JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 23, No. 9, pp. 1803-1812, Sep. 2022

Check for updates

연관 규칙 기반 네트워크 공격 탐지 및 분석 기법 연구

장 영 인[1] · 최 영 준[2*]
[1]아주대학교 컴퓨터공학과 박사과정
[2*]아주대학교 컴퓨터공학과 교수

A study of network attack detection and analysis based on association rule

Young-In Jang[1] · Young-June Choi[2*]

[1]Doctor's Course, Department of Computer Engineering, Ajou University, Suwon, 16499, Korea
[2*]Professor, Department of Computer Engineering, Ajou University, Suwon, 16499, Korea

[요 약]

최근 네트워크 공격 탐지 방안을 위해 머신러닝 모델을 활용하는 사례가 증가하고 있다. 특히, 정상과 비정상으로 구분하는 비정상 탐지 모델이 주류를 이루고 있으나, 이진 분류 및 다중클래스 분류 모델은 다양한 공격을 분류하기 위해서는 모델의 크기가 커지거나 time-complexity가 길어지는 등의 오버헤드가 생기게 된다. 연관 규칙 분석은 데이터 안의 숨겨진 패턴을 찾을 수 있으므로, 이점에 착안하여 연관 규칙 분석 결과를 사용하여 네트워크 공격을 분류하는 기준을 추출한다. 본 연구에서는 연관 규칙을 기반으로 한 다양한 네트워크 공격을 분석하고, 분석 결과를 바탕으로 입력 데이터와 가까운 공격들을 특정화하여 제공하는 모델을 제안한다. 실험 결과에 따르면 제시한 기법이 높은 탐지율을 보였다.

[Abstract]

These days, the use of machine learning models for network attack detection has been increasing. Especially anomaly detection models that distinguish normal and abnormal are the mainstream, but binary classification and multiclass classification models have some limitations; when a model needs to classify various attacks, the model's size can be huge, or the model will be complex. That causes the time complexity of the model to increase. Since association rule analysis figure out hidden patterns in data, we extract the criteria for classifying network attacks using the result of association rule analysis. Therefore, in this study, we analyze various network attacks based on association rules, and based on the analysis results, we propose a model that provides ranking of attacks close to the input data. The proposed method showed a high detection rate according to the experimental results.

색인어 : 연관규칙 분석, 네트워크 공격 탐지, 네트워크 공격 분석, 네트워크 보안, 네트워크 공격 분류

Keyword : Association rule, Network attack detection, Network attack analysis, Network security, Network attack classification

http://dx.doi.org/10.9728/dcs.2022.23.9.1803

Received 20 July 2022; **Revised** 04 August 2022
Accepted 09 August 2022

***Corresponding Author; Young-June Choi**

Tel: +82-31-219-2634
E-mail: choiyj@ajou.ac.kr

Ⅰ. 서 론

스마트폰, IoT, 스마트시티, 커넥티드 카 및 최근 대두되고 있는 메타버스까지 우리의 삶은 이더넷과 일체화된 삶을 살고 있다고 해도 과언이 아니다. 하지만 우리의 삶이 이더넷과 가까워질수록 우리는 해킹이나 보안 위협상황에 쉽게 노출된다. 이러한 추세에 맞추어 네트워크 공격 탐지 방안에 관한 연구는 점점 더 중요성이 높아지고 있는 시점이다.

기존에 시행되어온 네트워크 공격 탐지 기법은 크게 오용 탐지(misuse detection)와 비정상 탐지(anomaly detection)가 있다. 오용 탐지는 전문가가 찾아낸 패턴을 시스템에 입력해두어 들어오는 데이터와 일치하는 패턴을 탐지하는 기법이다. 하지만 공격을 조금만 우회하더라도 탐지율이 떨어지기 때문에 이를 해결하고자 비정상 탐지 기법이 사용되었다. 비정상 탐지 기법은 주로 시스템 내에서 학습된 패턴과 다른 것으로 식별된 패턴들을 비정상으로 간주하여 침입으로 판단한다.

본 연구에서는 침입을 판단하는 것에서 그치지 않고, 입력된 패킷과 가장 유사한 공격들을 추출하여 제시하는 것을 목표로 한다. 본 연구의 기여는 다음과 같다.

- 연관 규칙 분석을 통해 각 공격별로 특이 특성들을 추출 및 분류하는 기법을 제안한다.
- 값의 변조와 같은 우회 공격에 영향을 받지 않고 특성의 존재 여부에 따라서 공격을 탐지하는 기법을 제안한다.
- 실험 결과를 통해 공격 탐지뿐만 아니라 유사한 공격을 제시함으로써 추가적인 대응에 도움을 줄 수 있음을 보인다.

본 논문의 구성은 2장에서는 비정상 탐지 기법에 관한 관련 연구를 서술하고, 3장에서는 데이터 전처리 및 제안하는 기법에 관해 서술한다. 4장에서는 실험에 사용한 데이터와 실험 결과에 대해 논의하고 5장에서 결론을 서술한다.

Ⅱ. 관련 연구

최신 비정상 탐지 기법들[1]-[11]은 머신러닝 기법을 사용하여 정상과 비정상을 구분한다. 지도학습 기반 탐지 기법을 사용하기 위해서 먼저 데이터에 label로 정상과 비정상이 구분되어 있어야 한다. 또한, 정상과 비정상으로 구분하는 탐지 시스템은 이진 분류(binary classification)모델로 불리고, 정상과 다양한 공격을 구분하는 경우는 다중클래스 분류(multi-class classification)모델이다.

논문[12]은 Random Forest, SVM, Naive Bayes 외 다양한 머신러닝 기반 알고리즘들과 Feature Selection을 통해 성능을 비교하였다.

논문[13]은 Chi-Square와 Information Gain, Recursive Feature Elimination을 사용한 Feature Selection을 SVM, Naive Bayes, Decision Tree, Random Forest 등의 머신러닝 기법과 조합하여 성능을 비교했다.

논문[14]은 네트워크 트래픽의 특성에 따라 분류 성능에 미치는 영향을 알아보기 위해 Convolution Neural Networks 기반 분류 실험을 진행하였다.

논문[15]은 전통적인 머신러닝 기법보다 다중클래스 분류에서 우수한 성능을 보이는 Convolution Neural Networks 기반의 Intrusion Detection Systems를 제안하였다. 논문[16]은 다양한 성능측정을 기반으로 ANN(Artificial Neural Network)을 사용한 모델을 제안하였다. 논문[17]은 여러 개의 이진 분류와 하나의 집계 모듈로 k-NN 알고리즘을 사용하여 2단계 하이브리드 방식을 제안했다.

비정상 탐지 기법에 연관 분석 기법을 도입하여 모델의 성능을 높인 연구들도 있다. 논문[18]은 연관 규칙 분석을 사용하여 비정상으로 잘못 분류된 정상 데이터를 필터링하는 모델을 제안했다. 논문[19]은 연관 규칙 마이닝 및 AdaBoost와 ANN의 하이브리드 모델을 제시했다.

기존 분류모델 중 이진 분류모델은 다양한 공격들을 다루지 못하기 때문에 “정상과 공격1”, “정상과 공격2” 등의 이진 분류기를 여러 개 사용하는 하이브리드 모델을 구현하거나 다중클래스 모델은 클래스가 늘어날수록 성능이 떨어지게 되므로 정상과 몇 가지 공격유형을 분류하는 다중클래스 모델을 사용한다. 하지만 위의 대안들은 모델의 크기가 커서 시스템의 요구사항이 높아지거나 결과가 나오기까지의 실행시간이 늘어나는 등의 오버헤드가 생기게 된다. 연관 분석 기법을 도입한 기존 연구들은 주로 정상 데이터만을 기준으로 하였기 때문에 정상 데이터와 유사한 공격 데이터도 같이 제외되거나 공격 데이터에 대한 분석은 이루어지지 않았다.

이러한 문제점에서 착안하여, 본 연구에서는 네트워크 공격 탐지 및 분석을 위해 연관 규칙 분석 기법을 사용하여 다양한 네트워크 공격에 대한 규칙을 분석하고, 분석된 공격별 규칙을 실제 네트워크 데이터에 적용하여 입력된 네트워크 패킷의 공격을 특정화한 정보를 제공하는 네트워크 공격 분석 및 탐지 기법을 제안한다.

Ⅲ. 본 론

3-1 데이터 전처리

연관 규칙 분석을 위해서는 데이터를 transaction 형태로 변환해야 하므로 우리는 먼저 데이터 feature에 대한 분석을 진행하였다. 우리가 사용한 데이터 안에는 범주형과 수치형 데이터가 혼재되어 있기 때문에, 유형별로 데이터를 분류하는 작업을 거쳤다. 범주형 데이터는 범주값 별로 다른 의미를 지니므로 범주형 데이터를 각각 하나의 feature로 분리하는 one-hot 인코딩 기법을 사용한 후 수치형 데이터와 함께 값의 유무에 따라 T/F 형태로 전환하였다. 최종적으로 T/F 형태로 전환된 데이터를 transaction 형태로 변환하여 연관 규칙 분석에 사용한다.

3-2 연관 규칙 분석

연관 규칙 분석은 장바구니 분석이라는 이름으로도 잘 알려진 일종의 규칙 기반 학습방법으로, 대용량 데이터베이스에서 기존에는 발견할 수 없었던 데이터 속의 숨겨져 있는 패턴이나 규칙을 탐색할 수 있는 장점이 있다. 규칙은 "if X, then Y" 또는 "if X -> Y"의 형식으로 표현되고 연관 규칙은 특정 사건이 발생하였을 때 빈번하게 같이 발생하는 또 다른 사건의 규칙을 말한다. 연관 규칙에서 항목 집합(Item set)이란 전체 item 중에서 가능한 부분 집합을 말하며 항목 집합의 집합(The set of item sets)은 item의 부분 집합들로 구성된 집합을 말한다. 즉, 연관 규칙이란 특정 항목 집합이 발생하였을 때 또 다른 항목 집합이 발생하는 규칙이라고 볼 수 있다. 본 연구에서는 이 점에 착안하여 공격별로 feature들간의 관계가 다르다면, 공격별로 등장하는 feature도 달라질 것이므로, 들어온 패킷의 feature에 따라 공격을 특정할 수 있을 것으로 가정한다.

연관 규칙 분석을 평가할 수 있는 척도로 지지도(Support), 신뢰도(Confidence), 향상도(Lift)가 있다. 지지도는 수식 (1)과 같이 두 항목 X와 Y의 지지도는 전체 transaction 중에서 항목 집합 X와 Y를 모두 포함하는 transaction의 비율을 말한다. 즉 지지도는 빈도나 구성비가 높은 좋은 규칙을 찾거나, 불필요한 연산을 줄이기 위한 기준으로 사용한다. 수식 (2)는 신뢰도를 표현하며 항목 집합 X를 포함하는 transaction 중에서 항목 집합 Y도 포함하는 transaction 비율을 의미한다. 신뢰도가 높을수록 유용한 규칙일 가능성이 높다. 향상도는 항목 집합 X가 주어지지 않았을 때의 항목 집합 Y의 확률 대비 항목 집합 X가 주어졌을 때의 항목 집합 Y의 확률 증가 비율을 뜻하며 수식 (3)으로 표현할 수 있다.

$$\mathrm{Support}(X \rightarrow Y) = P(A \cap B) \tag{1}$$

$$\mathrm{Confidence}(X \rightarrow Y) = P(Y|X) = \frac{P(A \cap B)}{P(A)} \tag{2}$$

$$\mathrm{Lift}(X \rightarrow Y) = \frac{P(A \cap B)}{P(A) \cdot P(B)} \tag{3}$$

데이터 feature의 수가 많을수록 가능한 모든 경우의 수를 탐색하여 만들어지는 연관 규칙은 가장 정확한 결과를 보여주지만, 그 수가 기하급수적으로 증가하게 되므로 연산 시간도 증가하기 때문에 비효율적인 방법이다. 이로 인해 효율적으로 연관 규칙을 생성하기 위한 다양한 알고리즘이 등장하였으며 본 연구에서는 apriori 알고리즘을 채택하였다. Apriori 알고리즘은 최소 지지도 이상의 값을 갖는 항목 집합을 말하며, 빈발 항목 집합으로 불린다. 빈발 항목 집합의 추출을 위한 apriori 알고리즘의 원리는 다음과 같다.

- 한 항목 집합이 빈발하다면 이 항목 집합의 모든 부분 집합은 빈발 항목 집합임
- 한 항목 집합이 비빈발하다면 이 항목 집합을 포함하는 모든 부분 집합 또한 비빈발 항목 집합임

여기서 두 번째 원리를 통해 처음부터 항목 집합을 가지치지(pruning)해서 모든 가능한 항목 집합의 개수를 줄이는 방식을 적용한다.

최종적으로 apriori 알고리즘을 사용하여 최소 신뢰도에 미달하는 연관 규칙을 제외하고 남은 연관 규칙들을 사용한다.

3-3 네트워크 공격 분석

3-2장에서 얻어진 각 공격 별로 연관 규칙들을 하나의 집합으로 통합하면 최대한 많은 유형의 규칙을 포함하며 각 공격에서 주로 등장하는 feature의 집합이 만들어진다. 즉, 본 연구의 목적은 연관 규칙 분석을 통해 각 공격의 특이 feature를 추출하고 이를 이용하여 공격을 특정화하는 것이다.

그림 1은 서로 다른 공격의 특이 feature 집합과 input feature 집합과의 관계를 보여준다. 노란 원은 input 중 한 packet의 feature만 추출한 집합을 의미하며, 위 그림의 (a)부터 (d)의 노란 원은 모두 동일한 하나의 input feature 집합이다. 노란 원을 제외한 나머지는 연관 분석을 통해 추출된 각기 다른 공격의 특이 feature 집합을 의미한다. 즉, 그림 1은 같은 input feature 집합에 대해 각기 다른 공격의 feature 집합과의 교집합 사례이다.

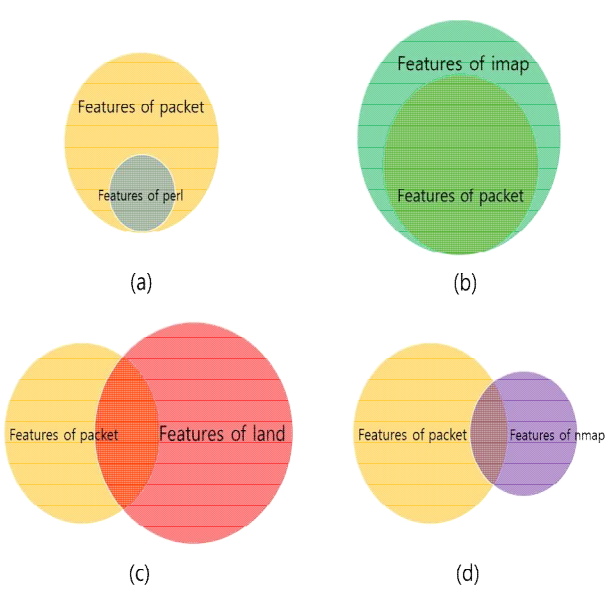

그림 1. 각 공격별 feature 집합과 input feature 집합과의 관계
Fig. 1. Diagram between different attack and input feature sets

그림 1-(a)은 공격 feature 집합의 크기가 input feature 집합보다 작고 모든 feature들이 input feature 집합에 포함되어 있음을 보여주며 그림 1-(b)에서는 그 반대의 경우로 공격 feature 집합이 더 크고 모든 input feature가 공격 feature에 속해있음을 확인할 수 있다. 그림 1-(c)와 (d)에서는 각기 다른 공격 feature 집합과 input feature 집합의 교집합의 크기가 다른 것을 관찰할 수 있다.

하나의 input feature 집합 A와 하나의 특정 공격 feature 집합 B와의 교집합을 $A \cap B$라고 할 때, 그림 1에서 볼 수 있듯이, 각 집합 A와 B에서 $A \cap B$가 차지하는 비율은 서로 다르다. 위 사례에서 착안하여 본 연구에서는 두 종류의 평가기법을 사용하였다.

① 집합 A에서 $A \cap B$가 차지하는 비율
② 집합 B에서 $A \cap B$가 차지하는 비율

①의 경우, 집합 A에서 $A \cap B$가 차지하는 비율이 높을수록 input packet은 집합 B에 해당하는 특정 공격일 가능성이 높다. ②의 경우에는 집합 B에서 $A \cap B$가 차지하는 비율이 높다는 것은 그 공격의 특이 feature들이 많이 포함되어 있다는 의미이다. ①의 평가 기법만을 사용하게 되면, 그림 1-(b)나 (c)의 경우처럼 ①의 비율이 높다고 해서 그 특정 공격의 feature마저 모두 만족시키는지는 알 수 없다. 또한, 그림 1-(a)나 (d)처럼 ②의 비율이 높더라도 input feature 집합의 과반수를 만족하지 않으면 input packet을 대표한다고 볼 수 없다. 그러므로 우리는 공격 분석 및 특정화를 하기 위해 최종적으로는 두 종류 평가기법 결과의 평균을 사용한다.

Ⅳ. 실 험

4-1 데이터셋

본 연구에서 실험을 위해 사용한 데이터셋은 NSL-KDD 데이터셋[20]으로 KDD'99 데이터에서 지적된 고유한 문제를 해결하기 위해 제안되었다. NSL-KDD 데이터셋은 여전히 몇 가지 문제를 내포하고 있지만, 다양한 침입 탐지 방법을 탐색하는데 효과적인 데이터셋이다.

표 1은 NSL-KDD 데이터셋에 대한 요약으로, NSL-KDD 데이터셋은 정상 패킷과 네 가지 서로 다른 유형의 다양한 공격으로 구성되어 있다. 한 유형당 최소 6종류에서 15종류의 공격이 존재하며 데이터셋에는 총 39종류의 공격이 있다. 하지만 모든 종류의 공격이 train과 test 데이터셋에서 등장하는 것이 아니며 test 데이터셋에서 새로이 등장하는 공격들이 존재한다. NSL-KDD의 train 데이터셋에서 등장하는 공격 label은 normal을 포함하여 총 23개가 존재하고 test 데이터셋에서는 normal 포함 총 38개의 공격 label이 등장한다.

표 1. NSL-KDD 데이터셋 요약
Table 1. Summary of NSL-KDD dataset

Class	Distinct attack	Dataset		
		Train	Test	Subtotal
Normal	-	67,343	9,711	77,054
DoS	11	45,927	7,460	53,387
Probe	6	11,656	2,421	14,077
R2L	15	995	2,885	3,880
U2R	7	52	67	119
Total	39	125,973	22,544	148,517

실질적으로 train 데이터셋과 test 데이터셋에서 등장한 공격 label이 겹치는 경우는 총 21종류이다.

본 연구의 목적은 train 데이터셋에서 등장한 공격을 분석한 결과를 통해 test 데이터셋에서 얼마나 정확하게 분류할 수 있는지를 확인하는 것이므로, test 데이터셋에서 새로이 등장하는 공격들은 본 연구에서 다루지 않는다. 그러므로 각 데이터셋에서 공통으로 등장한 21종의 공격 label에 대해서만 실제 네트워크 공격 분석을 진행한다.

4-2 평가 기법

3-3장에서 논의한 두 종류의 평가기법을 수식화하면 다음과 같다.

$$A_{ratio} = \frac{f(A \cap B)}{f(A)} \tag{4}$$

$$B_{ratio} = \frac{f(A \cap B)}{f(B)} \tag{5}$$

$$LabelScore = \frac{A_{ratio} + B_{ratio}}{2} \tag{6}$$

f는 각 집합의 발생 횟수를 계산하는 함수이며 집합 A와 B는 각각 하나의 input feature 집합과 공격별 특이 feature의 집합을 의미한다. 수식 (4)와 (5)를 통해 집합별 비율을 계산하고, 수식 (6)을 사용하여 하나의 input feature에 대한 각각의 공격별 LableScore(LS)를 결정한다. 최종적으로는 LS를 내림차순으로 정렬하고 top 5, top 10에 해당하는 공격명을 추출하여 실제 label이 포함된 경우를 세어 실제 label 등장 횟수와 비교한다.

4-3 실험 결과

1) 공격별 연관 규칙 분석

표 2. 공격 label이 neptune일 때 추출된 연관 규칙의 예
Table 2. Examples of extracted association rules analysis when the attack label is neptune

No.	LHS	RHS	Support	Confidence	Lift
7	{protocol_type_tcp}	{neptune}	1	1	1
228	{num_failed_logins_0,num_access_files_0}	{neptune}	1	1	1
346043	{count,dst_host_count,wrong_fragment_0,root_shell_0,su_attempted_0,num_access_files_0,protocol_type_tcp}	{neptune}	1	1	1
781581	{dst_host_srv_count,wrong_fragment_0,num_failed_logins_0,root_shell_0,su_attempted_0,num_outbound_cmds_0,is_guest_login_0,protocol_type_tcp}	{neptune}	1	1	1
1485337	{dst_host_count,urgent_0,su_attempted_0,num_shells_0,num_access_files_0,land_0,num_outbound_cmds_0,is_host_login_0,protocol_type_tcp}	{neptune}	1	1	1
2433088	{count,srv_count,dst_host_count,urgent_0,num_failed_logins_0,root_shell_0,num_shells_0,num_outbound_cmds_0,is_host_login_0,is_guest_login_0}	{neptune}	1	1	1
3490670	{count,srv_count,dst_host_srv_count,wrong_fragment_0,num_failed_logins_0,root_shell_0,su_attempted_0,num_file_creations_0,land_0,num_outbound_cmds_0,protocol_type_tcp}	{neptune}	1	1	1
3491084	{srv_count,wrong_fragment_0,urgent_0,num_file_creations_0,num_shells_0,num_access_files_0,land_0,num_outbound_cmds_0,is_host_login_0,is_guest_login_0,protocol_type_tcp}	{neptune}	1	1	1
3491165	{dst_host_count,dst_host_srv_count,wrong_fragment_0,num_failed_logins_0,num_shells_0,num_access_files_0,land_0,num_outbound_cmds_0,is_host_login_0,is_guest_login_0,protocol_type_tcp}	{neptune}	1	1	1
3494100	{count,dst_host_count,dst_host_srv_count,root_shell_0,num_file_creations_0,num_shells_0,num_access_files_0,land_0,num_outbound_cmds_0,is_host_login_0,is_guest_login_0}	{neptune}	1	1	1
3498443	{srv_count,dst_host_count,dst_host_srv_count,wrong_fragment_0,urgent_0,num_failed_logins_0,root_shell_0,su_attempted_0,num_file_creations_0,num_shells_0,num_access_files_0}	{neptune}	1	1	1

NSL-KDD 데이터셋의 총 feature 수는 43개이고, 그중에 2개의 feature는 데이터셋의 실제 label과 difficult level이다. 본 연구에서는 이 둘을 제외한 나머지 41개의 feature를 3-1장에 따라 전처리하면 136개의 feature가 되고, 이후 연관 규칙 분석을 통해 각 공격별 특이 feature를 추출한다.

표 2는 연관 규칙 분석을 통해 추출된 공격 label이 neptune일 때의 규칙들 중 일부를 보여준다. 연관 규칙 분석 결과는 Left-hand side(LHS), Right-hand side(RHS), 지지도(support), 신뢰도(confidence), 향상도(lift)로 구성된다. LHS와 RHS는 각각 3-2장에서의 X와 Y에 대응한다.

표 3. 추출된 feature별 빈도수
Table 3. Frequency by each extracted feature

Features	Count	Features	Count	Features	Count	Features	Count
count	21	land_0	22	num_shells_0	19	service_imap4	1
dst_bytes	8	land_1	1	num_shells_1	1	service_private	1
dst_host_count	23	logged_in_0	7	protocol_type_icmp	2	service_telnet	3
dst_host_same_src_port_rate	1	logged_in_1	5	protocol_type_tcp	14	src_bytes	10
dst_host_same_srv_rate	4	num_access_files_0	18	protocol_type_udp	1	srv_count	21
dst_host_srv_count	23	num_access_files_1	1	rerror_rate	1	srv_rerror_rate	1
duration	2	num_failed_logins_0	20	root_shell_0	17	srv_serror_rate	1
flag_S0	1	num_failed_logins_1	1	root_shell_1	2	su_attempted_0	22
flag_SF	12	num_file_creations_0	14	same_srv_rate	12	urgent_0	21
hot	2	num_file_creations_2	1	service_ecr_i	1	wrong_fragment_0	21
is_guest_login_0	19	num_outbound_cmds_0	23	service_ftp_data	1		

표 4. 공격별 추출된 feature 수
Table 4. Number of extract features for each attack

Attack label	Count	Attack label	Count	Attack label	Count	Attack label	Count
normal	20	smurf	22	loadmodule	15	phf	24
back	21	nmap	17	rootkit	14	warezmaster	24
land	20	satan	13	buffer	18	warezclient	18
pod	20	portsweep	14	multihop	14	ftp	15
teardrop	20	ipsweep	13	imap	19	spy	19
neptune	17	perl	25	guess_passwd	26		

먼저 추출된 규칙들 중 지지도와 신뢰도 및 향상도가 일정 수준 이상인 규칙들을 필터링하여 의미 있는 규칙들을 선별한다. 표 2에 따르면, 필터링 된 규칙은 7번 규칙과 같이 LHS의 feature가 1개부터 3498443번 규칙처럼 11개의 feature가 있는 경우가 다양하게 존재한다.

각 공격별 연관 규칙 결과에서 LHS의 feature 개수가 최대인 경우의 규칙들을 모두 통합하면, 최대한 많은 경우의 특이 feature를 포함하므로 공격에 대한 다양한 패턴을 고려할 수 있다. 공격 label이 neptune인 경우에 분석된 규칙의 최대 feature 수는 11개이지만, 모든 규칙을 통합하고 난 특이 feature의 수는 17개가 된다.

표 3은 모든 공격에 대한 통합된 특이 feature들을 종합한 빈도수를 보여준다. 총 136 feature 중에서 46 feature들이 뽑혔으며, 각 feature들은 공격에 따라 최소 1회에서 최대 23회 등장하였다. 위에서 논의한 것처럼, train 데이터셋에서 등장하는 공격 label은 23개이므로 23회 등장한 feature는 모든 공격에서 등장한 것이다.

그러한 feature들은 공격을 구분하는 데 있어 큰 영향을 미치진 않지만 1회만 등장한 특이 feature의 경우는 하나의 공격에서만 등장한 feature이므로 공격을 특정화하는 데 큰 역할을 할 수 있다.

표 5. Attack label이 neptune일 때 분석 결과 예
Table 5. Example of analysis result when the attack label is neptune

Attack label	$f(A \cap B)$	A_{ratio}	B_{ratio}	LS	Rank
neptune	17	0.708	1.000	0.854	1
nmap	17	0.708	1.000	0.854	2
imap	17	0.708	0.895	0.802	3
portsweep	14	0.583	1.000	0.792	4
land	17	0.708	0.850	0.779	5
teardrop	17	0.708	0.850	0.779	6
warezclient	16	0.667	0.889	0.778	7
satan	13	0.542	1.000	0.771	8
ipsweep	13	0.542	1.000	0.771	9
guess_passwd	19	0.792	0.731	0.761	10
back	17	0.708	0.810	0.759	11
warezmaster	18	0.750	0.750	0.750	12
smurf	17	0.708	0.773	0.741	13
normal	16	0.667	0.800	0.733	14
pod	16	0.667	0.800	0.733	15
loadmodule	13	0.542	0.867	0.704	16
multihop	12	0.500	0.857	0.679	17
rootkit	12	0.500	0.857	0.679	18
buffer_overflow	21	0.875	0.457	0.666	19
ftp_write	21	0.875	0.457	0.666	20
phf	15	0.625	0.625	0.625	21
spy	13	0.542	0.684	0.613	22
perl	14	0.583	0.560	0.572	23

표 6. 공격 label별 분석 결과
Table 6. Analysis result by attack label

Attack label	total count	top5 count	top10 count	top5 ratio	top10 ratio
neptune	4657	4657	4657	1.000	1.000
normal	9711	9635	9699	0.992	0.999
guess_passwd	1231	468	473	0.380	0.384
smurf	665	665	665	1.000	1.000
satan	735	0	453	0.000	0.616
buffer_overflow	20	0	0	0.000	0.000
back	359	359	359	1.000	1.000
warezmaster	944	452	926	0.479	0.981
pod	41	41	41	1.000	1.000
nmap	73	73	73	1.000	1.000
ipsweep	141	0	0	0.000	0.000
portsweep	157	150	157	0.955	1.000
multihop	18	1	5	0.056	0.278
loadmodule	2	2	2	1.000	1.000
teardrop	12	12	12	1.000	1.000
rootkit	13	1	2	0.077	0.154
perl	2	2	2	1.000	1.000
land	7	7	7	1.000	1.000
ftp_write	3	0	0	0.000	0.000
imap	1	1	1	1.000	1.000
phf	2	1	1	0.500	0.500

이 외에도 feature가 2회 등장한 경우는 2종류의 공격을 분류할 수 있으며, 22회 등장한 경우에는 그 feature가 없는 공격을 특정할 수 있는 등 공격을 클러스터링하는 데 도움을 줄 수 있다.

표 4에서는 각 공격별로 추출된 특이 feature의 수를 확인할 수 있으며, 공격별로 최소 13개의 feature부터 최대 26개의 특이 feature를 가지고 있다. 이 feature들은 연관 분석으로 뽑힌 규칙들을 합쳐 놓은 것이기 때문에 일부 feature만을 포함하더라도 공격을 특정할 수 있다. 하지만 특이 feature의 수가 적은 공격의 경우에는 그 특이 feature를 포함하고 더 많은 특이 feature를 가지는 공격이 존재하면 공격을 특정할 때 어려움이 있다.

2) 실제 네트워크 패킷 분석

표 5는 들어온 네트워크 패킷의 label이 neptune일 때, 제안한 기법의 분석 결과의 한 예를 보여준다. 표 4의 결과를 통해 확인할 수 있듯이, 2-3장에서 논의한 것처럼 A_{ratio}만을 고려하면 특이 feature 수가 많은 공격만 상위권에 추출되고, B_{ratio}를 기준으로 삼으면 특이 feature 수가 적은 공격이 주로 상위권에 등장하게 된다. 우리가 제시한 평가기법인 LS 결과에 따르면 실제 label인 neptune이 top 5와 top 10 모두에 포함된 것을 확인할 수 있다.

표 6은 test 데이터셋에서 등장한 각 공격별 빈도수와 실제 공격 label이 top 5와 top 10에 속한 빈도수를 비교한 전체 공격 label별 분석 결과를 나타낸다.

우리의 결과는 neptune, smurf, back 등 빈도수가 높은 공격 및 loadmodule, perl, land, imap 등 빈도수가 낮은 공격도 잘 특정화한 것을 확인할 수 있다. 하지만, 위 결과 중에 satan, guess_passwd, multihop, rootkit 등의 공격 특정화 비율은 40% 미만으로 나왔으며, ipsweep, buffer_overflow, ftp_write 공격은 모두 특정화하지 못했다. 위 공격들은 특이 feature 수가 평균보다 적거나 많은 경우이다.

다시 말하자면, 적은 특이 feature 수를 가지는 공격들은 다른 공격들에 비해 A_{ratio}값이 낮거나 특이 feature 수가 너무 많은 공격들은 B_{ratio}의 값이 낮아진다. 이로인해 LS값이 낮아지 게되고 최종적으로 top 5와 top 10에서 누락되는 결과를 초래한다. 이러한 문제를 해결하기 위해서는 또 다른 평가기법이나 deep learning 또는 reinforcement learning 기법을 추후 연구로 고려해 볼 수 있다.

3) 기존 연구와의 비교

본 연구는 4-1장에서 논의한 대로 트레이닝셋과 테스트셋에서 동시에 등장한 공격들에 대해서만 분석하였으므로, 기존

연구들과의 비교를 위해 전체 테스트셋에서 공격 label일 때, normal로 판단되면 오탐으로 간주하는 실험을 하였다. 테스트셋의 22,544개의 데이터 중 공격 데이터는 12,833개이며 실험 결과 1,666개의 오탐이 발생하였다. 그러므로 본 연구에서 제안한 기법의 accuracy는 87.01%의 정확도를 보인다고 할 수 있다. 표 7은 기존 연구들과의 비교 결과이다. 제안한 기법은 기존 연구들 대비 좋은 성능을 보여주었다.

표 7. 다른 모델과의 비교
Table 7. Comparison of accuracy with different models

Model	Accuracy
Proposed Method	87.01%
BAT[5]	82.56%
BAT-MC[5]	84.25%
BLSTM(Bidirectional LSTM)[5]	79.40%
J48[21]	81.05%
Naive Bayes[21]	74.40%
Naive Bayes Tree[21]	75.40%
Random forest[21]	74.00%
MLP(multi-layer perceptron)[21]	78.10%
CNN(Convolutional Neural Networks)[18]	80.00%
DNN(Deep Neural Network)-4[18]	82.74%
SVM(Support Vector Machine)[15]	71.31%
DBN(Deep Belief Network)[15]	71.91%
LSTM(Long Short-term memory)[15]	73.18%

V. 결 론

본 연구에서는 연관 관계 분석을 통해 공격 label을 식별하는 방법에 대해 제안하였다. 기존의 공격 탐지 기법들은 공격과 normal label을 구분하는 것에 집중하였기 때문에 실질적으로 공격을 탐지하는 것에 그쳐, 어떠한 공격인지까지는 알 수 없다. multi-class classification 기법들은 multi-class들 중 가장 근접한 하나의 class만 제시하기 때문에 정확도가 낮으면 탐지율 자체가 낮아지게 된다. 반면에 본 연구에서는 normal을 포함한 top 5, top 10의 공격 label을 추출하여 가장 연관성이 높아 보이는 공격들을 제시한다. 실험 결과에 따르면 대부분의 공격 label은 90% 이상의 탐지율을 보였으나, 몇몇 공격 label은 다소 아쉬운 결과를 확인할 수 있었다. 본 연구의 다음 확장 방안으로 공격별 특이 feature가 아닌 다른 feature가 포함되어 있을 때 페널티를 주는 평가기법 및 deep learning 또는 reinforcement learning 기법과 연동을 통해 본 연구 결과를 또 다른 입력으로 사용하여 더 높은 정확도를 추구하는 방법들을 추후에 연구할 수 있다.

감사의 글

본 연구는 방위사업청과 국방과학연구소가 지원하는 미래전투체계 네트워크기술 특화연구센터 사업의 일환으로 수행되었습니다.(UD190033ED)

참고문헌

[1] Dhanabal, L., and S. P. Shantharajah. "A study on NSL-KDD dataset for intrusion detection system based on classification algorithms." *International journal of advanced research in computer and communication engineering,* Vol. 4, issue 6, pp 446-452, June 2015. https://doi.org/10.17148/IJARCCE.2015.4696

[2] Ever, Yoney Kirsal, Boran Sekeroglu, and Kamil Dimililer. "Classification analysis of intrusion detection on NSL-KDD using machine learning algorithms." *Mobile Web and Intelligent Information Systems. MobiWIS 2019. Lecture Notes in Computer Science*, vol 11673, pp. 111–122, July 2019. https://doi.org/10.1007/978-3-030-27192-3_9

[3] Thomas, Rajesh, and Deepa Pavithran. "A survey of intrusion detection models based on NSL-KDD data set." *2018 Fifth HCT Information Technology Trends (ITT)* , Dubai, United Arab Emirates, pp. 286-291, Nov 2018 https://doi.org/10.1109/CTIT.2018.8649498

[4] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," *2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 92-96, Jan 2015. https://doi.org/10.1109/SPACES.2015.7058223

[5] Su, Tongtong, et al. "BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset.", *IEEE Access*, vol 8 pp. 29575-29585, Feb 2020. https://doi.org/10.1109/ACCESS.2020.2972627

[6] Gurung, Sandeep, Mirnal Kanti Ghose, and Aroj Subedi. "Deep learning approach on network intrusion detection system using NSL-KDD dataset." *International Journal of Computer Network and Information Security*, Vol. 11, Iss. 3,pp. 8-14, Mar 2019. https://doi.org/10.5815/ijcnis.2019.03.02

[7] F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. Mohamed Chaabani and A. Taleb-Ahmed, "Network Intrusion Detection System Using Neural Network and Condensed Nearest Neighbors with Selection of NSL-KDD Influencing Features,", *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS)*. BALI,

Indonesia, pp. 23-29, Jan 2021.
https://doi.org/10.1109/IoTaIS50849.2021.9359689

[8] Masoodi, Faheem. "Machine Learning for Classification analysis of Intrusion Detection on NSL-KDD Dataset." *Turkish Journal of Computer and Mathematics Education (TURCOMAT),* Vol. 12, Iss. 10, pp. 2286-2293, Apr 2021. https://doi.org/10.17762/turcomat.v12i10.4768

[9] Cholakoska, Ana, et al. "Analysis of Machine Learning Classification Techniques for Anomaly Detection with NSL-KDD Data Set." *Proceedings of the Computational Methods in Systems and Software.* online, Czech Republic, pp.258–267, Nov 2021.
https://doi.org/10.1007/978-3-030-90321-3_21

[10] Devan, Preethi, and Neelu Khare. "An efficient XGBoost–DNN-based classification model for network intrusion detection system." *Neural Computing and Applications*, vol 32, pp. 12499-12514, Jan 2020.
https://doi.org/10.1007/s00521-020-04708-x

[11] Chkirbene, Zina, et al. "TIDCS: A dynamic intrusion detection and classification system based feature selection." *IEEE Access*, vol 8, pp. 95864-95877, May 2020. https://doi.org/10.1109/ACCESS.2020.2994931

[12] Revathi, S., and A. Malathi. "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection.", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2 Issue 12, 1848-1853, Dec 2013.
https://doi.org/10.17577/IJERTV2IS120804

[13] Thakkar, A., Lohiya, R. "Attack classification using feature selection techniques: a comparative study." *Journal of Ambient Intelligence and Humanized Computing*, vol 12, pp.1249–1266, June 2020.
https://doi.org/10.1007/s12652-020-02167-9

[14] W. Lee, B. Noh, and K. Jeong, "Performance Evaluation of a Machine Learning Model Based on Data Feature Using Network Data Normalization Technique," *Journal of the Korea Institute of Information Security & Cryptology*, vol. 29, no. 4, pp. 785–794, Aug 2019.
https://doi.org/10.13089/JKIISC.2019.29.4.785

[15] Ding, Yalei, and Yuqing Zhai. "Intrusion detection system for NSL-KDD dataset using convolutional neural networks." *Proceedings of the 2018 2nd International Conference on Computer Science and Artificial Intelligence*, Shenzhen, China, pp. 81–85, Dec 2018. https://doi.org/10.1145/3297156.3297230

[16] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," *2015 International Conference on Signal Processing and Communication Engineering Systems*, Guntur, India, pp. 92-96, Jan 2015, https://doi.org/10.1109/SPACES.2015.7058223

[17] L. Li, Y. Yu, S. Bai, Y. Hou and X. Chen, "An Effective Two-Step Intrusion Detection Approach Based on Binary Classification and k -NN," in *IEEE Access,* vol. 6, pp. 12060-12073, Dec 2017,
https://doi.org/10.1109/ACCESS.2017.2787719

[18] Gao, M.; Ma, L.; Liu, H.; Zhang, Z.; Ning, Z.; Xu, J. "Malicious Network Traffic Detection Based on Deep Neural Networks and Association Analysis." *Sensors*, vol 20(5), pp. 1-14, Mar 2020.
https://doi.org/10.3390/s20051452

[19] Safara, F., Souri, A., & Serrizadeh, M. . "Improved intrusion detection method for communication networks using association rule mining and artificial neural networks." *IET Communications,* Vol 14(7), pp. 1192-1197, Apr 2020.
https://doi.org/10.1049/iet-com.2019.0502

[20] M. Tavallaee, E. Bagheri, W. Lu, and A. Ghorbani, "A Detailed Analysis of the KDD CUP 99 Data Set," *Submitted to Second IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA),* Ottawa, ON, Canada, pp. 1-6, Jul 2009. https://doi.org/10.1109/CISDA.2009.5356528

[21] C. Yin, Y. Zhu, J. Fei and X. He, "A Deep Learning Approach for Intrusion Detection Using Recurrent Neural Networks," in *IEEE Access*, vol. 5, pp. 21954-21961, Oct 2017. https://doi.org/10.1109/ACCESS.2017.2762418.

장영인(Young-In Jang)

2013년 : 아주대학교 정보컴퓨터공학 (공학사)
2015년 : 아주대학교 소프트웨어 특성 화학 (공학석사)

2015년～현 재: 아주대학교 컴퓨터공학과 박사과정
※관심분야 : Machine Learning, Reinforcement Learning, Anomaly Detection, Network Security

최영준(Young-June Choi)

2000년 : 서울대학교 전기공학 (공학사)
2002년 : 서울대학교 전기컴퓨터공학 (공학석사)
2006년 : 서울대학교 전기컴퓨터공학 (공학박사)

2006년～2006년: 서울대학교공학연구소 박사 후 연구원
2006년～2007년: University of Michigan Research Fellow
2007년～2009년: NEC Laboratories America Research Staff Member
2013년～2015년: 중국 Xiangtan 대학교 명예교수
2015년～2018년: 서울대학교 융합기술대학원 겸임교수
2009년～현 재: 아주대학교 컴퓨터공학과 교수
※관심분야 : Machine Learning, Reinforcement Learning, Wireless Network, Artificial Intelligence