사용자 로그 스트림 클러스터링에 의한 실시간 침입탐지 기법

박남훈, 오상현, 이원석
연세대학교 컴퓨터과학과

Anomaly Intrusion Detection by Clustering Transactional Audit Streams in a Host Computer

Nam Hun Park, Sang Hyun Oh, Won Suk Lee
Yonsei University
E-mail : {zyonix, osh, leewo}@database.yonsei.ac.kr

요 약

침입탐지에 있어서 사용자 로그 분석은 중요한 주제로서, 기존의 연구들에서 클러스터링 기법들을 사용하여 저장된 사용자 로그들을 분석해왔다. 하지만, 이러한 방법은 고정된 사용자 패턴 분석에는 효율적이지만, 로그 스트림과 같이 무한히 생성되어 사용자 패턴이 변화하는 경우 변화하는 패턴을 분석할 수 없다. 본 연구에서는 무한히 생성되는 사용자 로그 스트림을 대상으로 실시간 침입탐지 방법을 제시한다. 사용자 로그의 정보는 사용자 행동에 대한 특성값으로 표현되어, 이러한 특성값들에 대해 실시간 데이터 스트림 클러스터링을 수행하여 이들을 클러스터로 분류한다. 각 클러스터는 사용자의 정상로그에 대한 특성값을 반영하게 되며, 그 결과 과거 사용자 로그에 대한 저장없이 새로운 로그 스트림을 지속적으로 분석할 수 있다. 결과적으로 사용자의 비정상행동을 실시간으로 탐지할 수 있으며, 이를 실험을 통해 평가하였다.

1. 서론

기존의 침입탐지 기법은 오용탐지[1]와 침입 탐지[2]으로 분류된다. 오용탐지에서는 대상 응용환경의 보안에 취약한 사용자의 오용가능한 부분을 분석하여 보완하는 방법이다. 하지만, 침입탐지에서는 항상 새로운 침입방법이 지속적으로 시도되므로 침입탐지에는 어려움이 있다. 이러한 주제로 기존의 침입탐지 연구[2]들이 진행되어 왔다. IDES[3], NIDES[4], EMERALD[5]는 사용자 로그에 대한 통계적 분석에 기반한 침입탐지 방법으로, CPU점유율, 시스템 호출 빈도 및 파일접근 횟수 등의 다양한 특성값들을 사용하였다. 일정횟수의 시스템 호출 빈도와 파일접근 횟수 등을 정상로그로 간주하였다.

침입탐지에 있어 사용 가능한 특성값의 수는 중요하지만, 사용자 행동은 주로 일부의 특성값에만

관련된다. 이러한 이유로, 사용자 로그는 각 특성값에 따라 분석할 필요가 있다. 전형적인 통계기반 침입탐지 기법들로 [3]가 있다. 통계기반 기법은 통계 요약 정보로부터 정상패턴에 대한 프로파일을 구축할 수 있어, 실시간 침입탐지에 필요한 부하를 줄일 수 있다. NIDES[4] 등의 연구에서 사용자 행동의 통계정보를 적용하였지만, 사용자 행동에 대한 특성값들의 분포가 클 경우 침입탐지에 활용하지 못한 단점이 있다.

호스트 컴퓨터 또는 네트웍 접속에서는 TELNET, FTP, WWW, SMTP 등을 통해 사용자 세션이 생성된다. 각 세션에서의 일련의 사용자 행동을 트랜잭션으로 정의하며, 이는 정상패턴에서의 기본 단위로 간주할 수 있다.

클러스터링은 연속된 수치의 특성값으로 구성된 데이터 집합에서 정상패턴의 특성값들을 추출하기에 효율적인 방법이다. 하지만, 기존의 클러스터링 기법들[1,2,3,4,5]은 트랜잭션 형태의 정보를 고려하지 않았다. [2]의 연구에서는 트랜잭션에 대한 클러스터링 방법을 주제로 하고 있지만, 대부분 클러스터링 기법과 같이 대상이 되는 데이터 집합은 한정되어 있어, 정적인 지식만을 찾을 수 있어, 데이터 스트림에서의 동적인 지식에 대한 수행은 어렵다.

최근, 데이터 스트림에서의 데이터 마이깅 연구[3,8,18]들이 제시되었다. 데이터 스트림은 지속적으로 무한히 생성되는 객체들의 집합으로 순서에 따라 한번의 검색만 가능하다. 그 결과 기존의 방법들과 같이 메모리 공간에 데이터 객체를 저장하고 처리하는 방법은 불가능하며, 무한한 데이터 객체에 비해 이를 처리하기 위한 메모리 공간은 한정된다. 이러한 제약으로 데이터 스트림은 일부 오차를 허용하는 대신에 그 성능을 제약에 맞게 변화시킨다.

클러스터링은 주어진 유사도함수에 따라 유사한 데이터 객체들의 그룹을 찾는 데이터 마이닝 방법이다. 대부분 클러스터링 연구들[6]은 고정된 데이터 집합을 대상으로 메모리 사용량과 처리 시간을 최소화하는데 목적을 두었다. 데이터 스트림 클러스터링 기법으로는 [7]의 O(1)-approximate k-medoid 연구가 있다. 데이터 스트림을 한정된 크기의 부분집합으로 분할하여 각 부분집합에 대한 클러스터링을 수행하여 이를 유지하는 방법을 사용하였지만, 분할기반(partitioning-based) 클러스터링 기법으로서 클러스터의 영역을 명확히 찾지못하고, 부분집합에서의 클러스터 중심점들을 다시 반복해서 클러스터링을 수행하는 단점이 있다.

본 연구에서는 사용자 감시 로그를 대상으로 격자기반의 클러스터링 기법을 적용한 침입탐지 기법을 제시한다. 사용자의 패턴은 감시 로그 데이터 스트림의 트랜잭션 내의 객체로 표현이 된다. 사용자 행동은 새로운 특성값으로 생성되어 해당 범위의 격자셀에서 그 통계적 정보를 관리한다. 격자셀의 트랜잭션에 대한 지지도가 높을 경우 해당 격자셀은 *μ-partition, σ-partition, hybrid-partition*에 의해 보다 정밀한 격자셀로 분할하게 되며 이를 반복하여 정밀한 격자셀들로 구성되는 클러스터를 찾을 수 있다.[8] 이러한 클러스터는 사용자 행동에 대한 정상패턴으로 분류되어 지속적으로 클러스터들을 갱신하면서 새로운 사용자 행동에 대한 침입 여부를 탐지할 수 있다.

본 연구의 구성은 다음과 같다. 2장에서 관련 연구들을 살펴본 후, 3장에서 감시 로그 데이터 스트림에 대한 클러스터링 방법을 제시한다. 4장에서 클러스터에 의한 사용자 패턴의 프로파일을 구성하고 이를 기반으로 침입탐지 방법을 제시한다. 5장에서는 실험을 통해 제시된 방법을 검증하고 6장에서 결론을 짓는다.

2. 트랜잭션 클러스터링

트랜잭션은 사용자에 의해 수행된 행동에 대한 일련의 특성값들을 포함한다. 하지만, 다수의 특성값 중 일부의 특성값만이 침입탐지에 중요한 영향을 미치는 만큼 트랜잭션 내의 특성값들에 대해

각각을 독립적으로 구분하여 클러스터링을 수행한다. i번째 생성된 트랜잭션 T^i는 $T^i=\{o_1^i, o_2^i,\cdots,o_k^i\}$의 i번째 생성된 객체들의 집합으로 정의되며, 데이터 스트림 $D^t=\{T^1,T^2,\cdots,T^t\}$으로 정의된다. 이 때, 데이터 스트림 내의 전체 트랜잭션 수는 $|D^t|$로 표시한다.

NIDES[4]와 같이 오래된 트랜잭션의 비중을 줄이기 위해 감쇄율[8]을 사용한다. 감쇄율은 과거의 정보의 비중을 지우는 비율을 의미하며, 이러한 방법에 의해 데이터 스트림의 최근 정보만을 유지한다.

먼저, 각 차원축 N_i를 주어진 p 개의 동일한 크기의 범위 $I_i^j = [s_i^j, f_i^j)$ $1\le j\le p$, 로 나눈다. s_i^j와 f_i^j는 i 차원상의 j 번째 범위의 시작좌표와 끝좌표를 나타낸다. 즉, 각 초기 격자셀은 d 개의 범위가 $\{I_1,I_2,\cdots,I_d\}$ $I_i\subseteq N_i$ $1\le i\le d$ 로 표시되며, 초기 격자셀 g 의 범위 $R(g)$는 각 범위가 교차하는 육방형 공간 $rs=I_1\times\cdots\times I_d$ 로 정의된다. 결과적으로 다차원 데이터 공간 N 은 p^d 개의 초기 격자셀로 구성된다. 초기 격자셀의 범위는 이 후 분할과정을 수행하여 분할된 공간 수 q 개의 육방형 공간의 집합 $RS=\{rs_1,rs_2,\cdots,rs_q\}$으로 정의된다. 이 때, 격자셀 g 의 i 번째 차원축에 대한 범위는 $IS_i(g)=\{I_i^1,I_i^2,\cdots,I_i^q\}$로 나타낸다. 즉, i 번째 차원에 대한 격자셀 g 의 크기 $|IS_i(g)|$는 이들 범위의 합으로 정의하며, 다차원 공간에서의 격자셀 g 의 범위는 육방형 공간 $rs_1,\cdots,rs_q$ 의 합, $R(g)=\bigtimes_{i=1}^{q} rs_i$ 로 나타낸다. 각 격자셀은 해당 공간 내의 데이터 객체들의 분포에 대한 통계정보를 저장하며, 정의 1 과 같이 정의할 수 있다.

정의 1. 격자셀 $g(RS,c,\mu,\sigma)$

최근 d 차원 데이터 스트림 D^t 에 대해, $g(RS, c^t, \mu^t, \sigma^t)$는 격자셀 g 의 공간 RS 내의 데이터 객체들의 분포에 대한 통계정보를 나타낸다. 격자셀 g 내의 데이터 객체들의 집합 $D_g^t=\{ e \mid e\in D^t$ and $e\in R(g)\}$에 대해, 격자셀 g 의 통계정보는 다음과 같이 정의된다.

i) c^t : D_g^t내의 감쇄율을 적용한 데이터 객체의 수. $c^t=\sum_{j=1}^{c^t}\tau^{(t-j)}$

ii) $\mu^t=<\mu_1^t,\cdots,\mu_d^t>$: D_g^t 내의 데이터 객체들의 i번째 차원에서의 평균값 μ_i^t

$$\mu_i^t=\sum_{j=1}^{c^t} e_i^j\times\tau^{(t-j)}/c^t,\ 1\le i\le d$$

iii) $\sigma^t=<\sigma_1^t,\cdots,\sigma_d^t>$: D_g^t 내의 데이터 객체들의 i번째 차원에서의 표준편차 σ_i^t

$$\sigma_i^t=\sqrt{\sum_{j=i}^{c^t}(e_i^j\times\tau^{(t-j)}-\mu_i^t)^2/c^t},\ 1\le i\le d\ \square$$

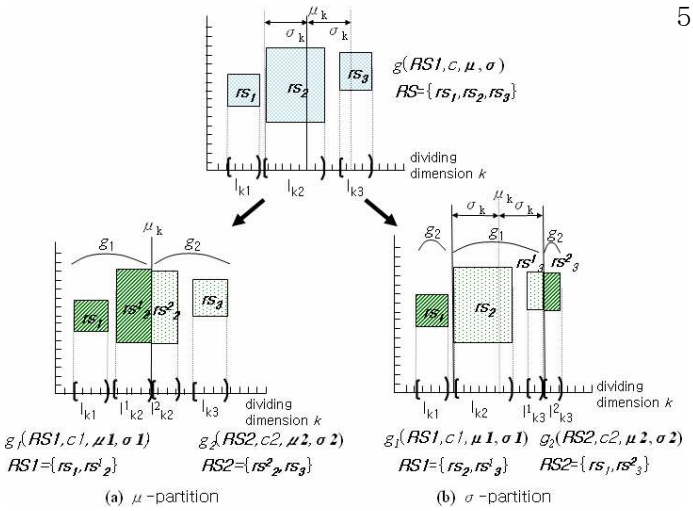
최근 데이터 스트림 D^t 에 생성된 데이터 객체 e^t 에 대해서 p^d 초기 격자셀 중 이를 포함하는 초기 격자셀 g 를 검색한다. 이전 $v(v\le t)$번째 생성된 데이터 객체에 의해 갱신된 격자셀 $g(RS, c^v,\mu^v,\sigma^v)$는 e^t 에 의해 감쇄율 τ 에 대해 다음과 같이 $g(RS, c^t,\mu^t,\sigma^t)$로 갱신된다.

$$c^t=c^v\times\tau^{(t-v)}+1,$$

$$\mu_i^t=\frac{\mu_i^v\times c^v\times\tau^{(t-v)}+e_i^t}{c^t},\qquad \text{for}\ \forall i,\ 1\le i\le d$$

$$\sigma_i^t=\sqrt{\frac{c^v}{c^t}\times\tau^{(t-v)}\times(\sigma_i^v)^2+\frac{(\mu_i^v)^2+(e_i^t)^2}{c^t}-(\mu_i^t)^2}$$

데이터 스트림 D^t 에 대해서, 격자셀 $g(RS, c^t, \mu^t, \sigma^t)$의 지지도는 D^t 내의 데이터 객체의 수에 대한 격자셀 내의 데이터 객체의 수의 비율, $c^t/|D^t|$로 정의된다. 격자셀의 지지도가 주어진 분할지지도 $S_{splt}(S_{splt}<S_{min})$ 이상인 경우, 해당 공간 내에 클러스터가 존재하는 것으로 간주하고 보다 정확한 범위를 찾기 위해 해당 격자셀을 그림 1 과 같이 두 개의 중간 격자셀 g_1 과 g_2 를 해당 초기 격자셀의 하위로 생성되며, 이 때 생성되는 중간 격자셀들의 범위와 통계정보는 사용된 분할방법에 따라 결정된다. [8]



(그림 1) 격자셀 분할방법

클러스터의 특성을 축약하기 위해서 정의 2와 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.

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

cdev(C^k)는 center(C^k)에 대한 표준 편차를 나타낸다.

5) **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.

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

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

3. 비정상행위탐지

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 표준편차인 *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.

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

$$= 1 - \sum_{j=1}^{m} \text{ratio}(C_j^k) \cdot \text{support}(C_j^k)$$

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. [9]

4. 실험

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)[9]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 1과 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.

그림 2는 제안된 알고리즘을 이용하여 최소 지지도가 10%~90% 일 때 DATA1으로부터 생성된 클러스터의 개수를 나타낸다. 이 실험에서 최소 지지도와 클러스터링 범위가 증가함에 따라 클러스터의 개수가 작아짐을 알 수 있다. 그림 3에서는 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.

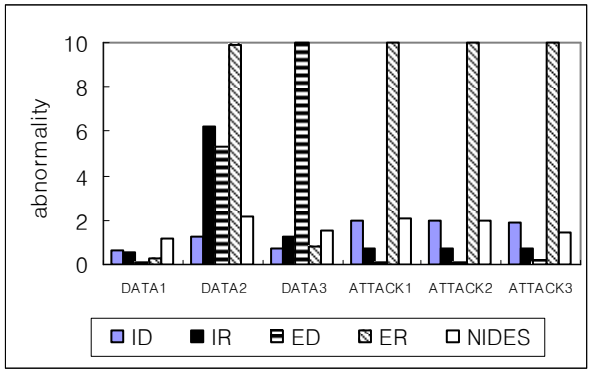


그림 2. 제안된 방법의 성능

5. 결론

본 논문에서는 감사 데이터 스트림에서 공통 지식을 찾는 데이터스트림 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

[참고문헌]

[1] K. Illgun, R. Kemmerer, Phillip A. Porras, "*State Transition Analysis: A rule-based intrusion detection approach*," IEEE Transaction on Software Engineering pp 181-199, March. 1995

[2] Vasilios Katos, Network intrusion detection: Evaluating cluster, discriminant, and logit analysis, *Information Sciences, Volume 177, Issue 15*, Pages 3060-3073, 1 August 2007

[3] 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

[4] 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.

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

[6] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu: "*A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise*," Proc. 2nd int. Conf. on Knowledge Discovery and Data Mining (KDD '96), Portland, Oregon, 1996, AAAI Press, 1996

[7] Charu C. Aggarwal, Jiawei Han, Jianyong Wang, and Philip S. Yu, "*A Framework for Clustering Evolving Data Streams*," In *Proc. VLDB 29th*, Berlin, 2003.

[8] Nam Hun Park and Won Suk Lee, "A statistical Grid-based Clustering over data streams," ACM SIGMOD Record, Volume 33, Issue 1, Page 32-37, 2004.

[9] Sang Hyun Oh and Won Suk Lee, "An Anomaly Intrusion Detection Method by Clustering Normal User Behavior," Computers and Security, Vol 22 No 7, Pages 596-612, 2003