침입탐지율 향상을 위한 네트웍 서비스별 클러스터링(clustering)

류희재(tin@ajou.ac.kr), 예홍진

아주대학교 정보통신전문대학원

To improve intrusion detection using clustering in a network service

Hee Jae Ryu, Hong Jin Yeh

Graduate School of Information and Communication, Ajou University

요약

네트웍 환경에서의 침입이 중요한 보안상의 문제점이 된 이래로, 네트웍 기반의 침입탐지시스템중에서 비정상 침입탐지(anomaly detection)의 방법 중 클러스터링을 이용한 시도들이 있었는데 기존의 방법이 네트웍 정보로부터 정상적인 클러스터들과 그렇지 않은 클러스터들 두 집단으로 크게 나누어 비교하는데 제안모델에서는 이를 좀 더 세분화하여 네트웍 서비스(network service)별로 정상적인 클러스터들과 그렇지 않은 클러스터들을 가지게 되는 방법으로 침입탐지율을 향상시켜 보고자 한다.

I. 서론

침입탐지 시스템의 목적은 네트웍의 상황을 감시하며 자동적으로 침입을 탐지하는데 있다. 공격이 탐지되면, 시스템 관리자에게 알려져야 하며 이에 따른 대응 행동이 필요하게 된다. 전통적으로 오용탐지 기반의 탐지시스템이 이와 같은 작업을 따랐으며, 이런 방법은 전문가에 의해 네트웍 데이터로부터 미리 선정된 몇 가지 특성들을 추출해 내어 내장된(hard-coded) 알고리즘을 통해 빠르게 침입을 탐지하게 된다. 하지만, 이런 방법은 새로운 공격 유형에 대해 적용할 수 없다는 단점을 가진다. 이런 문제점을 해결하기 위해 이상탐지-시스템을 시도했는데, 비정상 침입탐지 시스템이란 어떤 데이터가 정상적인 데이터로부터의 벗어난 정도를 미리 지정한 임계치와 비교하는 방식이다.

다양한 비정상 침입탐지시스템의 시도들이 있었으며, 정상이라고 알려진 데이터를 기반으로 하여 비정상적인 데이터를 침입탐지에 이용한 비정상 침입탐지시스템의 전형적인 방법이 잘 나타나 있다.[2]

클러스터링은 통계학[3], 기계학습(machine learning)[4], 데이터베이스(databases)[5] 등 다양한 분야에서 연구되어진 잘 알려진 분야이다. 클러스터링(clustering)의 기본 방법으로는 Linkage based[6], K-means[7] 가 있으며, 일반적으로 K-means 알고리즘이 좀 더 정확한 클러스터링을 만들어 낸다고 알려져 있으나 높은 시간복잡도(time complexity)를 가지게 되어 네트웍 데이터(network data) 같은 큰 데이터를 처리하기에는 부적합하다고 여겨진다. 이 밖에 Clarans[8], Birch[9], Dbscan[10]방법과 AI에서의 Self-Organizing Maps[11]과 Growing Networks[12]등을 이용한 클러스터링 방법들이 있다.

비정상 침입탐지 시스템은 컴퓨터 보안 분야에서 광범위하게 다루어지는 분야이며 침입을 탐지하는데 사용된 접근법들이 정리되어 있다.[14].

다음 장에서는 비정상 침입탐지 시스템에 쓰인 클러스터링 알고리즘, 클러스터링 방법 및 결론에 대해서 언급하도록 하겠다.

II. 본론

1. 비정상 침입탐지를 위한 클러스터링 방법

1) 개요

2001년 Columbia 대학에서는 정상과 공격에 대한 구분이 필요치 않은 네트웍 정보를 이용하여 구성된 클러스터들로 비정상 침입탐지 시스템을 제안했다.[1]

이 클러스터링을 이용한 침입탐지 방법은 유사한 유형을 가지는 데이터들은 일정 거리 안에서 가까운 클러스터에 모이게 되며, 다른 유형 혹은 특성을 가지는 데이터들은 반대의 양상을 띠게 됨을 가정하여 적용했다. 이 클러스터링 방법은 네트웍 정보를 가지고 클러스터를 만들며 단지 임의로 선정한 네트웍 정보의 특성(feature vector)들을 가지고 정상인지 비정상인지를 가려낸다.

이러한 방법은 기존의 오용탐지 시스템들이 공격 데이터를 가지고 규칙(Rule)을 생성해 내는 전처리 단계 및 알려진 공격에 대해서만 탐지를 하게 되는 단점을 피할 수 있게 한다.

2) 클러스터링

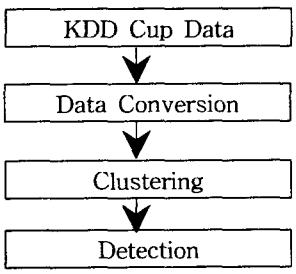
이 모델에서는 크게 두 가지 유형의 클러스터를 만들게 된다. 간단하게 말하자면 트레이닝 집합의 데이터를 통해서 미리 정해놓은 클러스터의 폭(CW: Cluster Width)과 전체 클러스터들 중 정상적인 클러스터들의 비율(PLC: Percentage of Largest Clusters)을 통해서 정상적인 데이터 유형의 클러스터와 비정상적인 데이터 유형의 클러스터를 나누어 놓으며, 이렇게 만들어진 클러스터를 통해서 들어오게 되는 데이터들을 비교하여 탐지하게 되는 방식이다.

클러스터링 방법간단하게 나타내자면 다음과 같다.

먼저, 선정된 네트웍 정보의 Feature들로 각 Feature들의 평균, 분산을 구하여 미리 지정한 CW를 가지는 클러스터들로 만들어지는 트레이닝 과정을 마치고 나면 지정된 PLC에 의해 정상적이라고 가정할 클러스터 집합의 비율이 정해지게 되고, 그 다음은 테스트를 위한 데이터들을 가지고 기존의 정상적이라고 가정한 클러스터들과 얼마나 거리가 떨어져 있는가를 따져보는 것이라고 할 수 있겠다.

클러스터들간의 거리를 측정하는 알고리즘은 standard Euclidean metric을 이용했으며, 이 경우 우리가 선정한 임의의 Feature 들 중 어느 하나가 다른 Feature 들 보다 큰 차이를 나타낸다면 클러스터가 위치하는 지점에 큰 영향을 미칠 수 있기 때문에 각 Feature별 평균과 분산을 구해서 Normalization을 거치도록 했다.

(보다 구체적인 클러스터링 방법은 [1]을 참고하기 바란다.)



다음은 클러스터를 만들 때 이용한 데이터와 중요하게 여겨지는 두 가지 매개변수에 대한 설명이다.

클러스터 트레이닝 및 테스트에는 KDD Cup 1999 Data[13]를 이용했으며, 약 700Mbyte에 달하는 이 파일에는 4,900,000 개의 네트웍 정보에 각각 41가지의 Feature를 산출한 다양한 유형의 정상 및 공격 데이터가 있다.

네트웍 Feature 선정에는 수치화 하기 힘든 Symbolic Feture들을 제외한 KDD Cup Data가 가지고 있는 모든 Feature들을 선택했다. 예를 들자면 duration time, source bytes, destination bytes, number of failed login' 등의 정보이다.

표 1: 클러스터링의 두가지 중요 매개변수

항목	설명
CW	standard Euclidean metric방법을 적용하여 클러스터 간의 거리를 측정하는 클러스터의 폭
PLC	정상적인 데이터라고 가정하는 클러스터 개수/전체 클러스터 개수의 비율

CW는 하나의 클러스터가 가지는 크기로 하나의 네트웍 정보가 선정한 Feature들에 의해 벡터로 만들어졌을때 어느 Cluster에 속하게 되느냐가 결정되는 중요한 역할을 한다. PLC는 트레이닝을 거친 클러스터들 사이에서 정상적인 데이터 클러스터를 몇 % 로 볼 것인가를 결정하는 변수이다. PLC를 50% 로 잡는다면 현재의 모든 클러스터중 정상적인 데이터를 표현하는 클러스터의 집합이 50%가 된다는 것을 의미한다.

2. 구성방법

1) 기존모델 실험결과

다음 표에 실험에 의한 두가지 매개변수의 적절한 값을 나타내었다.

DR(Detection Rate): 침입탐지율

FPR(False Positive Rate): 오용탐지율

표 2: PLC별 탐지율 (CW= 20)

PLC (%)	DR (%)	FPR (%)
15	35.7	1.44
7	66.2	2.7
2	88	8.14

클러스터 너비를 고정하고 PLC크기의 선정을 위해 임의적으로 크기를 조정했을때 PLC가 작아지면 DR은 높아지지만, FPR 또한 높아짐을 알 수 있다.

표 3: Cluster Width 별 탐지율 (PLC = 15%)

CW	DR	FPR
30	28.1%	1.07%
40	30.77%	0.84%
60	31.9%	0.7%
80	22.84	0.6%

이 실험이외에도 DR, FPR을 좋게 하기 위한 몇가지 테스트가 더 있었으며, DR, FPR 모두 실제 상용제품에서 쓰일만큼의 만족할만한 탐지율을 얻지는 못했으나, FPR에서 만큼은 비교적 양호한 편이었다.

2) 제안모델 실험결과

다음은 제안모델에서 클러스터를 만들 때 사용한 데이터 집합과 두 가지 중요 매개변수 설정에 관한 설명이다.

제안모델에서는 KDD Cup 1999 Data의 처리의 수행시간이 걸리는 문제로 원 Data 의 10% Data를 기반으로 TCP 서비스(ftp, telnet, http 등)만으로 제한하였으며 약 580000개의 네트워 정보를 가지고 테스트를 진행했음을 밝힌다.

CW와 PLC선정에 관한 결과는 다음 도표와 같다.

표 4: PLC별 탐지율 (CW = 40)

PLC (%)	DR (%)	FPR (%)
50	88.82	0.26
40	89.98	0.29
20	98.85	0.43
10	99.45	0.87

PLC가 작아질수록 매우 좋은 DR을 나타냄을 보이지만, FPR이 안좋아지는 상관관계를 가지게 된다.

표 5: CW별 탐지율 (PLC = 40%)

CW	DR (%)	FPR (%)
80	89.01	0.25
50	90.02	0.27
40	89.98	0.29
30	94.02	0.32
10	99.54	0.88

PLC를 40%로 정하고 CW의 크기를 조정해 보지만 특별히 나쁜 결과를 제시해 주는 경우가 없었으며 CW가 30이하로 떨어지는 경우 FPR이 약간 악화되는 경향을 보이고 있다.

3) 실험결과 비교

기존의 클러스터 모델은 네트웍으로 들어오는 모든 데이터에 대한 정상적인 클러스터와 비정상적인 클러스터를 가지고 이를 비교하여 침입탐지에 이용했으나 결과에서 DR, FPR 둘 다를 만족할만한 경우를 찾기가 쉽지 않았으므로 제안모델에서는 네트웍으로 들어오는 데이터를 각 서비스별로 클러스터를 나누어 만들어 해당하는 서비스별 정상적인 클러스터와 비정상적인 클러스터로 나누어 침입탐지 모델을 만들어 보았다.

먼저, 기존모델에서의 CW가 제안모델에서의 그것보다 작은 이유는 트레이닝 데이터 집합의 차이에서 비롯되었다고 말 할 수 있겠다. 기존모델이 훨씬 넓은 클러스터 분포를 가지기 때문에 제안모델보다 작은 PLC를 가지는 것이 적절했을 것이다.

기존모델에서는 그런데로 만족할만한 FPR을 얻을 수 는 있었지만, 실제 상용제품에 적용할수 있을만한 수준의 DR을 얻기는 어려워 보였다.

그러나, 제안모델에서 위의 실험결과는 실제 상용제품에 쓰일 수 있을만큼의 DR, FPR의 현저한 탐지율 향상이 있었다.

III. 결론

네트웍 데이터 전체를 클러스터로 잡은 기존의 연구와 비교해 볼 때 서비스별 클러스터링은 DR, FPR 모두에서 우수한 결과를 나타내었다.

트레이닝 데이터 집합을 통한 클러스터링은 만들어지는데 시간이 더 걸리긴 하지만, 이것이 테스팅 데이터를 탐지하는데 걸리는 시간은 거의 차이가 없다고 할 수 있다. 현재 제안모델에서는 TCP 프로토콜 서비스만 테스트를 해보았으나 다른 프로토콜의 서비스에 적용을 하는 경우에도

좋은 결과가 있을 것으로 생각된다.

현재 네트웍 데이터에서의 Feature 추출 과정에서의 선정이 전문가들에 의한 임의적인 선택에 의존하고 있는데 네트웍 Connection 데이터에서의 Feature 선정문제와 클러스터링을 만드는 과정에서의 두 가지 중요 매개변수(CW와 PLC)의 임의적인 선정이 아닌 자동화 등도 향후 다루어 져야 할 문제가 되어야 할 것으로 생각된다.

시간 복잡도가 고려되어야 하겠지만 클러스터링을 이용한 네트웍 기반의 비정상 침입탐지시스템에서의 서비스별 클러스터링 혹은 각 서비스의 특정한 Feature 별 클러스터링등도 향후 생각해 볼 수 있는 접근중 한가지가 될 수 있을것이다.

참고문헌

[1] Leonid Portnoy. Intrusion detection with unlabeled data using clustering. Data Mining Lab, Department of Computer Science, Columbia University, 2001

[2] H.S. Javitz and A. Valdes. The nides statistical component: description and justification. In Technical Report, Computer Science Laboratory, SRI International, 1993.

[3] P. Schnell. A method for discovering data-groups, 1964.

[4] R. Rojas. Neural Networks A systematic introduction. Springer, Berlin, 1996

[5] Alexander Hinnebrug and Daniel A. Keim. Clustering methods for large databases: From the past to the future. In Alex Delis, Christos Faloutsos, and Shahram Ghandeharizadeh, editors, SIGMOD 1999, Proceedings ACM SIGMOD International Conference on Management of Data, June 1-2, 1999, Philadelphia, Pennsylvania, USA. ACM Press, 1999.

[6] H.H. Bock. Automatic Classification. Vandenhoeck and Ruprecht, 1974.

[7] K. Fukunaga. Introduction to Statistical Pattern Recognition, Second Edition. Academic Press, Boston, MA, 1990

[8] R. Ng and J. Han. Efficient and effective methods for spatial data mining, 1994.

[9] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases, 1996.

[10] M. Ester, H. Kriegel, J. Sander, and X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, 1996.

[11] R. Rojas. Neural Networks A systematic introduction. Springer, Berlin, 1996.

[12] D. Touretzky B. Fritzke and T. Leen. Advances in neural information processing systems, 1995.

[13] KDD99. Kdd99 cup dataset, 1999 (http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html)

[14] D.E Denning. An Intrusion detection model. IEEE Transactions on Software Engineering, SE-13:222-232, 1987