비정상행위 탐지를 위한 사용자 정상행위 클러스터링 기법

오 상 현† · 이 원 석††

요 약

사용자 비정상 행위를 탐지하기 위해서 기존의 연구들은 주로 통계적 기법을 이용해 왔다. 그러나 이들 연구들은 주로 사용자의 평균적인 행위를 분석하기 때문에 사용자의 비정상행위가 정확하게 탐지될 수 없다. 본 논문에서는 사용자의 정상행위를 모델링하는 새로운 클러스터링 방법을 제안한다. 클러스터링은 분석 환경에서 임의 개수의 빈발 영역을 식별할 수 있기 때문에 통계적 기법에서의 부정확한 모델링 방법을 개선할 수 있다. 빈발 공통 지식은 트랜잭션 단위로 발생되는 유사 데이터 객체들의 빈도수와 각 트랜잭션에 포함된 유사 데이터 객체들의 반복 비율로 나타낼 수 있다. 이와 더불어, 제안된 방법은 공통 지식을 축약된 프로파일로 유지하는 방법을 설명한다. 따라서 생성된 프로파일을 이용하여 온라인 트랜잭션에서의 비정상 행위를 쉽게 탐지할 수 있다.

Clustering Normal User Behavior for Anomaly Intrusion Detection

Sang Hyun Oh† · Won Suk Lee††

ABSTRACT

For detecting an intrusion based on the anomaly of a user's activities, previous works are concentrated on statistical techniques in order to analyze an audit data set. However, since they mainly analyze the average behavior of a user's activities, some anomalies can be detected inaccurately. In this paper, a new clustering algorithm for modeling the normal pattern of a user's activities is proposed. Since clustering can identify an arbitrary number of dense ranges in an analysis domain, it can eliminate the inaccuracy caused by statistical analysis. Also, clustering can be used to model common knowledge occurring frequently in a set of transactions. Consequently, the common activities of a user can be found more accurately. The common knowledge is represented by the occurrence frequency of similar data objects by the unit of a transaction as well as the common repetitive ratio of similar data objects in each transaction. Furthermore, the proposed method also addresses how to maintain identified common knowledge as a concise profile. As a result, the profile can be used to detect any anomalous behavior in an online transaction.

키워드 : 침입 탐지(Intrusion Detection), 비정상 행위 탐지(Anomaly Detection), 데이터마이닝(Data Mining), 클러스터링(Clustering), 사용자 프로파일링(User Profiling)

1. 서 론

컴퓨터와 통신 기술의 발달로 컴퓨터 시스템과 관련된 예기치 않은 침입 및 범죄에 의한 피해가 급증하고 있다. 침입이란 권한이 없는 사용자가 일으키는 문제 또는 어떤 사용자가 자신의 권한을 남용하는 행위로 정의될 수 있다. 즉, 시스템 자원의 신뢰성, 안전성, 무결성에 위배되는 행위를 침입으로 간주될 수 있다. 본 논문에서 침입은 비정상적인 사용자 행위뿐만 아니라 사용자 자신의 권한을 벗어나는 행위로 정의된다. 침입 탐지 시스템은 보통 호스트 기반과 네트워크 기반으로 나눌 수 있다. 호스트기반 침입 탐지 시스템은 하나의 호스트로부터의 획득된 정보를 이용하여 침입을 탐지하는 반면 네트워크 기반 침입 탐지 시스템은 네트워크 상의 데이터 트래픽을 감시함으로써 침입 정보를 획득한다. 한편, 침입 탐지 모델은 오용 탐지 모델[1-3]과 비정상행위 탐지 모델[4-7]로 분류된다. 오용탐지 모델은 침입 대상 호스트에서 알려진 취약점을 이용한다. 하지만, 침입 기술이 보다 복잡하게 변하고 많은 침입 방법이 새로 개발되고 있기 때문에 침입 방법을 개별적으로 다루는 것은 시스템의 안전을 유지하는데 충분하지 않다. 이러한 문제를 해결하기 위해서, 비정상행위 탐지 모델이 연구되고 있다.

IDES[4], NIDES[5] 및 EMERALD[6]는 통계에 기반한 비정상행위 탐지 시스템이다. 통계적인 분석 방법의 장점은 통계적인 요약만을 포함하는 축약된 프로파일을 생성하기 때문에 실시간에 침입을 감시하기 위해 요구되는 컴퓨팅의 부담을 줄일 수 있다. 하지만 통계적인 분석은 사용자 정상행위 패턴의 다양한 행위를 통계적인 요약만으로 표현하기 때문에 부정확한 정상행위 패턴이 생성될 수 있다. 또한,

† 준 회 원 : 연세대학교 대학원 컴퓨터과학과
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
논문접수 : 2003년 6월 24일, 심사완료 : 2003년 10월 28일

통계적인 분석 방법은 빈도수가 작고 주기적인 사용자 행위를 모델링하는데 단점을 가지고 있다. 즉, 간헐적인 사용자 행위들이 주기적으로 발생된다면 이들은 사용자 정상행위의 중요한 정보가 될 수 있지만 서로간의 관련성을 고려하지 않고 저 빈도 행위들의 단위인 회소 카테고리로 관리된다. 따라서 효과적인 정상행위 모델링을 수행할 수 없다.

최근에는 대량의 데이터를 지능적이고 자동적으로 분석하기 위해서 사용자 정상행위를 데이터마이닝 기술을 이용해서 모델링하고 있다. 데이터마이닝 기법들 중에서 클러스터링은 데이터 집합을 클러스터라 하는 의미 있는 몇 개의 그룹으로 분할하는 과정이다. 즉, 클러스터링의 목적은 주어진 유사도 척도에 의해서 정의되는 유사 데이터 그룹들을 탐사하는 것이다. 따라서, 데이터 집합의 내재적인 그룹 및 구조를 식별할 수 있다. 그러나, 침입 탐지 환경에서, 사용자의 행위는 의미 단위인 트랜잭션 단위로 수집된다. 여기에서 트랜잭션이란 의미적으로 분할할 수 없는 데이터 집합을 말한다. 기존의 클러스터링 방법[8-14]에서는 이러한 트랜잭션 정보를 효과적으로 모델링할 수 없다. 즉, 기존의 클러스터링 방법에서는 클러스터링이 유사도에 기반하여 전체 데이터 집합을 트랜잭션의 구분 없이 몇 개의 데이터 집합으로 분류함으로써 트랜잭션에 공통적으로 발생되는 지식을 효과적으로 추출할 수 없다.

본 논문에서는 침입 탐지 환경에서 트랜잭션 집합에서 공통 특징을 분석하는 새로운 클러스터링 알고리즘을 제안한다. 본 논문에서 제안된 방법은 DBSCAN[8]과 같은 밀도기반 클러스터링 방법에 의해서 기술된다. DBSCAN에서는 주어진 영역안에 존재하는 객체들의 개수가 클러스터 생성시 중요한 척도인 반면 제안된 방법에서는 주어진 영역안에 포함된 서로 다른 트랜잭션의 개수가 중요한 척도가 된다. 본 논문에서 제안된 클러스터링 방법에서는 트랜잭션 집합에서 공통적으로 발생되는 유사 데이터들의 특징들을 추출하는데 사용할 수 있다. 이러한 특징들 중 하나가 연관 마이닝에서 빈발 항목집합과 유사한 빈발 범위(frequent range)이다. 따라서 연관 마이닝에서 사용되는 지지도(support) 개념이 공통 지식을 추출하기 위해서 클러스터링에 적용되어야 한다. 트랜잭션 집합에서 공통지식 추출은 새로운 트랜잭션이 과거 트랜잭션 집합에 내재된 지식과 비교할 때 유용하다. 예를 들어, 사용자의 정상 행위를 벗어난 비정상행위 탐지 및 위성 영상에서 변화를 탐지할 수 있다. 클러스터가 과거 트랜잭션 집합의 공통 특징에 의해서 식별되기 때문에 새로운 트랜잭션과 클러스터사이의 차이를 쉽게 식별할 수 있다. 제안된 방법에서 클러스터링의 목적이 더 이상 데이터 분류가 아니기 때문에 식별된 클러스터의 특징들은 프로파일로 간결하게 축약될 수 있다. 따라서, 새로 수집된 트랜잭션에 포함된 비정상행위를 쉽게 탐지할 수 있다. 이와 더불어, 트랜잭션 집합에서는 또다른 종류의 공통 지식이 모델링 될 수 있다. 즉, 각 트랜잭션에서 유사한 반복 비율을 갖는 데이터 범위가 존재한다면 새로운 트랜잭션에서 데이터 차이를 발견하는 중요한 공통지식으로 모델링 될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 다양한 침입 탐지 기법들을 설명하고 3장에서는 클러스터링 기반으로 정상 행위 패턴을 생성하는 방법이 제안된다. 4장에서는 비정상행위 탐지 방법이 설명되고 5장에서는 제안된 비정상행위 탐지 방법의 효과를 분석하기 위한 실험 결과가 설명된다. 마지막으로 6장에서 결론을 맺는다.

2. 기존 연구

오용 탐지 모델을 위해서 몇몇 연구들이 전문가 시스템(expert system)[1], 상태 전이 분석(state transition model)[2], 모델 기반 기법(model based technique)[3]등을 이용하였다. 전문가 시스템 방법에서는 침입 패턴들이 "if-then" 규칙 형태로 표현된다. 따라서 사용자 행위들이 특정 침입 규칙에 일치하면 침입으로 간주된다. STAT 및 USTAT와 같은 상태전이 분석 모델에서는 침입 패턴들이 상태 전이 그래프에 의해서 표현된다. 만일 어떤 행위가 시작 상태에서 여러 중간 상태를 거쳐서 마지막 상태까지 도달하게 되면 그 행위는 침입으로 간주된다. 모델 기반 기법에서는 사용자 행위의 시나리오들의 집합으로 표현된다. 기본적으로 모델 기반 기법에서는 예측자(predictor), 계획자(planner), 분석자(analyzer) 모듈로 이루어진다. 예측자는 다음 상황에서 발생될 시나리오 모델을 예측하기 위해서 가설을 세운다. 계획자는 이 가설을 감사 데이타에서 표현될 수 있는 형태로 변환한다. 분석자에서는 감사 데이터가 시나리오 집합에 매치되는지를 검사함으로써 침입을 탐지한다.

오용탐지 모델의 단점 중 하나는 침입 탐지가 잘 알려진 침입 방법에 기반한다는 것이다. 즉, 새로 개발된 침입 방법에 의해서 공격 받게 되면 침입을 탐지할 수 없다. 비정상행위 탐지 모델은 이러한 단점을 극복할 수 있으며 통계적인 분석 기법과 예측 패턴 생성 기법으로 분류된다. 통계적인 분석 방법은 사용자의 과거 행위를 통계적인 프로파일로 유지하고 이 프로파일에 일치하지 않는 어떤 행위라도 비정상행위로 간주된다. 통계적인 분석 방법의 대표적인 시스템으로 SRI에서 개발된 IDES[4], NIDES[5] 및 EMERALD[6]등이 있다. NIDES는 IDES의 개선된 버전으로 오용탐지를 위한 규칙 기반 기법과 비정상행위 탐지를 위한 통계적 기법을 포함한다. NIDES에서 비정상행위를 탐지하기 위해서 사용자의 과거 행위들은 다양한 판정요소로 분류되고 각 판정요소에 대해서 통계적 요약을 포함하는 장기 프로파일(long term profile)을 생성한다. 비정상행위 탐지를 위해서 그래서 온라인 사용자 행위의 정보가 단기 프로파

일(short term profile)로 생성되고 이를 장기 프로파일과 비교한다. 만일 두 프로파일의 차이가 충분히 크면 온라인 사용자 행위는 비정상행위로 간주된다. NIDES와 유사한 EMERALD 시스템에서는 침입 탐지의 대상을 호스트 기반에서 네트워크 환경으로 확대하였다.

예측 패턴 생성 기법[7]에서는 사건의 순서가 불규칙하지 않고 특정 패턴을 따른다고 가정한다. 이러한 접근 방법은 정상 사용자 행위 패턴을 나타내는 시간 기반 규칙을 이용한다. 규칙들은 학습단계에서 동적으로 수정되며 관심 있는 규칙들만이 시스템에 남게 된다. 따라서 사용자의 행위들이 순서적으로 규칙의 왼쪽 부분에 매치되고 이후에 발생되는 행위들이 규칙에 의해서 예측되는 부분에서 벗어나면 비정상행위로 간주된다.

클러스터링 기법에는 분할 클러스터링(Partitional Clustering), 계층적 클러스터링(Hierarchical Clustering), 밀도기반 클러스터링(Density based Clustering) 및 격자 기반 클러스터링(Grid based Clustering)으로 분류된다. 분할 클러스터링은 미리 정의된 개수의 클러스터에 따라서 데이터를 분할하고 몇 가지 척도(criterion)함수에 의해서 이들을 평가하여 척도 함수가 가장 작게 되는 분할을 클러스터로 생성하게 된다. 분할 클러스터링 방법에는 k-means[9] 및 k-medoid[10]등의 방법이 있다. 계층적 클러스터링 기법은 미리 정해진 개수의 클러스터들이 생성될 때까지 객체 또는 클러스터들을 계층적으로 병합하는 클러스터링 기법이다. 대표적인 계층적 클러스터링 기법과 관련된 연구에는 BIRCH[11] 및 CURE[12] 등이 있다. 밀도기반 클러스터링 기법은 주어진 영역에 적정 밀도 이상의 데이터가 포함되어 있을 때 이 영역을 클러스터로 생성하는 방법이다. 밀도기반 클러스터링의 장점은 임의 형태의 클러스터를 탐사할 수 있고 잡음 처리가 쉬우며 한번의 데이터 스캔만으로 클러스터링이 가능하다. 대표적인 밀도기반 클러스터링 방법은 DBSCAN[8]이다. 격자 기반 클러스터링 기법은 영역을 균일한 부분 영역으로 분할하고 데이터가 존재하는 부분 영역들을 병합하여 클러스터를 생성한다. 격자 기반 클러스터링은 시간 복잡도가 상당히 효율적인 반면, 클러스터의 정확도가 부분 영역의 크기에 많은 영향을 받게 된다. 격자 기반 클러스터링에는 STING[13] 및 CLIQUE[14] 등의 연구가 있다.

기존의 클러스터링 방법에서는 단순히 데이터가 밀집되어 있는 영역에 대한 모델링을 수행하였다. 이들 연구들은 데이터 사이의 응집 척도에 따라서 같은 클러스터로 모델링될 지를 결정하였다. 반면, 본 논문에서는 연관 규칙 탐사[15]와 순차 패턴 탐사[16]에서와 같이 트랜잭션 데이터에 대해서 트랜잭션 정보를 데이터 모델링에 적용하였다. 따라서, 데이터간의 응집 척도뿐만 아니라 데이터의 빈발 정도에 따라서 데이터 모델링이 수행된다. 한편, 기존 연구에서는 생성된 클러스터간에 데이터 특성에 따른 차이를 구분하지 않았다. 즉, 생성된 클러스터는 영역을 대표하는 데이터로만 인식되기 때문에 데이터의 빈발성 및 반복 비율에 따른 데이터 특성이 고려되지 않았다. 반면 본 논문에서는 트랜잭션 기반으로 클러스터가 생성되고 새로 수집된 트랜잭션과의 차이 비교가 수행되기 때문에 클러스터에 포함된 데이터의 빈발성 및 데이터의 반복적인 특성이 모델링된다. 기존의 클러스터링이 패턴인식, 영역 탐색 등과 같은 활용 분야에서 클러스터 탐색에 초점을 맞추고 있지만 본 논문에서 제안하는 클러스터링 알고리즘은 침입 탐지 및 지형 변화 탐지와 같이 생성된 클러스터들을 이용하여 새로 수집된 트랜잭션 데이터와의 차이 비교와 같은 응용 분야에서 활용될 수 있다.

3. 사용자 정상행위 모델링

사용자의 행위는 다양한 특징들에 기반하여 클러스터링에 의해서 모델링 될 수 있다. 따라서 새로운 사용자의 행위에 대한 비정상 행위의 정도는 사용자의 과거 행위들로부터 추출된 클러스터를 생성함으로써 파악될 수 있다. 시스템 로그 데이터로부터 사용자 행위의 특징을 나타내기 위해서 다양한 특징들이 고려될 수 있다. 예를 들어 CPU 사용량, 시스템 콜의 반복횟수, 파일 접근 횟수 등이 사용자 행위의 특징으로 사용될 수 있다.

클러스터링을 위한 공간이 일 차원 이상인 데이터 집합을 분류하기 위해서 다차원 클러스터링 방법[8-14]들을 사용할 수 있다. 하지만, 다차원 공간에서 불규칙한 형태의 클러스터를 간결한 프로파일로 표현하는 것은 매우 어려운 과제이다. 다차원 데이터에 대한 프로파일을 생성하기 위해서 다음의 두 가지 방법을 고려할 수 있다. 첫 번째 방법은 모든 데이터 객체들을 각 좌표 축에 투영(projection)하고 각 차원에서 일 차원 클러스터링을 수행하여 차원별로 생성된 클러스터들에 대한 프로파일을 생성하는 것이다. 두 번째 방법은 다차원 클러스터링 방법을 이용하여 다차원 클러스터를 생성하고 생성된 클러스터를 각 좌표축에 투영하여 간결한 프로파일을 생성하는 것이다. 따라서 두 번째 방법이 보다 정확하게 클러스터의 특성들을 축약할 수 있다. 하지만 본 논문에서는 다차원 클러스터링 방법 자체가 주요 주제가 아님으로 단순화를 위해서 첫 번째 방법을 이용한 클러스터링과 프로파일 생성 방법을 설명한다.

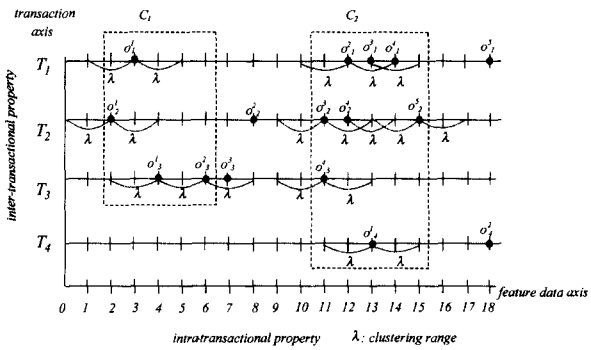
DBSCAN에서는 클러스터링 수행과정에서 유사 데이터 객체의 개수에 기반하여 클러스터가 확장된다. 반면, 제안된 방법에서는 트랜잭션 집합에서 공통 지식을 추출하기 위해서 각 클러스터는 서로 다른 트랜잭션의 개수에 기반하여 확장된다. 사용자 정의 클러스터링 범위 는 데이터 객체들간의 유사도 척도로 사용된다. 즉, 각 데이터 객체의

유사 데이터 그룹은 이 객체의 클러스터링 범위 안에 존재하는 데이터 객체들로 형성된다. 유사 데이터 그룹의 트랜잭션 수는 이 그룹에 포함된 서로 다른 트랜잭션의 개수를 나타낸다. 사용자 정의 최소 트랜잭션 수 *MinCnt*는 데이터 그룹의 빈발성 여부를 나타내는 척도로 사용된다. 따라서, 클러스터링에서 최소지지도는 전체 트랜잭션에 대해서 최소 트랜잭션 수의 비율로 구할 수 있고 지지도는 전체 트랜잭션의 수에 대해서 이 그룹에 나타난 트랜잭션 수의 비율로 정의된다. 만일 데이터 그룹의 지지도가 최소 지지도 이상이면 이 그룹은 클러스터로 생성된다. 그렇지 않으면 잡음(noise)로 처리된다.

기존의 클러스터링 방법과 달리, 제안된 방법은 트랜잭션 집합을 트랜잭션 축(transaction axis) 및 데이터 축(data axis)으로 모델링한다. 데이터 축에서는 문제 영역의 데이터 공간에 포함된 데이터 객체들간의 유사도가 모델링된다. 반면, 트랜잭션 축에서는 트랜잭션 내 특성(inter-transactional property) 및 트랜잭션간 특성(intra-transactional property)이 모델링된다. 트랜잭션간 특성으로 트랜잭션 집합에서 지지도 정보가 모델링되고 트랜잭션내 특성은 각 트랜잭션에서 유사 데이터 객체의 반복 비율로 모델링된다.

본 논문에서 감사 데이터(audit data)에 포함되는 트랜잭션은 다차원 공간에 존재하는 데이터 객체 집합으로 정의된다. 각 차원의 데이터 영역은 실수와 같은 연속성을 갖는다. TD를 트랜잭션 집합이라 하며 TD = $\{T_1, T_2, \cdots, T_n\}$와 같이 표현된다. TD^k는 TD에서 k번째 특징에 관련된 트랜잭션들의 집합이고 $|TD^k|$는 k번째 특징에 대한 로그에 포함된 트랜잭션의 개수를 나타낸다. 트랜잭션 T_i는 데이터 객체 집합이며 $T_i = \{o_i^1, o_i^2, \cdots, o_i^k\}$와 같이 표현되고 T_i에 포함된 데이터 객체의 수는 $|T_i|$와 같이 나타낸다. D^k는 TD^k에 포함된 데이터 객체들의 집합을 나타낸다.

(그림 1)은 제안된 클러스터링 방법이 수행되는 과정을 보여준다. 최소 트랜잭션 수와 클러스터링 범위가 각각 *MinCnt* = 3 및 λ = 2로 설정됐다고 하자. 클러스터링은 데이터 집합 D에서 가장 작은 값을 갖는 o^1_2로부터 시작된다. o^1_1, o^1_2, o^1_3가 o^1_2의 클러스터링 범위 내에 존재하기 때문에 이들은 o^1_2의 그룹으로 형성된다. 따라서 이 그룹의 트랜잭션 수가 MinCnt와 같기 때문에 이 그룹에 대해서 클러스터 C_1이 생성된다. 만일 클러스터 C_1와 유사한 데이터 객체들이 더 존재하면 클러스터를 확장시킬 수 있다. 즉, o^1_3의 데이터 그룹의 트랜잭션 수가 MinCnt와 같으므로 클러스터 C_1은 o^2_3로 확장된다. 그러나 o^2_3의 데이터 그룹의 트랜잭션 수가 MinCnt보다 작기 때문에 클러스터 C_1의 확장은 중단되고 남아 있는 데이터 객체들에 대해서 같은 방법으로 새로운 클러스터를 탐색한다. 한편, 각 클러스터의 객체 반복 비율은 트랜잭션에 포함된 전체 데이터 객체 수에 대해서 클러스터에 포함된 데이터 객체의 수로 정의된다. (그림 1)의 각 트랜잭션에 대해서, 클러스터 C_1의 객체 반복 비율은 각각 1/5, 1/5 및 2/4와 같다. 따라서, C_1의 평균 객체 반복 비율은 (1/5 + 1/5 + 2/4)/3 = 0.3와 같이 계산된다. 또한 C_2의 평균 객체 반복 비율은 (3/5 + 3/5 + 1/4 + 1/2)/4 = 0.4875와 같이 계산된다.



(그림 1) 클러스터링 개요

본 논문에서 데이터 객체 o^j_i에 대해서 유사 데이터 그룹은 $N(o^j_i) = \{o^k_m \in D \mid |v_k(o^j_i)\ v_k(o^k_m)| \le \lambda,\ 1 \le m \le |TD^k|\}$와 같이 정의된다. 이때 $v_k(o^j_i)$는 트랜잭션 T_i내에 j번째 데이터 객체의 k번째 특징값을 나타낸다. 객체 o^j_i에 대해서, 만일 $N(o^j_i)$의 지지도가 주어진 최소지지도 보다 크거나 같으면 $v_k(o^j_i)$는 $N(o^j_i)$의 핵심 값(core value)이 된다. 클러스터의 최소 핵심 값(minimum core value)과 최대 핵심 값(maximum core value)은 각각 클러스터에서 가장 작은 핵심 값 및 가장 큰 핵심 값을 나타낸다. (그림 1)에서 클러스터 C_1의 핵심 값은 2, 3, 4이므로 최소 및 최대 핵심 값은 각각 2, 4와 같다.

[정의 1] 두 인접한 클러스터 A와 B에 대해서, v^A_{max}와 v^B_{min} ($v^A_{max} < v^B_{min}$)를 각각 클러스터 A의 최대 핵심 값과 클러스터 B의 최소 핵심 값이라 할 때, $v^B_{min} - v^A_{max}$이면 두 클러스터 A와 B는 하나의 클러스터로 병합된다. □

[정의 1]은 두 인접한 클러스터들이 병합되기 위해서 두 클러스터의 핵심 값들이 범위 내에 존재해야 함을 나타낸다. 예를 들어, (그림 1)에서 데이터 객체 o^3_2 와 o^5_2에 대한 초기 클러스터가 각각 다음과 같다.

$N_\lambda(o^3_2) = \{o^2_1, o^3_1, o^3_2, o^4_2, o^4_3, o^1_4\}$
: core value set = {11, 12, 13}
$N_\lambda(o^5_2) = \{o^3_1, o^4_1, o^5_2, o^1_4\}$
: core value set = {13, 14, 15}

이때, $N(o^3_2)$의 최대 핵심 값은 $N(o^5_2)$의 최소 핵심 값과 같으므로 [정의 1]을 만족한다. 따라서 두 클러스터 $N(o^3_2)$와 $N(o^5_2)$는 병합된다. 하지만, o^3_2와 o^2_3에 의한 데이터 그룹들은 [정의 1]을 만족하지 않으므로 $N(o^3_2)$와 $N(o^2_3)$는 병

합될 수 없다. 결과적으로 (그림 1)에 대해서 다음 두 개의 클러스터가 생성된다.

$C_1 = \{o^1_1, o^1_2, o^1_3, o^2_3\}$
$C_2 = \{o^2_1, o^3_1, o^4_1, o^3_2, o^4_2, o^5_2, o^4_3, o^1_4\}$

본 논문에서 제안된 알고리즘은 최소 지지도 *minsup*와 클러스터링 범위 가 주어졌을 때 클러스터 집합을 탐사한다. 만일 데이터 집합의 지지도가 *minsup*보다 크거나 같으면 이 데이터 그룹은 클러스터가 된다. 생성된 클러스터는 인접한 데이터 객체들을 병합하면서 확장된다. 클러스터링 과정에서 각 데이터 객체의 현재 상태를 나타내기 위한 상태 표시자가 각 객체에 포함된다. 데이터 객체의 상태는 *미분류(unclassified)*, *잡음(noise)*, *클러스터 식별자(cluster identifier)*로 나타낸다. 클러스터링 수행 전에 D^k에 포함된 모든 데이터 객체들은 미 분류로 표시되고 클러스터링 과정에서 객체의 상태에 잡음 또는 클러스터 식별자가 부여된다. 제안된 알고리즘의 자세한 과정은 다음과 같다.

Step 1 : D^k를 오름차순으로 정렬한다.
Step 2 : D^k에서 미 분류인 데이터 객체들 중 가장 작은 값을 가지는 객체 o^j_i를 선택한다. 만일 이러한 객체가 존재하지 않으면 알고리즘이 종료된다.
Step 3 : 데이터 객체 o^j_i에 대해서 $N_\lambda(o^j_i)$를 얻은 후 $N_\lambda(o^j_i)$의 지지도가 *minsup*보다 작으면 o^j_i의 상태를 잡음으로 표시해서 Step2를 수행한다. 그렇지 않으면 $N_\lambda(o^j_i)$의 모든 데이터 객체의 상태를 새로운 클러스터 식별자가 부여된다.
Step 4 : $N_\lambda(o^j_i)$에서 최대 핵심 값을 갖는 데이터 객체 o를 선택한다. 만일 $v_k(o) = v_k(o^j_i)$이면 Step 2를 수행한다.
Step 5 : 객체 o를 *current_object*에 설정하고 N(*current_object*)를 얻는다. N(*current_object*)에 포함되는 전체 데이터 객체의 상태를 현재의 클러스터 식별자를 부여한다.
Step 6 : N_λ(*current_object*)에서 최대 핵심 값을 갖는 데이터 객체를 선택한다. 만일 v_k(*current_object*) < $v_k(o)$이면, Step5을 수행하고 그렇지 않으면 Step 2를 수행한다.

클러스터의 특성을 축약하기 위해서 [정의 2]와 같은 클러스터의 속성들을 사용한다.

[정의 2] 클러스터 C^k의 속성(Properties)
클러스터 C^k에 대해서, $O_C{}^k \subseteq D^k$는 클러스터 C^k에 포함된 데이터 객체 집합을 나타낸다. 클러스터 C^k의 속성들은 C^k ***(center, cdev, min, max, tcount, ratio, rdev)***로 구성된다.
① **min(C^k), max(C^k)** : 클러스터의 범위는 $O_C{}^k$에 포함된 데이터 객체들 중 최소 값 min(C^k) 및 최대 값 max(C^k)에 의해서 나타낸다.
② **tcount(C^k)** : tcount(C^k)는 $O_C{}^k$에 포함되어있는 서로 다른 트랜잭션의 개수이며 클러스터의 지지도는 support(C^k) = tcount(C^k)/$|TD^k|$와 같이 계산된다.
③ **center(C^k), cdev(C^k)** : $avg_i(C^k)$을 $O_C{}^k$에 포함된 데이터 객체들 중에서 T_i에 포함된 데이터 객체들의 평균이라 하면 클러스터 C^k의 중심값 center(C^k)는 다음과 같이 계산된다.

$$\text{center}(C^k) = \sum_{i=1}^{|TD^k|} \text{avg}_i(C^k)/\text{tcount}(C^k)$$

cdev(C^k)는 center(C^k)에 대한 표준 편차를 나타낸다.
④ **ratio(C^k), rdev(C^k)** : ratio(C^k)는 $O_C{}^k$에 포함된 데이터 객체들의 트랜잭션 별 평균 객체 반복 비율이고 rdev(C^k)는 ratio(C^k)에 대한 표준 편차를 나타낸다. $r_i(C^k)$를 트랜잭션 T_i에서의 객체 반복 비율이라 하면 $r_i(C^k) = |S_i|/|T_i|$와 같이 계산된다. $|S_i|$는 트랜잭션 T_i중에서 $O_C{}^k$에 포함되는 데이터 객체들의 수를 나타낸다. 따라서 클러스터 C^k의 객체 반복 비율은 다음과 같이 계산된다.

$$\text{ratio}(C^k) = \sum_{i=1}^{|TD^k|} r_i(C^k)/\text{tcount}(C^k)$$

rdev(C^k)는 ratio(C^k)에 대한 표준 편차를 나타낸다. □

따라서 (그림 1)의 예에서 생성된 클러스터들의 속성은 다음과 같이 계산될 수 있다.

C_1 (3.33, 1.2481, 2, 6, 3, 0.3, 0.141)
C_2 (12.125, 2.8, 11, 15, 4, 0.4875, 0.143)

4. 비정상행위 탐지

4.1 프로파일 생성

프로파일은 빈발 공통지식 추출을 위한 클러스터링 결과를 포함하며 내부 및 외부 요약으로 구성된다. 내부 요약은 (internal summary) 각 클러스터의 속성들을 포함하고 외부 요약(external summary)은 클러스터 외부에 존재하는 잡음(noise)의 통계를 표현한다. 외부 요약은 두 가지 형태의 속성으로 구성된다. 첫번째는 외부 데이터 비율 *extratio*과 표준 편차인 *extratio_dev*이고 두 번째는 외부 거리 *extdist*와 표준편차 *extdist_dev*이다. 트랜잭션 데이터 집합에서 k번째 특징에 대해 *m*개의 클러스터가 생성되었을 때 $extratio^k$는 다음과 같이 트랜잭션별 평균 잡음 비율로 나타낸다.

$$\text{extratio}_i^k = \frac{1}{|TD^k|} \cdot \sum_{i=1}^{|TD^k|} \left(1 - \sum_{j=1}^{m} r_i(C_j^k)\right)$$
$$= 1 - \sum_{j=1}^{m} \text{ratio}(C_j^k) \cdot \text{support}(C_j^k)$$

따라서, (그림 1)의 예에서 트랜잭션별 평균 잡음 비율 extratio와 표준편차 extratio_dev는 다음과 같이 계산될 수 있다.

$$\begin{aligned}\text{extratio} &= 1 - \{\text{ratio}(C_1)\text{support}(C_1) + \text{ratio}(C_2)\text{support}(C_2)\} \\ &= 1\ 0.7125 = 0.2875\end{aligned}$$

$$\text{extratio\_dev} \cong 0.124$$

잡음 객체의 외부 거리는 프로파일에서 가장 가까운 클러스터와의 거리로 정의된다. 두 인접한 클러스터 A와 B (max(A) < min(B))가 주어졌을 때, 잡음 객체 o_i^j가 A와 B 사이에 존재하면 o_i^j는 두 클러스터들 중 하나를 거리 계산을 위해 선택될 수 있다. 이를 위해서 두 클러스터 사이의 상대적 평형점(relative equilibrium point) *rep*(*A*, *B*) 이 계산될 수 있다. 즉, rep(A, B)는 두 클러스터에 상대적으로 같은 거리의 위치를 나타내는 값이다. 만일 데이터 객체 o_i^j가 rep(A, B)보다 크면, 외부 거리는 클러스터 B의 최소값과의 차로 계산될 수 있고 그렇지 않으면 클러스터 A의 최대 값과 외부 거리가 계산된다.

[정의 3] 상대적 평형점

$f_C(\varepsilon)$를 데이터 객체 ε가 클러스터 C에 가까운 정도라 하자. 그러면 $f_C(\varepsilon)$는 클러스터 C의 지지도에 비례하고 ε와 클러스터 C사이의 외부거리와 반비례 한다. 즉, 클러스터 C의 지지도가 커지거나 ε가 클러스터 C에 가까워질 때 $f_C(\varepsilon)$가 커지게 된다. 따라서, $f_C(\varepsilon) = \frac{\text{support}(C)}{|V(\varepsilon) - \min(C) \text{ or } \max(C)|}$와 같이 계산될 수 있다. 이를 이용하여, 인접한 두 클러스터 A, B (max(A) < min(B))에 대해서, rep(A, B)는 $f_A(\varepsilon) = f_B(\varepsilon)$인 ξ의 값으로 나타낼 수 있다. ρ = support(B)/support(A)일 때, rep(A, B)는 다음과 같이 계산될 수 있다.

$$\text{rep}(A,B) = \frac{\min(B) + \rho \cdot \max(A)}{1+\rho} \qquad \square$$

[정의 3]에 의해서, 잡음 객체의 외부거리는 두 클러스터들 중 하나와 거리계산이 수행된다. P^k를 k번째 특징에 대한 프로파일이라 할 때 $dist(P, o_i^j)$를 데이터 객체 o_i^j와 [정의 3]에 의해서 P^k로부터 선택된 클러스터 C^k와의 외부거리를 나타내며 다음과 같이 계산된다.

$$\text{dist}(P^k, o_i^j) = \begin{cases} |v_k(o_i^j) - \min(C)| & \text{if } v_k(o_i^j) < \min(C) \\ |v_k(o_i^j) - \max(C)| & \text{if } v_k(o_i^j) < \max(C) \end{cases}$$

따라서, E_i^k를 T_i내의 잡음 객체들의 집합이라 하고 $|E_i^k|$를 E_i^k에 포함된 객체들의 개수라 하면, 프로파일 P^k에 대해서 트랜잭션 T_i의 외부 거리는 다음과 같이 계산된다.

$$\text{ex\_d}(P^k, T_i) = \frac{1}{|E_i^k|} \sum_{j=1}^{|E_i^k|} \text{dist}(P^k, o_i^j) \quad \text{where } o_i^j \in E_i^k$$

각 트랜잭션의 외부 거리에 기반하여 $\text{extdist} = \sum_{i=1}^{|TD^k|} \text{ex_d}(P^k, T_i)/|TD^k|$와 같이 계산된다. 따라서, (그림 1)의 예에서 트랜잭션 T1에 대한 외부거리는 $\text{ex_dist}(P, T_1) = |v(o_1^5) \max(C_2)| = 3$와 같이 계산된다. 마찬가지로 ex_dist(P, T_2), ex_dist(P, T_3) 및 ex_dist(P, T_4) 는 각각 2, 1 및 3와 같이 계산된다. 따라서 평균 외부거리 extdist와 표준편차 extdist_dev는 각각 2.25 및 0.829와 같이 계산된다.

4.2 데이터 차이

새로 수집된 트랜잭션에서 비정상행위를 탐지하기 위해서는 이 트랜잭션을 이미 생성된 프로파일과 비교하여 데이터 차이가 식별되어야 한다. 이러한 비교는 내부 및 외부 비정상행위로 표현된다. 내부 비정상행위(internal abnormality)는 프로파일의 내부 요약과 새로 수집된 트랜잭션 내에서 클러스터에 포함되는 데이터 객체들과의 차이를 나타내고 외부 비정상행위(external abnormality)는 프로파일의 외부 요약과 새로운 트랜잭션에서 잡음 객체들과의 차이를 나타낸다. 각 차이는 거리 차이(distance difference)와 비율 차이(ratio difference)로 분류된다. $MS = \{ID, IR, ED, ER\}$를 비정상행위를 위한 판정요소 집합이라 할 때 *ID*, *IR*, *ED* 및 *ER*는 각각 내부 거리 차이, 내부 비율 차이, 외부 거리 차이 및 외부 비율 차이를 나타낸다.

클러스터 C^k와 트랜잭션 T가 주어졌을 때, S는 트랜잭션 T 중에서 클러스터 C^k에 포함된 데이터 객체 집합을 나타내고 avg(C^k)는 S에 포함되어 있는 데이터 객체들의 평균값을 나타낸다. 따라서, 내부 거리 차이(internal distance difference)는 클러스터 C^k의 중심값 center(C^k)와 avg(C^k)의 차로 정의된다. 이때, 내부 거리 차이는 각 클러스터의 분포가 다르기 때문에 center(C^k)에 대한 표준편차로 정규화 되어야 한다. 마찬가지로 내부 비율 차이(internal ratio difference)도 ratio(C^k)에 대한 표준편차 rdev(C^k)에 의해서 정규화 되어야 한다.

$$\text{dist\_diff}_{in}(C^k, T_v) = \frac{|\text{center}(C^k) - \text{avg}_v(C^k)|}{\gamma \cdot \text{cdev}(C^k)} \quad \text{if } \text{codev}(C^k) \neq 0$$

$$\text{ratio\_diff}_{in}(C^k, T_v) = \frac{|\text{ratio}(C^k) - r_v(C^k)|}{\gamma \cdot \text{rdev}(C^k)} \quad \text{if } \text{rodev}(C^k) \neq 0$$

만일 클러스터 C^k에 대해서 cdev(C^k) 또는 rdev(C^k)가 0에 가까운 값이면 내부 차이가 매우 커지게 된다. 따라서 새로 수집된 트랜잭션에 대한 전체 데이터 차이는 이 클러스터에 의해서 상당히 많은 영향을 받게 된다. 이를 보완하기 위해서, 하나의 클러스터에 대한 내부 차이가 적정 값을 넘지 않도록 내부 차이의 최대값이 설정돼야 한다. 정규화 요소인 는 내부 차이에 대한 효과를 조절할 수 있는 사용자 정의 매개변수이다. 가 작아지면 데이터 차이의 효과가

증가하기 때문에 보다 세밀한 차이계산이 가능하다. 한편, 각 클러스터의 지지도가 서로 다르기 때문에 클러스터의 내부 차이 계산시 각 클러스터의 지지도를 내부 차이에 적용해야 된다. 또한 각 특징에서 생성되는 클러스터가 하나 이상이므로 전체 클러스터에 대한 데이터 차이는 각 클러스터에 대한 데이터차이들의 합으로 계산된다. 즉, 클러스터의 개수가 m일 때 모든 클러스터들에 대한 내부 거리 차이와 내부 비율 차이는 다음과 같이 계산된다.

$$\text{total\_dist}_{in}(P^k, T_v) = \sum_{i=1}^{m} \text{dist\_diff}_{in}(C_i^k, T_v) \cdot \text{support}(C_i^k)$$

$$\text{total\_ratio}_{in}(P^k, T_v) = \sum_{i=1}^{m} \text{ratio\_diff}_{in}(C_i^k, T_v) \cdot \text{support}(C_i^k)$$

이때, 각 특징에 대한 내부 거리 차이들을 이용하여 트랜잭션 T_v에 대한 전체 내부거리 차이는 모든 특징들의 내부 거리 차이의 평균으로 구할 수 있다. 이와 유사하게, 트랜잭션 T_v에 대한 전체 내부 비율 차이는 모든 특징들의 내부 비율 차이의 평균으로 구할 수 있다. 특징의 개수가 전체 n이고 각 특징들에 대한 프로파일 집합이 $\mathbf{P} = \{P^1, P^2, \cdots, P^n\}$라 할 때, 전체 내부 거리 차이와 전체 내부 비율 차이는 다음과 같이 계산된다.

$$\text{overall}_{ID}(\mathbf{P}, T_v) = \frac{1}{n} \sum_{k=1}^{n} \text{total\_diff}_{ex}(P^k, T_v)$$

$$\text{overall}_{IR}(\mathbf{P}, T_v) = \frac{1}{n} \sum_{k=1}^{n} \text{total\_ratio}_{ex}(P^k, T_v)$$

프로파일 P^k와 새로운 트랜잭션 T_v와의 외부 비정상행위도는 내부 비정상행위도 계산과 유사하다. $dist_diff_{ex}(P^k, T_v)$와 $ratio_diff_{ex}(P^k, T_v)$를 각각 외부 거리 차이(external distance difference) 및 외부 비율 차이(external ratio difference)라 하고 T_v에서 잡음 객체의 개수를 $|E^k_v|$라 할 때 외부 비정상행위도는 다음과 같이 계산된다.

$$\text{dist\_diff}_{ex}(P^k, T_v) = \frac{|\text{extdist}^k - \text{ex\_d}(P^k, T_v)}{\gamma \cdot \text{extdist\_dev}^k} \quad \text{if } \text{extdist\_dev}^k \neq 0$$

$$\text{ratio\_diff}_{ex}(P^k, T_v) = \frac{|\text{extratio}^k - |E^k_v| / |T_v||}{\gamma \cdot \text{extdist\_dev}^k} \quad \text{if } \text{extratio\_dev}^k \neq 0$$

내부 차이에서와 마찬가지로 외부차이가 적정 값을 넘지 않도록 하기 위해서 외부 차이의 최대값이 설정돼야 한다. 이때, 특징의 개수가 전체 n일 때 전체 외부 거리 차이와 전체 외부 비율 차이는 다음과 같이 계산된다.

$$\text{overall}_{ED}(\mathbf{P}, T_v) = \frac{1}{n} \sum_{k=1}^{n} \text{dist\_diff}_{ex}(P^k, T_v)$$

$$\text{overall}_{ER}(\mathbf{P}, T_v) = \frac{1}{n} \sum_{k=1}^{n} \text{ratio\_diff}_{ex}(P^k, T_v)$$

사용자의 온라인 트랜잭션에서 비정상행위의 정도를 결정하기 위해서 본 논문에서는 각 판정요소마다 서로 다른 비정상행위 레벨을 두며 이러한 레벨을 과거 사용자의 행위에 기반한 프로파일을 이용하여 설정 될 수 있다. 즉, 사용자의 행위에 대해서 두개의 레벨(green 및 red)을 설정하여 사용자의 행위를 분류한다. green 레벨은 사용자의 행위가 정상적임을 나타내며 red 레벨은 사용자의 행위가 비정상적임을 나타낸다. $\mu \in$ MS 및 온라인 트랜잭션 T_v에 대해서, 각 레벨의 범위는 평균 비정상행위도 $\mathbf{P}_\mu(TD)$와 이에 대한 표준편차 sd_μ로 다음과 같이 나타낼 수 있다. 여기에서 ξ는 정상행위의 범위를 조절하는 파라미터이다. 이때, *false alarm rate*는 전체 정상행위 트랜잭션 개수에 대해서 *red* 레벨에 포함되는 트랜잭션 개수의 비율을 나타내고 detection rate는 전체 비정상행위 트랜잭션 개수에 대해서 *red* 레벨에 포함되는 트랜잭션개수의 비율을 나타낸다.

$$\Phi_\mu(\text{TD}) = \frac{1}{|\text{TD}|} \cdot \sum_{i=1}^{|\text{TD}|} \text{overall}_\mu(\mathbf{P}, T_i)$$

$$sd_\mu = \sqrt{\frac{1}{|\text{TD}|} \cdot \sum_{i=1}^{|\text{TD}|} \text{overall}_\mu(\mathbf{P}, T_i)^2 - \Phi_\mu(\text{TD})^2}$$

- *green* : *if* $0 \leq overall_\mu(\mathbf{P}, T_v) \leq \Phi_\mu(TD) + sd_\mu \cdot \xi$
- *red* : *if* $\Phi_\mu(TD) + sd_\mu \cdot \xi < overall_\mu(\mathbf{P}, T_v)$.

5. 실험 결과

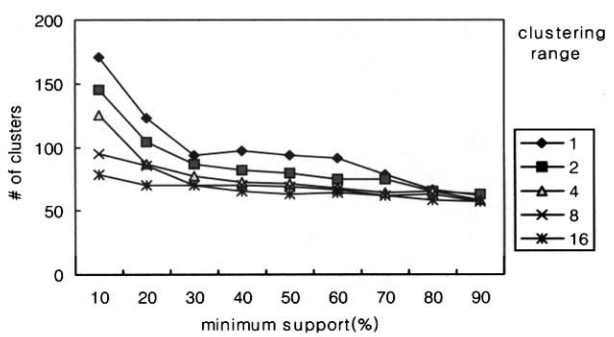
본 논문에서는 4장에서 제시한 수식을 이용하여 다양한 실험으로 통해 비정상행위 판정율을 향상시키도록 하였다. 모의 실험 데이타는 UNIX 기반의 Solaris 2.6을 사용하는 사용자에 대해서 두 달 동안의 데이타를 수집하여 사용자 정상행위 패턴을 생성하였다. 이를 위해서 UNIX시스템 기반에서 Solaris 2.6용 BSM(Basic Security Module)[17]을 활용하여 로그 데이타를 수집하였다. BSM은 C2 레벨의 보안(security)을 제공하는 툴로써, 228개의 커널 신호(signal)를 인식하고 감사 로그파일에 기록한다. 이러한 커널 신호들 중에서 52개의 신호들을 실험을 위한 특징으로 추출하였다. 본 논문에서 사용자의 로그인(Log-in)으로부터 로그아웃(Log-out)까지의 작업들의 집합인 세션을 트랜잭션으로 간주하여 실험하였다.

〈표 1〉 실험에서 사용된 데이터 집합

데이터 집합	트랜잭션 수	트랜잭션 평균 크기(객체 수)	데이터 집합의 크기(bytes)
DATA1	40	300	2.8M
DATA2	20	80	2.6M
DATA3	40	400	3.9M
ATTACK1	1	352	43K
ATTACK2	1	520	56K
ATTACK3	1	219	32K

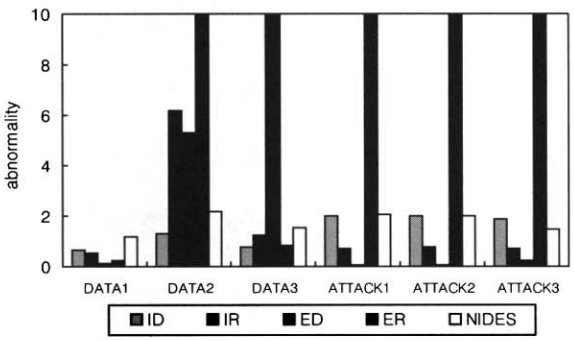
실험에서 사용된 데이터들은 <표 1>과 같다. DATA1 및 DATA3은 서로 다른 프로그래머에 의해서 생성된 로그 데이터이고 DATA2는 시스템 관리자에 의해서 생성된 로그 데이터다. ATTACK1은 버퍼 오버플로우(buffer overflow)에 의해서 생성된 데이터고 ATTACK2는 패스워드 추론(password guessing)에 의해서 생성된 데이터이다. 한편 ATTACK3에는 디렉토리 검색, 파일 삭제 및 파일 복사와 같은 침입자의 행위들이 포함되어 있다.

트랜잭션 단위의 모델링의 효과를 검증하기 위해서 본 논문에서 제안하는 알고리즘을 NIDES와 비교하였다. 제안된 알고리즘에서, 클러스터링의 결과는 최소지지도와 클러스터링 범위에 영향을 받는다. 만일 최소 지지도가 높게 설정되면 보다 빈발한 공통지식이 클러스터로 식별된다. 하지만 너무 높게 설정되면 공통 지식 중 일부가 잡음으로 처리될 수 있다. 한편, 클러스터링 범위가 크게 설정되면 크기가 큰 빈발 범위가 클러스터로 식별되기 때문에 클러스터링 범위를 작게 설정했을 때보다 클러스터의 정보가 보다 부정확해지게 된다.



(그림 2) DATA1로부터 생성된 클러스터의 개수

(그림 2)는 제안된 알고리즘을 이용하여 최소 지지도가 10~90%이고 클러스터링 범위가 1~16일 때 DATA1으로부터 생성된 클러스터의 개수를 나타낸다. 이 실험에서 최소 지지도와 클러스터링 범위가 증가함에 따라 클러스터의 개수가 작아짐을 알 수 있다. (그림 3)에서는 제안된 알고리즘의 성능을 NIDES와 비교하였다. 제안된 알고리즘에서는 네 가지 비정상행위도인 내부 거리 차이(ID : internal distance difference), 내부 비율 차이(IR : internal ratio difference), 외부 거리 차이(ED : external distance difference) 및 외부 비율 차이(ER : external ratio difference)를 사용하였다. 이 실험에서 최소 지지도와 클러스터링 범위는 각각 60% 및 2로 설정하였고 정규화 요소 γ는 3으로 설정하였다. 이 실험에서 DATA1의 네 가지 비정상행위도는 다른 데이터 집합보다 작게 나타났다. 이것은 실험에서 프로파일로 사용된 데이터 집합이 DATA1이기 때문이다. 또한 제안된 알고리즘은 다른 데이터 집합들에 대해서 NIDES에서 보다 더 큰 비정상행위도가 나타났다. DATA1과 DATA3가 프로그래머에 의한 로그 데이터들이지만 서로 다른 행위를 수행하였기 때문에 비정상행위도가 크게 나타났다. 한편, 시스템 공격을 수행한 데이터들에 대해서 외부 비율 차이가 크게 나타났다. 이것은 공격 로그 데이터들은 DATA1에서 생성된 클러스터들에 포함되는 행위들이 매우 작아지기 때문이다.



(그림 3) 제안된 방법의 성능

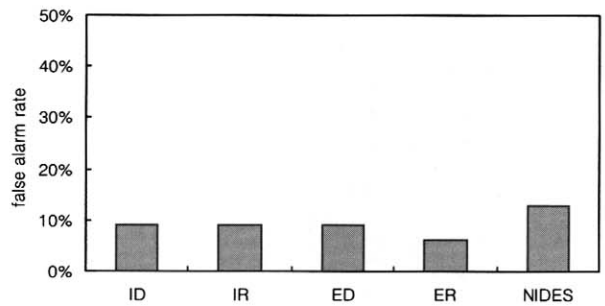
<표 2>는 DATA1의 비정상행위도에 대해서 다른 데이터 집합들에 대한 상대적 비정상행위도를 나타낸다. <표 2>에서 내부 거리 차이와 내부 비율 차이에 대한 상대적 비정상행위도는 외부 거리 차이와 외부 비율 차이에 대한 상대적 비정상행위도에 비해서 작은 값을 갖는다. 이것은 내부 비정상행위도를 계산하기 위해서 단지 클러스터에 포함되는 행위들만이 고려되기 때문이다. 즉, 비정상적인 행위의 대부분은 클러스터에 포함되지 않기 때문에 각 데이터 집합에 대한 외부 비정상행위도는 내부 비정상행위보다 커지게 된다.

〈표 2〉 상대적 비정상행위도

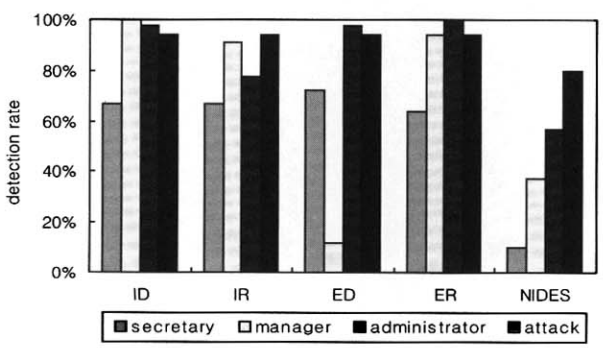
데이터 집합	ID	IR	ED	ER	NIDES
DATA1	1	1	1	1	1
DATA2	2.06	11.26	53.41	37.61	1.86
DATA3	1.21	2.25	100.70	3.23	1.29
ATTACK1	3.17	1.31	0.56	37.91	1.73
ATTACK2	3.18	1.39	0.56	37.91	1.70
ATTACK3	2.99	1.30	2.26	37.91	1.25

(그림 4)에서는 1998년 DARPA에서 수집된 BSM로그 데이터 [18]를 이용하여 제안된 방법을 통계적인 비정상행위 판정 기법인 NIDES와 비교한 결과이다. 수집된 로그 데이터에는 프로그래머(programmer), 비서(secretary), 매니저(manager), 시스템 관리자(System Administrator) 및 공격(Attack) 데이터가 포함되어 있다. 프로그래머는 vi편집기를 열어서 C언어 코드를 작성하고 이를 컴파일하는 작업 및 메일 전송, 유닉스 명령어등을 수행하였다. 비서는 메일 쓰기/전송 및 latex 화일 편집과 같은 작업을 수행하였고 매니저

는 메일 읽기/전송등을 수행하였다. 한편, 시스템 관리자는 시스템 관리에 필요한 명령어를 수행하였고 공격은 버퍼 오버플로 및 비정상행위가 수행되었다. 이 실험에서 프로그래머가 비정상행위 탐지를 위한 대상 사용자로 선정하였다. 위에서와 동일한 방법으로 클러스터링을 위한 최소지지도와 클러스터링 범위는 각각 5% 및 1로 설정하였고 정상행위 범위를 조절하는 파라미터 ξ는 2로 설정하였다. (그림 4)(a)는 제안된 방법과 NIDES의 *false alarm rate(FAR)*를 비교하였다. 제안된 방법에서는 모든 판정요소에 대한 *FAR*이 10%보다 작게 나타났지만 NIDES에서는 10%보다 조금 높게 나타났다. (그림 4)(b)는 프로그래머가 정상 사용자라고 했을 때 다른 사용자들에 대해서 비정상행위 탐지율(Detection rate)을 나타낸다. 이 그림에서 제안된 방법은 공격에 대해서 거의 100%의 탐지율을 보였지만 NIDES에서는 80% 정도의 탐지율만을 보였다.



(a) False alarm rates



(b) Detection rates

(그림 4) DARPA 로그데이타 결과(최소지지도 = 5%, 클러스터링 범위 = 1, ξ = 2)

6. 결 론

본 논문에서는 감사 데이터에서 공통 지식을 찾는 새로운 클러스터링 알고리즘이 제안되었다. 제안된 방법에서는 각 특징에 따라서 클러스터링이 수행되고 생성된 클러스터들에 대한 다양한 통계적인 정보를 프로파일로 모델링 하였다. 이를 위해서 본 논문에서는 두 가지 종류의 비정상행위도, 즉 내부 차이와 외부 차이를 제안하였다. 클러스터링에 의해서 각 특징은 빈발 영역(frequent range) 및 희소 영역(infrequent rage)으로 나뉘며 두 영역에 대한 정상행위는 거리 차이와 비율 차이로 분류된다. 결과적으로 사용자의 행위를 다양한 각도에서 분석할 수 있다. 반면, NIDES는 본 논문에서와 같이 사용자 행위의 다양한 면을 하나의 평균으로 유지하기 때문에 정상 사용자와 비정상 사용자간의 차이를 판별하기 어렵다. 이와 더불어 응용 영역에 따라서 데이터 분석의 정밀도를 제어하는 정규화 요소가 내부 및 외부 데이터 차이에 적용되었다.

참 고 문 헌

[1] B. Mukherjee, T. L. Heberlein and K. N. Kevitt, "Network intrusion Detection," IEEE Network, Vol.8, No.3, pp.26-41, May/June, 1994.

[2] K. Illgun, "USTAT : A Real-Time Intrusion Detection System for UNIX," in Proc. Of the 1993 Symposium Security and Privacy, pp.16-28, May, 1993.

[3] T. D. Garvey and Teresa, F. Lunt, "Model based intrusion detection," In Proc. Of the 14th National Computer Security Conference, pp.372-385, October, 1991.

[4] H. S. Javitz, A. Valdes, "The SRI IDES Statistical Anomaly Detector," In Proc. of the 1991 IEEE Symposium on Research in Security and Privacy, May, 1991.

[5] Harold S. Javitz and Alfonso Valdes, The NIDES Statistical Component Description and Justification, Annual report, SRI International, 333 Ravenwood Avenue, Menlo Park, CA 94025, March, 1994.

[6] Phillip A. Porras and Peter G. Neumann, "EMERALD : Event Monitoring Enabling Responses to Anomalous Live Disturbances," 20th NISSC, October, 1997.

[7] Henry S. Teng, Kaihu Chen and Stephen C. Lu, "Security Audit Trail Analysis Using Inductively Generated Predictive Rules," In Proceedings of the Sixth Conference on Artificial Intelligence Applications, Piscataway, New Jersey, pp.24-29, March, 1990.

[8] Martin Ester, Hans-Peter Kriegel, Sander, Michael Wimmer, Xiaowei Xu, "Incremental Clustering for Mining in a Data Warehousing Environment," Proceedings of the 24th VLDB Conference, New York, USA, 1998.

[9] MacQueen, J., "Some Methods for Classification and Analysis of Multivariate Observations," Proc. 5th Berkeley Symp., pp.281-297, 1967.

[10] Kaufman, L. and Rouseeuw, P., Finding Groups in Data : an Introduction to Cluster Analysis, John Wiley & Sons, 1990.

[11] Tian Zhang, Raghu Ramakrishnan and Miron Livny, "Birch : An Efficient data clustering method for very large da-

tabases," Proceedings for the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June, 1996.
[12] Sudipto Guha, Rajeev Rastogi and Kyuseok Shim, "CURE : An Efficient Clustering Algorithm for Large Databases," ACM SIGMOD International Conference on Management of Data, Seattle, Washington, 1998.
[13] W. Wang, J. Yang and R. Muntz, STING : A statistical information grid approach to spatial data mining, 1997.
[14] Rakesh Agrawal, Johannes Gehrke, Dimitrios Gunopulos, Prabhakar Raghavan, "Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications," Proc. of the ACM SIGMOD Int'l Conference on Management of Data, Seattle, Washington, June, 1998.
[15] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules," Proc. of the 20th Int'l Conference on Very Large Databases, Santiago, Chile, Sept., 1994.
[16] R. Agrawal, R. Srikant, "Mining Sequential Patterns," Proc. of the Int'l Conference on Data Engineering (ICDE), Taipei, Taiwan, March, 1995.
[17] Sun Microsystems. SunShield Basic Security Module Guide.
[18] http://www.ll.mit.edu/IST/ideval/index.html.

오 상 현

e-mail : osh @amadeus.yonsei.ac.kr
1996년 제주대학교 정보공학과
1998년 연세대학교 컴퓨터과학과 석사
1998년~현재 연세대학교 컴퓨터과학과 박사과정
관심분야 : 침입탐지 시스템, 데이터마이닝, 에이전트 시스템

이 원 석

e-mail : leewo@amadeus.yonsei.ac.kr
1985년 미국 보스턴 대학교 컴퓨터과학과 (학사)
1987년 미국 퍼듀 대학교 컴퓨터공학과 (석사)
1990년 미국 퍼듀 대학교 컴퓨터공학과 (박사)
1990년~1992년 삼성전자 선임 연구원
1993년~1999년 연세대학교 컴퓨터과학과 조교수
1999년~현재 연세대학교 컴퓨터과학과 부교수
관심분야 : 분산 데이터베이스, 멀티미디어 데이터베이스, 객체지향 시스템, 데이터마이닝