러운 행위를 나타내는 속성이 필요하다. 이러한 종류의 속성을 콘텐츠 속성이라고 한다. 예를 들어, Table 2.에서 num_failed_logins는 로그인 시도 실패 횟수를 나타낸다.

Table 2.는 NSL-KDD 데이터 셋의 41 속성 중 일부를 보여준다. 표의 가장 오른쪽 칼럼은 해당 속성이 속한 카테고리를 나타낸다. 따라서 앞에서 설명한 바와 같이, 동일한 연결 수(number of connections)라고 해도 각각의 속성 카테고리에 따라서 다른 통계치 값을 갖게 된다.

Table 2. Example of NSL-KDD record features(Column 'Cat': B - Basic features, TH - Time-based window of the same host features, TS - Time -based window of the same service features, CH - Connection-based window of the same host features, CS - Connection-based window of the same service features)

Feature	description	Cat
duration	number of seconds of the connection	B
protocol _type	type of protocol, e.g. tcp, udp, icmp	B
service	network service on the destination, e.g. http, telnet	B
count	number of connections	TH
serror	% of connections that have "SYN" errors	TH
srv_count	number of connections	TS
srv_serror_rate	% of connections that have "SYN" errors	TS
dst_host _count	number of connections	CH
dst_host _serror_rate	% of connections that have "SYN" errors	CH
dst_host _srv_count	number of connections	CS
dst_host _srv_serror_rate	% of connections that have "SYN" errors	CS
num_failed_logins	number of failed login attempts	C
num_outbound_cmds	number of outbound commands in an ftp session	C

III. 기존 연구

서론에서 언급한 바와 같이 기계학습 및 딥러닝 기술을 활용한 네트워크 비정상 탐지 연구는 매우 다양하게 진행되어 왔다[17]. 하지만, 본 장에서는 본 논문과 직접적인 연관이 있는 DT를 기반으로 진행된 기존의 네트워크 비정상 탐지 기술과 기계학습의 과대적합 문제 해결을 위한 속성 축소 기술에 대해서

만 언급한다.

3.1 의사결정나무(DT: Decision Tree)

의사결정나무(DT)는 의사결정 규칙(decision rule)을 나무구조로 시각화하여 관심대상이 되는 집단을 몇 개의 소집단으로 분류(classification)하거나 예측(prediction)을 수행하는 분석방법이다. Fig.1.에서 보는 바와 같이, DT는 여러 선택(alternatives) 중 하나를 선택하는 가지 노드(branch node)와 해당 입력 레코드의 레이블(혹은 클래스)를 표현하는 잎새 노드(leaf node)로 구성된다.

패턴 혹은 특징 벡터를 표현하는 q-차원 벡터 $X=(x_1, x_2, \cdots, x_q)$ 와 X의 클래스 레이블을 나타내는 벡터 Y를 정의하면, DT를 활용하여 관찰된 X를 기준으로 $Y=(y_1, y_2, \cdots, y_p)$ 를 예측하는 것이 최종 목표가 된다. 이때 DT를 이용한 분류법은 다음과 같다.

(i) 벡터 X 내 하나의 독립변수를 선택하고 그 독립변수에 대한 기준 값을 정한다.

(ii) 전체 학습 데이터 집합을 선택된 독립 변수 값의 기준 값보다 작은 데이터 그룹과 큰 데이터 그룹으로 나눈다.

(iii) 각각의 자식 노드에 대해 (i),(ii) 단계를 반복하여 하위 자식 노드를 생성한다. 자식 노드에 한 가지 클래스의 데이터만 존재하면 더 이상 자식 노드를 나누지 않고 중지한다.

DT는 나무 구조(독립변수 선택)를 만들고 이들 노드의 기준 값을 정하기 위한 매우 다양한 분류규칙 알고리즘이 있다[5]. Ross Quinlan에 의해 개발된 ID3(Iterative Dichotomiser 3)는 범주형

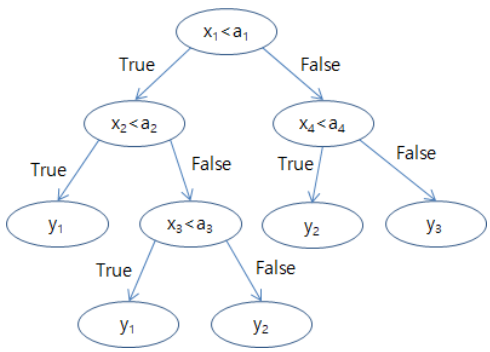

Fig. 1. Simple example of Decision Tree

(categorical) 목표 값에 대해 가장 큰 정보이득(information gain)을 만드는 범주형 속성을 각 노드에 배치하는 다중경로(multiway) 나무를 만든다. 속성(feature)들이 반드시 범주형 이어야만 가능한 ID3의 단점을 극복하기 위해 C4.5는 연속적인 속성 값을 이산적인(discrete) 구간으로 나누어 처리하였다. 또한 C4.5는 if-then 규칙의 나무구조를 생성하고 각 규칙의 정확도를 계산하여 적용될 순서를 결정한다. CART(Classification and Regression Tree)는 C4.5와 매우 유사하며 분류와 예측(회귀 regression) 모두 제공한다. 결국 CART는 속성과 기준 값을 사용해 이진 나무(binary tree)를 생성하며 각 노드는 최대 정보이득을 만들어 낸다. 본 연구에서 사용할 사이킷런의 DT (클래스명: DecisionTreeClassifier[14])는 CART 알고리즘을 사용하고 있으며 범주형 변수는 지원하지 않는다. 따라서 본 연구에서는 데이터 전처리 과정을 통해 범주형 변수 값들은 이산적인 정수 값으로 변경하여 사용한다.

3.2 의사결정나무(DT: Decision Tree)를 이용한 네트워크 비정상 검출 기법

DT는 분류(classification) 문제를 해결하기 위해 가장 대표적으로 사용되어 온 데이터 마이닝 기술 중 하나다[3,4,5,6,7]. DT는 마케팅, 여론 조사, 과학적 발견 등 여러 분야에 걸쳐 광범위하게 사용되어 왔다. 이러한 DT 분류 문제를 네트워크 비정상 탐지에 적용한 다수의 연구들이 존재한다. J. Lee et al.[5]은 DT 기술을 사용한 네트워크 비정상 검출 가능성을 KDD 데이터 셋을 활용하여 제시하였다. 이들은 ID3 DT 알고리즘을 이용하였으며, 데이터 셋에서 제공하는 네 가지 공격타입별로 DT를 생성하여 공격을 검출하였다. 훈련 데이터 셋과 동일한 공격에 대한 재현율(recall)이 77.6%, 그리고 훈련 데이터 셋과 다른 공격에 대한 재현율이 55.5%의 정확도를 보였다. 또한 각각의 공격타입 별로 재현율을 제시하였다. 하지만, 제 2장에서 언급한 바와 같이, KDD 데이터 셋 내에는 많은 중복된 레코드들이 포함되어 있어 오늘날 대부분의 연구들은 NSL-KDD 데이터 셋을 사용할 것을 강력히 권고하고 있다. 한편, 앞에서 언급한 바와 같이 이들은 공격타입별로 별도의 DT를 생성하고 각 공격타입별로 시험을 실시하였다. 하지만 실제 네트워크 환경에

서는 네트워크 비정상 타입을 사전에 알 수 없기 때문에 이들 각각의 DT의 결과를 합성할 수 있는 앙상블 알고리즘이 추가로 필요하다.

Hota와 Shrivas[6]는 C4.5를 포함한 다양한 DT 알고리즘과 Info Gain을 포함한 다양한 속성 선택 알고리즘을 사용하여 KDDTrain+ 데이터 셋에 대한 정확도를 제시하였다. 이들은 KDDTrain+ 데이터 셋을 분리하여 훈련과 시험을 진행하였으며 99.56%의 정확도를 보였다. 하지만, 학습에 사용된 동일한 유형의 공격만을 포함하고 있는 데이터 셋을 대상으로 제안 모델의 정확도를 검증한다는 것은 기계학습 혹은 딥러닝을 이용하여 네트워크 비정상 검출 성능을 평가하는 방법으로는 부적합하다. 따라서 유사한 공격과 새로운 타입의 공격을 포함하고 있는 KDDTest+와 KDDTest-21을 대상으로 모델의 정확도를 검증하여야 한다. Tavallaee et al.[7]는 C4.5 DT알고리즘을 사용하여 KDDTest+에 대해 81.05%와 KDDTest-21에 대해 63.97%의 정확도(accuracy)를 제공하였다. 한편, Lee at al.[5]는 네 가지 공격 유형별로 분리된 ID3 알고리즘을 사용한 DT를 각각 생성하고 각 공격 유형별로 검출 성능에 대한 재현율(recall)을 제시하였다. 그러나 이들 연구에는 두 가지 문제점이 있다. 첫째, 실제 네트워크 환경에서는 공격 유형을 사전에 확인할 수 없기 때문에 이들 네 가지 분리된 DT를 조합해 사용할 수 있는 방법이 필요하며 그 결과를 이용해 최종 결정할 수 있는 앙상블 알고리즘이 필요하다. 두 번째 문제점은, 본 논문에서 사용한 NSL-KDD 데이터 셋 이전 버전의 DARPA 데이터 셋을 사용하여 실험을 진행함으로써 2.1절에서 설명한 바와 같이 성능 결과를 과대평가하게 된다.

3.3 속성 선택(Feature Selection)

속성 선택은 훈련 데이터 셋의 차원(dimension)을 축소하여 잡음 영향을 감소시킴으로서 데이터 처리속도를 향상시키고 궁극적으로 분류의 성능을 증가시킨다[10,11,12]. 따라서 주어진 속성 중에서 일부 최적의 속성을 선택하고 나머지는 제거함으로서 분류의 정확도와 같은 성능을 극대화 하는 것이다. 특히, DT 모델의 경우 훈련 데이터 셋에 과대적합되는 경향이 있으며 이를 해소하기 위한 한 가지 방법으로 속성 선택을 통해 속성을 축소하여 모델에 적용한다.

Zainal et al.[12]은 IDS/IPS에 사용되는 데이터 특징들 중에서 일부 특징들만 IDS/IPS 설계에 효율적이며 이외의 특징들은 부수적이며 별다른 영향을 미치지 않는다고 주장하였다. 이들은 Rough Set 이론을 사용하여 KDD 데이터 셋 중 중요한 특징들을 선택하고 그 결과를 제시 하였다. 한편, Hota and Shrivas[6]은 앞에서 언급한 바와 같이 정보이득과 같은 다양한 속성 선택 기법을 적용하여 NSL-KDD 데이터 셋에 적용한 결과를 제시하였다.

IV. 텐서플로우와 사이킷런(Scikit-Learn)을 활용한 DT 모델링

본 논문에서는 오픈 소스 기계학습 플랫폼을 제공하는 텐서플로[15,16]와 다양한 기계학습 관련 라이브러리를 제공하는 사이킷런[13,14]을 사용하여 DT를 모델링하였다. Fig.2.는 DT를 생성하는 전체 과정을 보여준다.

Fig. 2. Process to make the Decision Tree

4.1 전처리 과정

NSL-KDD 데이터 셋의 레코드들은 41개의 속성과 40개의 공격타입을 레이블 값으로 가지고 있다. 본 논문에서 사용하는 사이킷런의 주요 라이브러리는 범주형 자료를 처리하지 못한다. 따라서 Fig.2.의 전 처리 과정(pre-processing)을 통해 범주형 타입의 속성들(protocol_type, service, flag)을 숫자(int) 타입으로 대치한다. 또한 본 논문에서는 공격 타입 혹은 공격 그룹별로 분류(multi-variate classification)하는 것이 아니라, 정상과 공격 여부만 검출하는 2진 분류(binary classification)를 목표로 한다. 따라서 전 처리 과정을 통해 NSL-KDD 데이터 셋의 공격 타입 레이블을 정상(normal)과 공격(attack)만을 나타내는 이진 레이블로 변환한다.

4.2 속성 선택 과정

DT는 과대적합(over-fitting)하는 단점을 보이

는 것으로 보고되고 있다. 즉 훈련 데이터 셋에 지나치게 잘 맞지만 일반성이 떨어져 실제 시험 데이터 셋에서 우리가 원하는 성능을 얻지 못하는 경향이 있다. 본 논문에서는 사이킷런 feature_selection 서브 모듈의 SelectKBest 클래스[13]를 사용하여 실제 훈련 데이터 셋의 41개 속성과 레이블 사이 카이-제곱 테스트를 사용해 가장 높은 점수를 가진 k개의 속성을 선택한다. Fig.3.은 최적의 속성을 선택하기 위한 흐름도를 보여준다. 이때 훈련 데이터 셋을 사용하여 k를 2부터 연속적으로 증가시키면서 SelectKBest를 사용하여 41개 속성 중 k개의 속성을 선택(Fig.3.의 (3))하고 선택된 속성을 기준으로 훈련 및 시험 데이터 셋을 재정리(Fig.3.의 (4))한다. 한편 축소된 속성이 반영된 훈련 데이터 셋을 사용하여 DT 모델을 생성(Fig.3.의 (5))하고 시험 데이터 셋을 사용해 생성된 DT 모델의 정확도를 측정(Fig.3.의 (6))한다. 만약 축소된 속성이 반영된 새로운 데이터 셋으로 만들어진 DT 모델의 정확도가 이전 정확도와 비교해 향상(Fig.3.의 (7))되었다면 축소된 속성을 업데이트(Fig.3.의 (8))하여 최종적으로는 최적의 속성을 선택하게 된다.

Fig.3.의 (3)에서, 속성 선택 과정을 위해 사이킷런이 제공하는 SelectKBest는 score 함수를 인자로 사용한다. 본 연구에서는 각각의 속성과 레이블 사이 카이-제곱 값을 계산하는 score 함수를 사용하여 큰 값을 가지는 상위 k개의 속성을 선택하게 된다. 즉 임의의 하나의 속성과 클래스 사이 계산된 카이-제곱 값이 작다는 것은 해당 속성이 클래스에 비교적 독립적이라는 것을 의미한다.

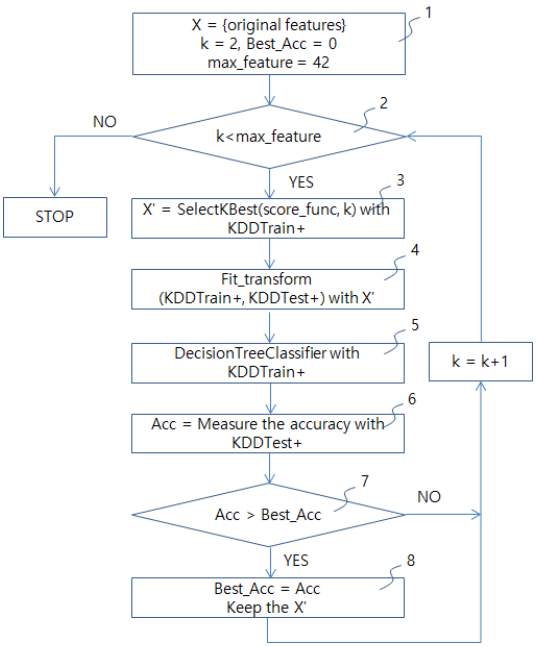

Fig. 3. Process to select the optimal features

Fig.4.와 Fig.5.는 Fig.3.의 (3)~(6) 과정에서 속성 선택 개수 k 값에 따른 DT 모델링을 통해 측정된 KDDTest+와 KDDTest-21에 대한 공격 검출 정확도를 각각 보여준다. 그림에서 보듯이, 속성 개수 k값의 변화에 따라 모델의 정확도가 다양하게 변화하고 있으며 최대 약 8% 차이를 보인다. 따라서 우리가 이미 예상한 바와 같이 DT 모델은 시험 데이터 셋에 과대적합되는 경향이 있으며, 속성 축소를 통해 모델의 정확도를 향상 시킬 수 있게 된다. 특히, k=13에서 최고의 공격 검출 정확도를 확인할 수 있다. 한편, Fig.4.와 Fig.5.를 비교하면 KDDTest-21에 대한 정확도가 KDDTest+와 비

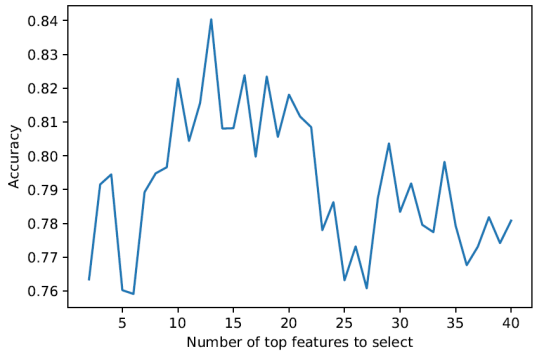

Fig. 4. Accuracies of DT model according to the number of feature selection on the KDDTest+ data set

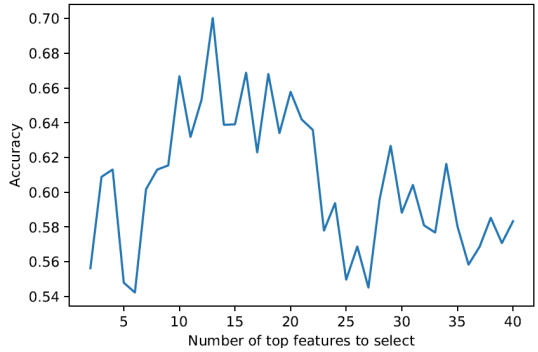

Fig. 5. Accuracies of DT model according to the number of feature selection on the KDDTest-21 data set

교해 20% 감소하였다. 이러한 정확도 감소는 KDDTest-21 데이터 셋은 KDDTest+ 데이터 셋에서 검출이 가장 쉬운 공격 레코드들이 제거되었기 때문에 우리가 예상한 결과이다. 그러나 이들 두 그림을 비교하면 정확도에 대한 값만 차이가 나고 전체적인 정확도 변화의 패턴은 거의 동일한 것으로 조사되었다. 이러한 결과는 가장 쉽게 검출되는 공격 레코드들은 DT 모델에 의해 모두 검출되기 때문이다.

한편, 본 논문에서는 제 2.2절에서 설명한 41개 속성 (속성 그룹 B(9개), C(13개), TH(5개), TS(4개), CH(5개), 그리고 CS(5개)) 모두를 고려한 DT 모델을 DT-41이라고 칭하고 선택된 13개의 속성 값을 사용한 DT 모델을 DT-13이라고 부른다. 선택된 13개 속성(Table 3.)을 살펴보면, 여섯 가지 속성 그룹 B(5개), C(1개), TH(2개), TS(1개), CH(2개), 그리고 CS(2개)에 속한 속성들이 비교적 균일하게 반영되어 있음을 확인할 수 있다.

Table 3. The 13 significant features selected by feature selection(Column 'Cat': B - Basic features, TH - Time-based window of the same host features, TS - Time -based window of the same service features, CH - Connection-based window of the same host features, CS - Connection-based window of the same service features)

Feature	description	Cat
duration	number of seconds of the connection	B
service	network service on the destination, e.g. http, telnet	B
flag	normal or error status of the connection	B
src_bytes	number of data bytes from source to destination	B
dst_bytes	number of bytes from destination to source	B
logged_in	1 if successfully logged in; 0 otherwise	C
count	number of connections	TH
serror_rate	% of connections that have "SYN" errors	TH
srv_serror_rate	% of connections that have "SYN" errors	TS
dst_host_count	number of connections	CH
dst_host_srv_count	number of connections	CS
dst_host_serror_rate	% of connections that have "SYN" errors	CH
dst_host_srv_serror_rate	% of connections that have "SYN" errors	CS

4.3 모델 검증

DT 모델링에 대한 분류 성능을 검증하기 위해 오차행렬(confusion matrix)을 통해 정밀도(precision), 재현율(recall), 정확도(accuracy), 그리고 정밀도와 재현율의 조화평균인 f1-점수(f1-score) 등을 활용할 수 있다. 이들 각각의 성능지표는 오차행렬(confusion matrix) Table 4.로부터 다음과 같이 계산할 수 있다.

$$\text{정밀도} = \frac{TP}{TP+FP}$$

$$\text{재현율} = \frac{TP}{TP+FN}$$

$$\text{정확도} = \frac{TN+TP}{TN+FP+FN+TP}$$

$$f1-\text{점수} = \frac{TP}{TP+(FN+FP)/2}$$

Table 4.의 오차행렬에서, TN(True Negative)은 레이블이 정상으로 표시된 레코드를 DT 모델이 정상으로 제대로 예측한 경우, FP(False Positive)는 레이블이 정상으로 표시된 레코드를 DT 모델이 공격으로 잘못 예측한 경우, FN(False Negative)은 레이블이 공격으로 표시된 레코드를 DT 모델이 정상으로 잘못 예측한 경우, 그리고 TP(True Positive)는 레이블이 공격으로 표시된 레코드를 DT 모델이 공격으로 제대로 예측한 경우를 각각 나타낸다.

Table 5.와 Table 6.은 NSL-KDD 데이터 셋의 41개 속성 모두 사용한 DT 모델링(DT-41)을

Table 4. Confusion matrix

Label \ Prediction	Normal	Attack
Normal	True negative(TN)	False Positive(FP)
Attack	False negative(FN)	True Positive(TP)

통해 NSL-KDD KDDTest+ 및 KDDTest-21 시험 데이터 셋에 대한 성능 결과를 보여준다. KDDTest+에 대한 f1-점수는 0.79 그리고 KDDTest-21에 대한 f1-점수는 0.64를 보여준다. 이러한 결과는 제2장에서 설명한 바와 같이 KDDTest-21 시험 데이터 셋 내에는 상대적으로 검출하기 어려운 레코드들이 존재하기 때문이다. 한편, Table 5.와 Table 6.으로부터 정상 레코드에 대한 정밀도와 공격 레코드의 재현율이 상대적으로 낮게 조사되었다.

Table 7.과 Table 8.은 제4.2절에서 선택한 13개 속성을 사용한 DT 모델링(DT-13)을 통해 NSL-KDD 시험 데이터 셋에 대한 성능 결과를 보여준다. 41개 속성을 모두 고려한 DT 모델(Table 5.와 Table 6.)과 비교하면 KDDTest+에 대해 f1-점수가 약 4%, 그리고 KDDTest-21에 대해 9% 향상되었음을 확인할 수 있다.

한편, Table 9.는 DT-41과 DT-13 각각에 대한 KDDTest+와 KDDTest-21에 대한 정확도를 보여준다. 속성 선택이 반영된 DT-13의 경우, 기존의 DT에 대한 연구결과[7]와 비교해 보면 훈련 데이터 셋 KDDTest+에 대해 약 3%, 그리고 KDDTest-21에 대해 약 6% 정도 정확도가 향상되었다. 특히, 검출하기 어려운 훈련 데이터 셋에 대해 더 많은 정확도 향상을 확인할 수 있다.

Table 10.은 제2.1절에서 설명한 네 가지 공격 유형별(DoS, R2L, U2R, Probe)로 재현율을 각각 조사하였다. 표에서 보듯이, U2R 공격 유형에 대한 재현율이 극히 낮게 조사되었다. 결국 U2R과 같이 호스트 내부적으로 일어나는 공격 패턴을 네트워크 비정상 검출용 DT 모델링을 통해 검출하는 것은 매우 어려운 작업이 될 것이다.

Lee et al.[5]는 이들 네 가지 공격 유형별로 분리된 ID3 알고리즘을 사용한 DT를 각각 생성하고 각 공격 유형별로 검출 성능에 대한 재현율을 제시하였다. 한편, 이들은 Lee et al.[18]가 제시한 침입검출 모델과 성능비교를 하기 위해 본 논문에서 사용한 NSL-KDD 이전 버전의 DARPA 데이터 셋을 사용하였으며 성능평가를 위해 시험 데이터 셋을 Old와 New로 구분하고, New는 훈련 데이터 셋에 존재하지 않는 레코드로 구성하였다. 따라서 본 논문에서 제시한 Table 10.과 참고문헌 [5]가 제시한 성능을 직접 비교하는 것은 불가능하다. 다만, NSL-KDDTest+ 내에는 훈련 데이터 셋에 존재하지 않는 14개 타입의 공격이 존재하기 때문에 Table 10.의 결과와 참고문헌 [5]의 Old 와 New의 평균값과 간접적으로 비교할 수 있다. Table 11.에서 보듯이, 참고문헌 [5]에서 제시한 R2L과 U2R의 재현율이 본 문문에서 제시한 DT 모델 결

Table 5. Performance of DT-41 on KDDTest+ data set

Cat	Precision	Recall	f1-score
Normal	0.68	0.97	0.80
Attack	0.97	0.65	0.78
Avg	0.84	0.79	0.79

Table 6. Performance of DT-41 on KDDTest-21 data set

Cat	Precision	Recall	f1-score
Normal	0.29	0.87	0.44
Attack	0.95	0.54	0.69
Avg	0.83	0.60	0.64

Table 7. Performance of DT-13 on KDDTest+ data set

Cat	Precision	Recall	f1-score
Normal	0.75	0.95	0.84
Attack	0.95	0.76	0.84
Avg	0.87	0.84	0.84

Table 8. Performance of DT-13 on KDDTest-21 data set

Cat	Precision	Recall	f1-score
Normal	0.36	0.81	0.49
Attack	0.94	0.68	0.79
Avg	0.83	0.70	0.73

Table 9. Comparing accuracies of DT models

Cat	DT-41	DT-13	[7]
KDDTest+	78.85	84.04	81.05
KDDTest-21	59.78	70.03	63.97

Table 10. Recalls of DT-13 according to each attack types on NSL-KDD test data set

Category	KDDTest+	KDDTest-21
DoS	88.32	79.96
R2L	33.58	33.58
U2R	19.5	19.5
Probe	88.68	88.59

과보다 우수함을 볼 수 있다. 이것은 참고문헌 [5]의 경우, 앞에서 언급한 바와 같이 각 공격 타입별로 별도의 DT를 생성하여 검출한 결과이며 또한 데이터 셋 역시 중복된 레코드가 다량으로 존재하여 성능을 과대평가하기 때문이다. 실제 네트워크 비정상 검출 환경에서는 해당 레코드가 사전에 어떤 공격 타입인지 알 수 없으며 따라서 공격 타입별로 생성된 DT 모델을 적용하기 위해서는 각각의 DT 모델을 통합하여 결과를 추출할 수 있는 앙상블 알고리즘이 반드시 필요하다.

Table 11. Performance (recalls) comparison

Category	KDDTest+	[5]
DoS	88.32	71.3
R2L	33.58	51.15
U2R	19.5	58.7
Probe	88.68	78.45

V. 결 론

본 논문은 네트워크 비정상 탐지를 위해 속성 축소를 고려한 의사결정나무를 활용하는 모델을 제시하고 NSL-KDD 데이터 셋을 사용하여 제안 모델의 공격 검출 성능을 검증하였다. 제안된 모델은 의사결정나무의 단점인 훈련 데이터 셋에 과대적합되는 문제점을 해소하기 위해 카이-제곱 값을 이용한 속성 축소 방법을 사용하였으며, 그 결과 기존 의사결정나무 기법 결과와 비교해 시험 데이터 셋 KDDTest+에 대해 약 3% 그리고 KDDTest-21에 대해 약 6% 정도 정확도가 향상되었다. 특히, 검출하기 어려운 훈련 데이터 셋에 대해 더 높은 정확도 향상을 확인할 수 있었다. 뿐만 아니라, 네 가지 공격 타입별로 제안 모델의 검출 능력을 살펴본 결과, 서비스거부 공격과 스캐닝 공격에 대한 탐지 능력은 실제 네트워크에 적용할 수 있는 수준인 88% 이상인 것으로 최종 확인되었다.

References

[1] A. Patel, Q.S. Qassim, and C. Wills, "Survey of intrusion detection and prevention systems," Information Management & Computer Security, vol. 18, no. 4, pp. 277-290, Oct. 2010.

[2] M. Ahmed, A.N. Mahmood, and J. Hu, "A survey of network anomaly detection techniques," Journal of Network and Computer Applications, vol. 60, issue C, pp.19-31, Jan. 2016.

[3] S.R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE Trans. on Systems, Man and Cybernetics, vol. 21, no. 3, pp.660-674, June 1991.

[4] C. Kruegel and T. Toth, "Using decision trees to improve signature-based intrusion detection," RAID 2003, LNCS 2820, pp.173-191, Feb. 2004.

[5] J. Lee, J. Lee, S. Sohn, J. Ryu, and T. Chung, "Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system," Proceedings of the 10th International Conference on Advanced Communication Technology, pp. 1170-1175, Feb. 2008.

[6] H. Hota and A.K. Shrivas, "Decision tree techniques applied on NSL-KDD data and its comparison with various feature selection techniques," Advanced Computing, Networking and Informatics - Volume 1 Proceedings of the Second International Conference on Advanced Computing, Networking and Informatics, pp. 205-211, 2014.

[7] M. Tavallaee, E. Bagheri, W. Lu, and A.A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," Proceedings of 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1-6, July 2009.

[8] KDD Cup 1999 Data, Available on: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

[9] NSL-KDD dataset, Available on:

https://www.unb.ca/cic/datasets/nsl.html, March 2009.
〔10〕 K. Cios, R.W. Swiniarski, and W. Pedrycz, Data mining methods for knowledge discovery, 3rd Ed., Kluwer Academic Publishers, 2000.
〔11〕 S. Chebrolu, A. Abrahan, and J.P. Thomas, "Feature deduction and ensemble design of intrusion detection system," Journal of Computers and Security, vol. 24, issue 4, pp. 295-307, June 2005.
〔12〕 A. Zainal, M.A. Maarof, and S.M. Shamsuddin, "Feature selection using rough set in intrusion detection," TENCON 2006 - 2006 IEEE Region 10 Conference, pp. 1-4, Dec. 2006.
〔13〕 SelectKBest, Available on: https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
〔14〕 DecisionTreeClassifier, Available on: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html
〔15〕 Get started with TensorFlow, https://www.tensorflow.org
〔16〕 A. Geron, Hands-On Machine Learning with Scikit-Learn & TensorFlow, O'REILLY, 2017
〔17〕 D. Kwon, J. Kim, H. Kim, and S. Cuh, "A survey of deep learning-based network anomaly detection," Cluster Computing Journal, Springer, pp. 1-13, 2017.
〔18〕 W. Lee, S.J. Stolfo, and K.W. Mok, "A data mining framework for building intrusion detection models," Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp. 120-132, May 1999.

〈저자소개〉

강 구 홍 (Koohong Kang) 정회원
1985년 8월: 경북대학교 전자공학과 졸업
1990년 2월: 충남대학교 전자공학과 석사
1998년 2월: 포항공과대학교 전자계산학과 박사
1985년 9월~1999년 3월: 한국전자통신연구원 선임연구원
2008년 1월~2009년 2월: Purdue University Visiting Scholar
2016년 1월~2017년 2월: 제주대학교 방문교수
2000년 9월~현재: 서원대학교 정보통신공학과 교수
〈관심분야〉 성능분석, 네트워크 보안, 머신 러닝