情 報 保 護 學 會 誌
第 13 券 第 1 號, 2003. 2

침입탐지율 향상을 위한 네트웍 서비스별 클러스터링(clustering)

류 희 재*, 예 홍 진*

요 약

네트웍 기반의 컴퓨터 보안이 컴퓨터 보안분야의 중요한 문제점으로 인식이 된 이래 네트웍 기반의 침입탐지 방법 중 클러스터링(Clustering)을 이용한 비정상 탐지기법(Anomaly detection)을 사용하는 시도들이 있었다. 네트웍 데이터 같은 대량의 데이터의 처리에 클러스터링을 통한 방법이 효과적인 결과를 나타내었음이 다수의 논문에서 제기되어왔으나 이 모델에서의 클러스터링 방법은 네트웍 정보로부터 추출한 정보들을 정상적인 클러스터들과 그렇지 않은 클러스터들 크게 두 집단으로 나누는 방법을 택했었는데 침입탐지율에서 만족할만한 결과를 얻지 못했었다. 본 논문에서 제안하고자 하는 모델에서는 이를 좀 더 세분화하여 네트웍 서비스(Network service)별로 정상적인 클러스털들과 그렇지 않은 클러스터들을 가지게되는 방법을 적용하여 기존 모델에서의 침입탐지율 결과의 개선을 도모해 보고자 한다.

Ⅰ. 서 론

침입탐지 시스템(Instrusion detection system-IDS)의 목적은 호스트 또는 네트웍을 감시하며 자동적으로 침입을 탐지하는데 있다. 공격이 탐지되면, 시스템 관리자에게 알려져야 하며 이에 따른 대응 행동이 필요하게 된다. 전통적으로 네트웍 기반의 오용탐지 탐지시스템(Misuse detection)이 이와 같은 작업을 따랐으며, 이런 방법은 전문가에 의해 네트웍 데이터로부터 미리 선정된 몇 가지 특성들을 추출해 내어 내장된 알고리즘을 통해 빠르게 침입을 탐지하게 된다. 하지만, 이런 방법은 새로운 공격 유형에 대해 적용할 수 없다는 단점을 가진다. 이런 문제점을 해결하기 위해 비정상 탐지 시스템(Anomaly detection)을 시도했는데, 비정상 침입탐지 시스템이란 어떤 데이터가 정상적인 데이터로부터의 벗어난 정도를 미리 지정한 임계치와 비교하는 방식이다.

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

클러스터링(Clustering methodology) 은 통계학, 기계학습 (Machine learning), 데이터베이스 등 다양한 분야에서 연구되어진 잘 알려진 분야이다.

클러스터링의 기본 방법으로는 연결기반(Linkage based), K-means 방법이 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, Birch, Dbscan 방법과 인공지능(Artificical intelligence) 에서의 자기조직화 신경망(Self-organizing maps)과 구성적 학습(Growing networks) 등을 이용한 클러스터링 방법들이 있다.

본 논문에서는 기존의 클러스터링을 이용한 비정상 침입탐지 시스템의 침입탐지율 향상을 위한 네트웍 서비스별 클러스터링을 이용한 제안모델과 이의

* 아주대학교 정보통신전문대학원 ({tin, hjyea}@madang.ajou.ac.kr)

결과를 보일 것이다.

본론의 첫 장에서는 침입탐지 시스템에 대해서 현재의 침입탐지 시스템의 종류와 기법등을 설명하고,[5] 비정상 침입탐지 시스템에 적용되는 클러스터링 알고리즘과 이에 따른 특징 등을 두 번째 장에 설명하도록 할 것이다.[4] 마지막 세 번째 장에서는 클러스터링을 이용한 비정상 침입탐지시스템에 대한 자세한 내용을 서술하기로 한다.[1]

II. 본론

1. 침입탐지 시스템

1.1 침입탐지

침입은 허가되지 않은 접근임은 물론 호스트(Host)의 보안 요소를 침해하는 모든 행위를 침입으로 간주할 수 있다. 이는 비밀번호 해킹을 통한 접근을 포함하여 실제적인 침입을 위한 포트 스캐닝(Port scanning) 등 그 종류는 무한하다. 이러한 침입과 칩입을 위한 시도 등에 대해 보호하고자 하는 호스트나 네트워크에 대해 감시하고 실제 발견시 경고 및 대응하는 행위를 침입 탐지라 한다. 이러한 개념은 1980년 J.P Anderson에 의해 소개되어졌고 최근 몇 년 사이에 실제 적용할 수 있는 제품들이 출시되고 있다.

침입 탐지 시스템의 일반적인 동작원리는 다음과 같다.

데이터 수집 -> 데이터 축약 (Reduction) -> 탐지 -> 응답(Response)

데이터를 수집하고 중복된 데이터를 필터링하고 다양한 탐지 기법을 사용해 침입을 탐지하고 탐지된 결과에 따라 적절한 대응조치를 결정하여 응답을 하는 시스템이다.

1.2 침입탐지 시스템의 분류

침입탐지시스템의 분류는 보호하고자 하는 대상 시스템 즉, 침입을 판단하기 위한 데이터를 제공하는 소스(Source)에 따라 분류가 이루어지는데 크게 네트워크 기반(Network-based)과 호스트 기반 (Host-based)으로 분류된다. 네트워크 기반이 좀 더 일반적이며 네트워크를 통해 전송되는 트래픽을 검사하여 침입을 탐지하게 된다. 반면 호스트 기반은 로컬 호스트에서 사용자의 행위나 프로세스(Process)들을 검사하여 침입을 탐지하게 된다.

1.2.1 호스트 기반의 침입탐지 시스템

호스트 기반 IDS는 단일 호스트에서 침입을 탐지 하는 것으로 그 호스트의 감사 (Audit) 기록이나 들어오는 패킷 등을 검사하여 침입을 탐지하게 된다. 예를 들어 호스트에 login 프로세스를 감시하고 root 사용자의 행동을 감시하며, 파일 시스템 감시 등을 통해 침입을 발견하는 것이다.

호스트 기반 IDS는 가능한 공격에 대해 꽤 강력한 도구로 사용될 수 있다. 예로 시스템 로깅(System logging) 을 통해 공격자가 어떤 행위를 했는지 어떤 파일을 열었고, 어떤 시스템 콜(System call)이 실행되었는지 등을 알 수 있다. 또 네트워크 기반 IDS 보다 잘못된 탐지, 즉 침입이 아님에도 불구하고 침입이라 판단하는 경우(False Alarm)가 좀 더 적다.

호스트 기반 IDS의 단점으로는 우선 IDS를 대상 호스트에 설치해야 하므로 대상 호스트의 성능이 저하되고 데이터를 얻기 위한 시스템 설정이 번거로우며 대상 호스트가 있는 네트워크 내의 다른 호스트들의 상태에 대해서는 알 수가 없는 점이라고 말할 수 있겠다.

1.2.2 네트워크 기반 침입탐지 시스템

네트워크 기반 IDS는 패킷 스니퍼(Packet sniffer)와 패킷 모니터(Packet monitor) 도구의 발전으로 볼 수 있다. 네트워크의 의 모든 트래픽에 대해 패킷을 수신하고 분석하여 침입을 발견하는 일이 엄청나게 복잡함은 당연하다. 이를 자동으로 처리하는 것이 바로 네트워크 기반 IDS라고 볼 수 있다.

네트워크 기반 IDS는 특히 권한 없이 접근한다거나 권한을 초과하는 접근에 대한 탐지에 뛰어난 편이다. 또한 네트워크내의 호스트나 서버의 별도의 설정 없이 사용이 가능하며, 방화벽처럼 라우팅같은 심각한 역할을 담당하지 않기 때문에 오류 발생시에 큰 피해를 주지 않게 된다.

반면 네트워크 기반 IDS는 성능에 대한 요구사항 때문에 서명 분석(Signature analysis)을 하는 경우가 많은데 이는 일반적인 알려진 공격을 탐지하는데는 뛰어나나 복잡한 정보를 가진 위협요소에 대한 공격은 탐지하기가 어렵다. 또한 분석을 위해 엄청난 양의 데이터 교환을 필요로 할 수 있다.

이를 위해 분석을 위한 데이터를 축약 과정 통해 필터링하게 된다. 물론 모든 패킷에 대한 분석이 좀 더 많은 침입을 좀 더 정확히 탐지할 수 있음은 당연하다. 네트워크 기반 IDS는 암호화 세션(Encrypted session)에 대한 침입 탐지의 결점이 있다.

1.3 침입탐지 기법

탐지 방식들을 크게 분류하면 비정상 행위 탐지와 오용 탐지 두가지로 분류가 가능하며 비정상 행위 탐지는 알려지지 않은 새로운 공격 기법도 탐지가 가능하다는 장점이 있지만 그에 앞서 정상적인 행위에 대한 프로파일을 구축해둬야 하기 때문에 많은 데이터의 분석이 필요하게 된다. 때문에 상대적으로 구현 비용이 큰 편이고 그만큼 어렵기 때문에 상용 제품에서는 오용 탐지를 주로 사용하고 비정상 행위 탐지는 보조하는 측면에서 사용되고 있다.

1.3.1 비정상 행위 탐지(Anomaly detection)

비정상 행위 탐지는 정상적인 시스템 사용에 대한 프로파일(Profile) 상태를 유지하며 이에 어긋나는 행위를 탐지하는 방식이다. 즉 시스템 가동 전에 정상적인 행동에 대한 프로파일을 작성해 두고 가동 후에 현재 행위들을 정상적인 프로파일과 비교하여 공격을 탐지하게 된다. 비교 과정에서 기존의 프로파일을 수정하거나 새로운 프로파일을 추가하기도 한다.

비정상 행위를 위한 탐지 기법은 다음과 같다.

1.3.1.1 통계적 접근(Statistical approach)

이 방식은 과거의 통계 자료를 바탕으로 현재 프로세스의 행위를 관찰하여 프로파일을 작성하고 작성된 프로파일을 통해 비정상 정도(Anomality)를 측정하여 침입을 탐지한다. 비교적 정확한 탐지가 가능하다고 알려져 있다. 클러스터링을 이용한 접근 방식도 이 범주에 포함된다.

1.3.1.2 예측 가능 패턴 생성(Predictive pattern generation)

이 방식은 해당 순간까지 발생한 이벤트들을 바탕으로 다음 이벤트를 예측하여 침입을 탐지하게 된다. 즉 룰에 따라 어떤 이벤트들이 순차적으로 발생했다고 가정하면 그 후에 발생할 수 있는 이벤트는 어떤 것들이고 그 발생 확률이 어느정도이다까지 예측이 가능하게 된다.

1.3.1.3 신경망(Neural networks)

이 방식은 현재까지의 사용자의 행동이나 명령이 주어졌을 때 사용자의 다음 행동이나 명령을 신경망이 예측하도록 훈련시킨 후 실제 사용자들의 프로파일을 작성케 하여 이를 이용하여 침입을 탐지한다.

1.3.2 오용 탐지(Misuse detection)

오용 탐지는 알려진 취약성을 통한 공격에 대한 정보를 가지고 실제적인 공격이 시도될 때 이를 탐지하는 방식이다. 비정상 행위 탐지가 침입으로 여겨지는 행위를 탐지한다면 오용 탐지는 명백한 침입을 탐지하게 된다.

오용 탐지를 위한 접근 방식은 다음과 같다.

1.3.2.1 전문가 시스템(Expert system)

이 방식은 매칭 부분과 액션 부분을 구분한 if-then 룰을 이용해 현재 행위와 일치하는 공격 패턴을 찾는 방식으로 정해진 액션을 통해 대응하게 된다. SRI에 의해 개발된 NIDES(Next Generation Intrusion Detection Expert System)가 이 방식을 사용하고 있다.

1.3.2.2 키 스트로크 모니터링(Keystroke monitoring)

이 방식은 매우 간단한 것으로 공격 패턴을 keystroke를 모니터링하여 발견하게 된다.

1.3.2.3 상태 전이 분석(State transition analysis)

시스템 내부에 침입자가 시스템 관리자 권한을 얻기까지의 과정을 단계별로 기술하여 지식 베이스로 저장한 다음, 이를 감사자료에 나타난 사용자의 사용패턴과 비교하여 침입을 찾아내는 것이다. 이 방식은 시스템의 상태에 따라 전이하면서 공격을 감지하게 된다.

1.3.2.4 패턴 매칭(Pattern matching)

이 방식은 알려진 공격 유형들을 패턴으로 가지고 있으면서 현재 행위와 일치하는 패턴을 찾아내 침입을 탐지한다.

오용 탐지는 비정상 행위 탐지와 비교하여 비교적

구현 비용은 저렴하나 탐지를 위한 데이터가 시스템의 감사 정보를 주로 이용하며 또 최신 공격 기법이 발견되면 룰을 추가해줘야 하는 번거로움이 있다.

2. 클러스터링

2.1 개 요

클러스터링이란 주어진 데이터 셋을 서로 유사성을 가지는 몇 개의 클러스터로 분할해 내가는 과정으로, 하나의 클러스터에 속하는 데이터 점들 간에는 서로 다른 클러스터 내의 점들과는 구분되는 유사성을 갖게 된다. 데이터 마이닝에서 클러스터링 방법은 기존의 통계학, 기계 학습, 패턴 인식에서 쓰이던 방법에 부가적으로 데이터베이스 지향적인 제약 사항들 (제한된 메모리 양, I/O 시간 최소화 등)을 첨가 시킨 것으로써, 최근의 멀티미디어 데이터와 같이 혼합되고 다양한 다차원 데이터를 효율적으로 분류해 나가기 위한 방안으로 연구되고 있다. 클러스터링 방법은 크게 분할적 접근 (Partitioning approach)과 계층적 접근 (Hierarchical approach)으로 나눌 수 있다.

분할적 접근 방법은 어떠한 범주 함수를 최적화 시키는 K개의 구획 (Partition) 을 결정해 나가는 방법으로, 유클리드 거리(Euclidean distance) 측정법에 기반 한다. 여기에는, 클러스터의 무게중심점을 대표 값으로 분할해 나가는 K-means 방법과, 클러스터내의 중심과 가장 가까운 개체를 대표 점을 찾아 가는 K-medoid 방법이 있으며, 분할을 위한 초기 값 선정방식이나 대표 값 선정 방식에 따라, 또는 거리 대신 밀도를 기반으로 하느냐에 따라 여러 가지로 변형될 수 있다.

계층적 접근 방법은 처음에 각각의 데이터 점을 하나의 클러스터로 설정 한 후 이들간의 거리를 기반으로 하여 분할/합병 해 나가는 상향식(Bottom up) 방식으로 모든 점들이 하나의(Large single cluster)에 속하게 될 때까지 그 연관된(History) 정보를 유지해 나가게 된다. 한 쌍(Pair) 간의 거리를 어떻게 측정하느냐에 따라 단일 연결법(Single linkage), 최장연결법 (Complete linkage), 중심연결법(Centroid linkage) 등을 이용하는 다양한 방법이 존재한다.

2.2 알고리즘

2.2.1 CLARANS

K-medoid 방법을 사용하는 대표적인 알고리즘인 PAM과 CLARA를 바탕으로 개발된 알고리즘으로 적절한 클러스터 값을 찾아가는 각 단계마다 이웃의 샘플만을 고려한다. PAM은 클러스터에서 가장 중앙에 위치한 대표 점 medoids와 다른 객체들 사이 모든 쌍들을 분석해가면서 반복적으로 최상의 medoids를 선택해 나가는 것이다. CLARA는 Sampling에 기초하여 실제 데이터의 적은 부분만 사용하여 medoids를 선정하며 큰 데이터 검색이 가능하다고 알려져 있다.

CLARANS 는 고정 Sampling이 아닌 검색의 각 단계에서 특정한 무작위성 Sample을 뽑아서 사용하고 있지만 1000여 개 이상의 데이터 집합에 대해서는 적용이 불가능한 제약점을 가지고 있다.

2.2.2 BIRCH

기존의 클러스터링 알고리즘에서 입력 데이터 N이 커지면 이들에 대한 다중 입출력 스캔으로 인해 병목현상이 발생하게 되고 비선형시간 복잡도로 인한 처리 비용이 급격히 증가된다는 제약점을 극복하기 위해 제안 되었다. 알고리즘 수행 방식은 먼저, 전체 데이터를 스캔해내는 전-클러스터링 단계를 수행한 후 유용한 메인메모리 내에 맞는 서브클러스터에 대해 검색하는데 서브클러스터에 대한 요약 정보를 갖고 있는 CF-tree를 사용함으로써 다량의 데이터 베이스에 대해 효율적인 클러스터링을 수행한다. 입출력 비용을 최소화하면서 모든 가능한 서브클러스터를 파생시키고 유용한 메모리를 최대한 사용 가능하게 하며, 다차원 데이터들이 증가되거나 동적으로 입력되는 상황에서 좋은 결과를 갖는 클러스터를 생성하는 특징을 갖는다.

2.2.3 DBSCAN

기존의 클러스터링 알고리즘은 대규모의 데이터 집합을 효율적으로 다루는 방법에 대해서만 다루었던 데 비해, DBSCAN 에서는 다차원적이고 공간적인 특성을 갖는 다양한 모양과 크기의 데이터에 대한 클러스터링 방법을 제시한다. DBSCAN에서는 클러스터의 밀도를 결정하기 위해 2개의 파라미터

즉 포인트의 이웃의 범위를 나타내는 반경 (Eps)와 최소 이웃의 수 (MinPts)를 입력받는다.

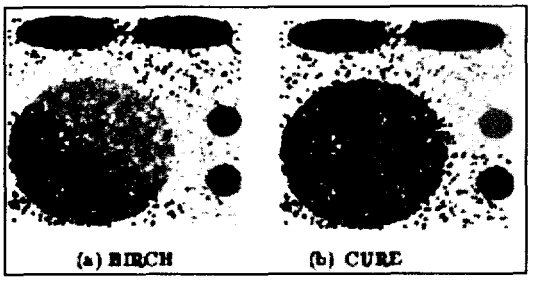
DBSCAN은 임의의 모양의 클러스터를 찾는데 있어서CLARANS 에 비해 훨씬 효율적일 뿐만 아니라 CLARANS에 비해 100배 정도의 효율성을 갖는다.

2.2.4 CURE

전통적인 계층적 클러스터링 알고리즘이 가질 수 있는 체인 효과 문제를 해결하기 위한 방안으로 제시 되었다.

CURE (Clustering Using Representatives)는 클러스터당 하나 이상의 대표 점을 가지며, 이들은 클러스터의 평균값으로 줄어드는 well-scattered point 로 지정된다. 계층적 클러스터링 방법을 적용시킬 때 합병되는 두 클러스터에 대한 대표 점은 합병된 클러스터내의 모든 점에 대해서가 아닌 두 오리지널 클러스터로부터 선택되어지며 특히, 랜덤 샘플링과 분배 (Partitioning), k-d tree와 heap data 구조를 사용 함으로써 기존에 단일 중심점 (Single centroid) 만으로는 찾아낼 수 없었던 비구형 클러스터, 특히 긴 모양의 클러스터를 발견 가능 하게 하는 특징이 있다. DBSCAN 과 비교할 때 입력파라미터에 대한 영향력이 적고, 밀도 높은 선으로 연결된 두 개의 서로 다른 클러스터를 구분해 낼 수 있으며 커다란 데이터베이스에 적용 할 때 전-클러스터링을 수행할 수 있는 장점을 갖는다.

다음의 그림 1은 CURE를 이용할 때 BIRCH에 비해 다양한 크기와 모양의 클러스터를 구분해 내는 예이다.

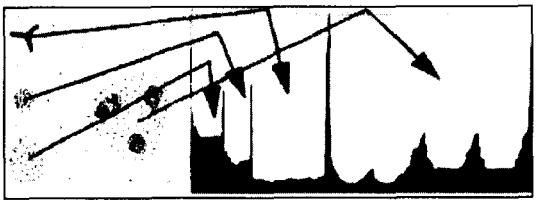


〔그림 1〕 BIRCH알고리즘과 CURE알고리즘의 비교

2.2.5 OPTICS

실제로 고차원적 데이터(High dimensional data)가 있는 실제 데이터에서는 차원에 따라 데이터의 분포가 치우치게 되기 때문에 글로벌한 입력 파라미터를 결정하기 어렵다. 이러한 경우 계층적 알고리즘을 사용하여 해결할 수 있는데 이 때는 싱글 링크 효과와 객체의 수가 수백개가 넘어가는 경우 분석이 어려워진다는 단점을 갖는다. 다른 방법으로는 서로 다른 여러 개의 파라미터를 이용하여 밀도 기반 구획(Partitioning) 알고리즘을 사용할 수 있지만 이 때 가능한 파라미터의 범위가 너무 많다는 단점을 갖는다. OPTICS 알고리즘은 실제로 클러스터를 생성하지는 않지만 밀도 기반 클러스터링 구조를 표현하는 데이터베이스의 Argumented ordering을 생성함으로써 하나의 글로벌 파라미터의 제한을 받지 않고 넓은 범위의 파라미터 셋팅과 연관한 밀도 기반 클러스터링과 동일한 정보를 나타낸다.

다음의 그림 2는 서로 다른 모양과 크기와 밀도를 갖는 클러스터들과 그들 사이의 계층적인 정보를 나타내는 OPTICS의 시각화된 결과이다.



〔그림 2〕 OPTICS visualization

2.2.6 ROCK

장바구니 DB(Market basket database)와 같이 Boolean이나 범주형 속성(Categorical attributes)을 가지는 데이터에 대한 계층적 클러스터링 알고리즘으로 제안되었다. 각각의 클러스터를 합병할 때 데이터간의 유사성 측정 기준으로 거리 대신 링크라는 새로운 개념을 도입하였다. 즉 두 개의 포인트 쌍이 유사성 측면에서 특정 임계치 이상인 경우 이웃라고 결정하고 포인트들 간의 공통 이웃의 개수를 두 점의 링크수라고 정의한다. 동일한 클러스터에 속하는 점들은 일반적으로 많은 수의 공통 이웃의 수를 갖고 동시에 많은 수의 링크를 갖는다. 그러므로 클러스터를 합병할 때 가장 많은 수의 링크를 갖는 것끼리 먼저 합병하는 것이 의미 있는 클러스터를 생성하게 된다. 클러스터의 합병을 위해 적합도 측정(Goodness measure) 이 제안되었고 랜덤 샘플링을 이용해 보다 많은 데이터에 적용이 가능하다.

클러스터링 알고리즘은 다량의 데이터 셋에 대해 효율성을 증진시키는 방법으로 여러 가지 샘플링기법 이나 경계 최적화 기법, 인덱스 기법, 집중화 기법등을 사용하고 있으며, 향후 대상 데이터 집합의 특성과 클러스터링의 목적에 따른 최상의 알고리즘 선택 기준에 대해 지속적인 연구가 필요하다. 응용 영역은 사용자 의도에 따른 원본 이미지 필터링이나 특성 인식, 티슈 세그멘테이션 등의 영상 이미지 분석 분야이다.

3. 비정상 침입탐지를 위한 클러스터링 방법

2001년 미국의 컬럼비아 (Columbia) 대학에서는 정상과 공격에 대한 구분이 필요치 않은 네트워 정보를 이용하여 구성된 클러스터들로 비정상 침입탐지 시스템을 제안했다.

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 네트워 정보를 가지고 클러스터를 만들며 단지 임의로 선정한 네트워 정보의 특성(Feature vector)들을 가지고 정상인지 비정상인지를 가려낸다[1].

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.

3.1 모델 개요

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 방법을 간단하게 나타내자면 다음과 같다.

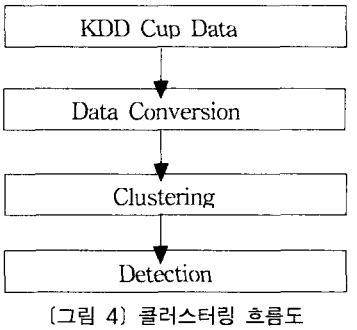
먼저, 선정된 네트워 정보의 Feature들로 각 Feature들의 평균, 분산을 구하여 미리 지정한 CW를 가지는 클러스터들로 만들어지는 트레이닝 과정을 마치고 나면 지정된 PLC 에 의해 정상적이라고 가정할 클러스터 집합의 비율이 정해지게 되고, 그 다음은 테스트를 위한 데이터들을 가지고 기존의 정상적이라고 가정한 클러스터들과 얼마나 거리가 떨어져 있는가를 따져보는 것이라고 할 수 있겠다.

```
Step 1. Initialize the set of clusters, S, to
        the empty set
Step 2. Obtain Data instance d from the
        given training set T
        if there is the cluster C in S with d
        associate d with cluster C
        else
        create cluster C
        add C to S
Step 3. Repeat Step 2 until no instances are
        left in T
```

〔그림 3〕 클러스터링 Pseudo Code

클러스터들간의 거리를 측정하는 알고리즘은 유클리드 거리 (Standard Euclidean metric)를 이용했으며, 이 경우 우리가 선정한 임의의 Feature들 중 어느 하나가 다른 Feature 들 보다 큰 차이를 나타낸다면 클러스터가 위치하는 지점에 큰 영향을 미칠 수 있기 때문에 각 Feature별 평균과 분산을 구해서 정규화를 거치도록 했다.

최종적인 침입탐지 흐름은 다음과 같은 경로를 거친다.



〔그림 4〕 클러스터링 흐름도

3.2. 정규화 (Normalization)

다음의 3가지 Feature 에 대한 거리측정 경우를 생각해보자.

예) 3가지 Feature를 가지는 두 벡터가 다음과

같은 경우
{(1, 3000, 2), (1, 4000, 3) }

두 벡터간의 거리:

$$\sqrt{(1-1)^2+(2000-3000)^2+(2-3)^2}$$

이런 경우 각 벡터의 두 번째 원소가 벡터간의 거리를 측정하는데 가장 큰 영향을 끼치게 되기 때문에 각 Feature들의 평균, 분산을 구한 값에서 각 벡터간의 거리를 측정해야 한다.

다음은 정규화를 하기 위한 각 벡터의 평균, 분산, 새로운 벡터가 위치하는 지점을 나타낸다.

- 평균

$$avgvector[j] = \frac{1}{N}\sum_{i=1}^{N} instance_i[j]$$

〔수식 1〕 벡터 평균

- 분산

$$stdvector[j] = \left(\frac{1}{N-1}\sum_{i=1}^{N}(instance_i[j] - avgv\ ector[j])^2\right)^{1/2}$$

〔수식 2〕 벡터 분산

- 새 벡터 위치

$$*winstance[j] = \frac{instance[j] - avgvector[j]}{stdvector[j]}$$

〔수식 3〕 새 벡터 위치

3.3. 매개변수

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

클러스터 트레이닝 및 테스트에는 KDD Cup 1999 Data를 이용했으며, 약 700Mbyte에 달하는 이 파일에는 4,900,000 개의 네트워 정보에 각각 41가지의 Feature를 산출한 다양한 유형의 정상 및 공격 데이터가 있다.[3]

〔표 2〕 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

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

CW는 하나의 클러스터가 가지는 크기로 하나의 네트워 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.

4. 실험설계 결과분석

본 논문에서 제안한 클러스터링 방법과 기존모델과의 차이점은 클러스터링 대상이 네트워 상의 데이터를 모두 정상 및 비정상 클러스터에 넣어 비교하는데 비해 각 네트워 서비스별 정상, 비정상 클러스터를 가지게 되는 점이다.

4.1 기존모델 실험결과

다음 표에 실험에 의한 두 가지 매개변수의 적절한 값을 나타내었다.

DR (Detection Rate): 침입탐지율
FPR (False Positive Rate): 오용탐지율

〔표 3〕 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.

이 실험이외에도 DR, FPR의 결과값의 상관관계와 적절한 값을 찾기 위한 위한 몇가지 테스트가 더 있었으나, DR, FPR 모두 실제 상용제품에서 쓰일만큼의 만족할만한 탐지율을 얻지는 못했다.

하지만, FPR의 결과만큼은 비교적 양호한 편이었다.

〔표 4〕 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%

4.2 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 Data의 10% Data를 기반으로 TCP 서비스(ftp, telnet, http 등)만으로 제한하였으며 약 76800개의 네트워 정보를 가지고 있는 Training Set과 190000여개의 Test Set 데이터를 가지고 진행했음을 밝힌다.

기존모델과 제안모델에서의 CW와 PLC선정에 관한 결과는 다음 도표와 같다.

〔표 5〕 PLC별 탐지율 (CW = 40) 기존모델

PLC (%)	DR (%)	FPR (%)
50	66.45	0.25
40	68.03	0.29
20	72.74	0.31
10	74.51	0.44

〔표 6〕 CW별 탐지율 (PLC = 40%) 기존모델

CW	DR (%)	FPR (%)
80	62.15	0.18
50	66.32	0.23
40	68.03	0.29
30	69.79	0.36
10	74.49	0.45

먼저 CW와 PLC와의 상관관계를 살펴보면 PLC가 작아질수록 매우 좋은 DR을 나타냄을 보이지만, FPR의 결과가 안 좋아짐을 알 수 있게 된다.

기존모델 결과에서의 특이한 점은 DR이 상당히 개선되었음을 알 수 있는데, 이는 일단 Training Data Set과 Testing Data Set에서의 http connection 데이터가 차지하는 비중이 약 80% 정도로 상당히 크기 때문이며, Data Set 이 TCP Service 만 가지고 있기 때문에 결과적으로 TCP 클러스터만 만들어진 후 탐지를 실행한 결과를 가지게 됨을 의미한다. 이로부터 Data Set 의 분포가 서비스별로 큰 편차를 나타내지 않았을 경우 기존모델에서의 침입탐지율의 저하를 가져옴을 예상케 한다.

〔표 7〕 PLC별 탐지율 (CW = 40) 제안모델

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

〔표 8〕 CW별 탐지율 (PLC = 40%) 제안모델

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

제안 모델에서는 PLC를 40%로 정하고 CW의 크기를 조정해 보지만 특별히 나쁜 결과를 제시해 주는 경우가 없었으며 CW가 30이하로 떨어지는 경우 FPR이 약간 악화되는 경향을 보이고 있으나 대체적으로 매우 높은 침입탐지율을 보여주고 있다.

4.3 실험결과 분석

기존의 클러스터 모델은 네트워으로 들어오는 모든 데이터에 대한 정상적인 클러스터와 비정상적인 클러스터를 가지고 이를 비교하여 침입탐지에 이용했으나 결과에서 DR, FPR 둘 다를 만족할만한 경우를 찾기가 쉽지 않았으므로 제안모델에서는 네트워으로 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 차이에

서 비롯되었다고 말 할 수 있겠다. 기존모델이 훨씬 넓은 클러스터 분포를 가지기 때문에 제안모델보다 작은 PLC를 가지는 것이 적절했다.

기존모델에서 FPR 수치는 만족할만 하다고 할 수 있겠으나, 실제 상용제품에 적용할수 있을만한 수준의 DR을 뒷받침 해주기에는 다소 무리가 있는 수치라고 말 할 수 있다.

그러나, 제안모델에서 제시한 서비스별 클러스터링의 실험결과는 실제 상용제품에 오용탐지 시스템 없이 독자적인 비정상 침입탐지 시스템이 쓰일 수 있을만큼의 DR, FPR의 현저한 탐지율 향상을 보여주었다.

Ⅲ. 결 론

네트웍 데이터 전체를 정상적인 클러스터와 비정상적인 클러스터로 잡은 기존의 연구와 비교해 볼 때 서비스별로 구분된 정상, 비정상 클러스터링은 DR, FPR 모두에서 만족할 만한 결과를 나타내었으며 무엇보다도 기존의 연구결과와 비교하여 두드러진 침입탐지율의 향상을 나타낼 수 있었다.

트레이닝 데이터 집합을 통한 클러스터링은 기존의 연구와 비교해 보아서 클러스터링 과정의 성능이 좋다고 할 수는 없지만, 일단 Training Set을 기반으로 한 클러스터가 만들어지고 나면 이것을 기반으로 한 탐지시의 성능은 차이가 없다고 할 수 있다.

현재 제안모델에서는 TCP 프로토콜 서비스만 테스트를 해보았으나 다른 프로토콜의 서비스에 적용을 하는 경우에도 우수한 침입 탐지율의 결과가 기대된다.

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 한다.

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 있을 것이다.

참 고 문 헌

[1] Leonid Portnoy. "Intrusion detection with unlabeled data using clustering" Data Mining Lab, Department of Computer Science, Columbia University, 2001. pp 5 - 17

[2] Alexander Hinnebrug and Daniel A. Keim. "Clustering methods for large databases: From the past to the future" SIGMOD99, 1999.

[3] "Kdd99 cup dataset http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html", 1999

[4] 김현희. 최근 데이터마이닝 알고리즘의 경향 및 추세. 2000. pp 8 - 11

[5] 유정각, "침입탐지 시스템의 개요 http://doit.ajou.ac.kr/~kagi/pub/intro_ids/", 2001

〈著 者 紹 介〉

류 희 재 (Hee-jae Ryu)
정회원

1999년 2월 : 경희대학교 섬유공학과 졸업
2003년 2월 : 아주대학교 정보통신공학 석사

예 홍 진 (Hong-jin Yeh)
정회원

1986년 1월 : 서울대학교 수학교육
1988년 1월 : 아주대학교 전자계산 석사
1990년 1월 : Grenoble1 대학교 응용수학, D.E.A
1993년 1월 : Lyon1 대학교 전자계산 박사
1993년 1월~현재 : 아주대학교 정보통신전문대학원 부교수