데이터 스트림 클러스터링을 이용한 침임탐지

Intrusion Detection based on Clustering a Data Stream

오상현, 강진석*, 변영철**
자바정보기술, 군산대학교*, 제주대학교**

Oh Sang-Hyun, Kang Jin-Suk*, Byun Yung-Cheol**
Java Information Tech., Kunsan National Univ.*, Cheju National Univ**

요약

비정상행위 탐지를 위해서는 사용자의 정상적인 행위 모델링이 중요한 이슈가 된다. 이러한 정상적인 행위를 간략한 프로파일로 생성하기 위해서 기존의 데이터 마이닝 기법들은 주로 고정된 데이터 집합을 이용하였다. 하지만 이러한 접근 방법들은 단순히 사용자 행위의 정적인 면만을 모델링 할 수 있다. 이러한 단점을 극복하기 위해서 사용자의 행위를 연속된 데이터 스트림으로 처리해야 한다. 본 논문에서는 데이터 스트림을 모델링하는 새로운 클러스터링 방법을 제안한다. 이를 위해서, 사용자의 행위의 특성을 표현하는 다양한 특징들로 분류한다. 따라서 각 특징에 대해, 제안된 클러스터링 알고리즘을 이용하여 지금까지 관찰된 특징 값들을 기반으로 클러스터 탐색하게 된다. 결과적으로 사용자의 과거 행위들을 유지할 필요 없이 사용자의 새로운 행위를 클러스터링 결과에 연속적으로 반영될 수 있다.

Abstract

In anomaly intrusion detection, how to model the normal behavior of activities performed by a user is an important issue. To extract the normal behavior as a profile, conventional data mining techniques are widely applied to a finite audit data set. However, these approaches can only model the static behavior of a user in the audit data set. This drawback can be overcome by viewing the continuous activities of a user as an audit data stream. This paper proposes a new clustering algorithm which continuously models a data stream. A set of features is used to represent the characteristics of an activity. For each feature, the clusters of feature values corresponding to activities observed so far in an audit data stream are identified by the proposed clustering algorithm for data streams. As a result, without maintaining any historical activity of a user physically, new activities of the user can be continuously reflected to the on-going result of clustering.

1. 서 론

컴퓨터와 통신 기술의 발달로 컴퓨터 시스템과 관련된 예기치 않은 침입 및 범죄에 의한 피해가 급증하고 있다. 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 모델[1, 2, 3]이 연구되고 있다. 비정상행위 탐지 모델에서는 사용자의 과거 행위들에 대한 프로파일을 생성하고 새로운 사용자의 행위가 발생했을 경우 이 프로파일과의 차이가 클 때 침입으로 간주된다.

본 논문에서는 데이터스트림 클러스터링 [4, 5, 6]

을 이용한 새로운 침입탐지 방법을 제안한다. 기존의 데이터 스트림 클러스터링 방법에서는 주어진 개수의 클러스터만을 생성하였다. 하지만 데이터 스트림 내에서 클러스터의 개수가 알려져 있지 않기 때문에 클러스터의 정확도가 떨어지게 된다. 이와 달리, 본 논문에서 제안하는 방법에서는 새로운 데이터가 입력될 때 마다, 하나의 클러스터가 두 개의 클러스터로 분할되거나 인접한 두 클러스터가 하나의 클러스터로 병합될 수 있다. 따라서 보다 효과적으로 클러스터를 생성할 수 있다. 제안된 방법에서는 생성된 클러스터에 대한 통계 수치를 이용하여 프로파일을 구성하고 이를 이용하여 침입 탐지의 성능을 높인다.

II. 스트림 데이터 클러스터링

데이터 스트림 S는 객체 집합 $\{o_1, o_2, \cdots, o_h\}$와 같이 표현되며 객체 o_i 는 n차원 벡터 $o_i = (o^1_i, o^2_i, \cdots, o^n_i)$로 구성된다. 이때, S의 k번째 차원에 투영된 스트림을 $S^k = \{o^k_1, o^k_2, \cdots, o^k_h\}$와 같이 표현한다. 또한, X^k를 데이터 스트림 S^k에서 생성된 클러스터들의 집합이라 정의한다. 생성되는 클러스터의 정확도를 높이기 위해서, 각 클러스터는 격자-셀(grid-cell) 집합을 포함한다. 하나의 격자-셀은 다른 격자-셀과 겹치지 않고 동일한 크기를 가지며 유일한 식별자를 가진다. 예를 들어, 객체 ok 의 식별자는 $I^k = \lceil o^k/\rho^k \rceil$ 와 같이 계산된다. 이때 ρ^k는 k번째 차원에서 격자-셀의 크기를 나타낸다. 격자-셀 $g^k = (I^k, gl^k, gs^k, gn^k)$는 다음과 같은 정보를 포함한다: 식별자 I^k, 격자-셀에 포함되는 객체들의 합 gl^k, 객체들의 제곱합 gs^k 및 객체들의 개수 gn^k. 하나의 클러스터에 포함되는 속성들은 다음과 같다.

δ^k: 클러스터 C^k에 포함된 전체 객체의 수
μ^k: 클러스터에 C^k에 포함된 전체 객체의 평균
SS^k: 클러스터 C^k에 포함된 각 객체 제곱의 합
σ^k: 클러스터에 C^k에 포함된 전체 객체들의 표준 편차
$gSet^k$: 클러스터에 C^k의 그리드셀(grid-cell) 집합

데이터 스트림 S^k로부터 새로운 객체 o^k가 발생했을 때 클러스터에 포함시키기 위해서 이 객체와 가장 가까운 클러스터를 X^k로부터 선택하게 된다. 선택된 클러스터를 갱신하는 방법은 다음과 같다.

$$C^k\left(\delta^k+1, \frac{\mu^k\cdot\delta^k+o^k}{\delta^k+1}, SS^k+(o^k)^2, gSet^k\right)$$

새로운 객체가 발생됐을 때 이 객체를 포함하는 격자-셀만을 갱신하게 된다. 즉, 격자-셀 g^k의 식별자가 객체 o^k의 식별자와 같을 때 g^k의 속성들이 갱신된다. $\overline{gl^k}$, $\overline{gs^k}$ 및 $\overline{gn^k}$ 를 각각 g^k의 현재 속성들이라 하고 gl^k, gs^k 및 gn^k 을 새로 갱신된 속성들이라 하자. 그러면, 유닛 식별자가 $\lceil o^k/\rho^k \rceil$ 인 g^k의 새로 갱신되는 속성들은 다음과 같이 계산된다.

$$g^k = \left(gl^k+o^k, gs^k+(o^k)^2, gn^k+1\right)$$

인접한 두 클러스터 사이에 많은 개수의 객체가 발생될 경우 두 클러스터는 서로 가까워지게 된다. 만일 두 클러스터에 포함된 객체들의 표준 편차가 사용자 정의 임계치인 최소 편차 *minimum_deviation* 이하면 두 클러스터는 하나의 클러스터로 병합된다. 두 클러스터 X^k에 포함된 C^k_1 및 C^k_2 에 대해서, 두 클러스터에 포함된 객체들의 표준 편차는 다음과 같이 계산된다.

$$\sigma = \sqrt{\frac{SS^k_1+SS^k_2}{\delta^k_1+\delta^k_2} - \left(\frac{\mu^k_1\cdot\delta^k_1+\mu^k_2\cdot\delta^k_2}{\delta^k_1+\delta^k_2}\right)^2}.$$

따라서, $\sigma \le$ *minimum_deviation* 일 때 두 클러스

터는 병합되며 병합된 클러스터의 속성들은 다음과 같이 계산된다.

$$C^k\left(\delta_1^k+\delta_2^k,\ \frac{\mu_1^k\cdot\delta_1^k+\mu_2^k\cdot\delta_2^k}{\delta_1^k+\delta_2^k},\ SS_1^k+SS_2^k,\ gSet_1^k\cup gSet_2^k\right)$$

위 수식에서, 클러스터 C^k의 중심 값은 두 클러스터 C^k_1와 C^k_2의 가중치 평균으로 구할 수 있다. 또한, 클러스터 C^k의 제곱합은 SS^k_1 와 SS^k_2의 합으로 구할 수 잇다. 클러스터 C^k_1과 C^k_2가 클러스터 C^k로 병합될 때, 클러스터 C^k의 격자셀 집합 $gSet^k$는 $gSet^k_1$와 $gSet^k_2$의 합집합으로 구성된다. 이는 격자 셀의 정의에 의해서 두 클러스터들의 격자 셀들이 겹치지 않기 때문에 가능하다.

한편, 데이터 스트림으로부터 발생되는 객체들의 수가 많아 질 경우, 하나의 클러스터에 포함된 객체들의 표준 편차가 커지게 되서 클러스터의 정확도가 떨어지게 된다. 따라서 클러스터의 정확도를 유지하기 위해서 최소 편차를 기반으로 클러스터를 두 개의 클러스터로 분할하게 된다. X^k에 포함된 클러스터 C^k에 대해서 $gSet^k$를 $\{g^k_1, g^k_2, ..., g^k_p, ..., g^k_q\}$라 하자. 만일 $\delta^k > minimum_deviation$이면 클러스터 C^k는 두 클러스터 C^k_1와 C^k_2로 분할된다. T^k_i를 g^k_i 에 포함된 객체 집합이라 하고 r^k_p 를 $\cup^p_{i=1}T^k_i$ 에 포함된 전체 객체 수라 하면, $r^k_{p-1} < \delta^k/2 \le r^k_p < \delta^k$ 일 때 각 클러스터의 격자집합은 $gSet^k_1 = \{g^k_1, g^k_2, ..., g^k_p\}$ 및 $gSet^k_1 = \{g^k_{p+1}, ..., g^k_q\}$ 와 같다. 결과적으로, C^k_1 및 C^k_2 의 속성들은 다음과 같이 얻을 수 있다.

$$C^k_1\left(\sum_{i=1}^{p} gn^k_i,\ \frac{1}{r^k_p}\sum_{i=1}^{p} gl^k_i,\ \sum_{i=1}^{p} gs^k_i,\ \bigcup_{i=1}^{p} g^k_i\right)$$

$$C^k_2\left(\sum_{i=p+1}^{q} gn^k_i,\ \frac{1}{\delta^i - r^k_p}\sum_{i=p+1}^{q} gl^k_i,\ \sum_{i=p+1}^{q} gs^k_i,\ \bigcup_{i=p+1}^{q} g^k_i\right)$$

각 특징에 대해서, 클러스터들은 클러스터 중심과 특징 값들의 표준편차를 포함하는 프로파일로 요약된다. 따라서 새로 발생된 행위는 특징별로 프로파일과 비교하여 비정상행위인지를 판별하게 된다. 새로 발생된 행위의 특징 값 o^k와 프로파일에서 이 특징 값과 가장 가까운 클러스터와의 차 $diff(X^k, o^k)$는 다음과 같이 정의된다.

$$diff(X^k, o^k) = \frac{|\mu^k - o^k|}{\sigma^k} \quad (C^k \in X^k)$$

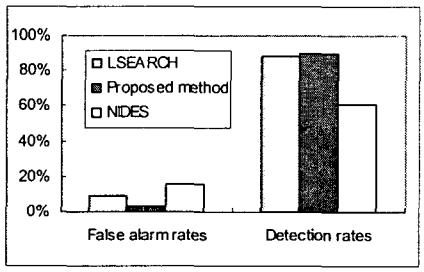
위의 수식에서 클러스터 C^k와 특징값 o^k와의 차이는 클러스터의 표준 편차로 나누어진다. 그 이유는 다양한 특징들 간을 표준화 시켜주기 위함이다. 따라서 비정상행위 탐지에 참여하는 특징들의 개수를 n이라 했을 때, 비정상행위도는 다음과 같이 구할 수 있다.

$$abnormality(o) = \frac{1}{n}\cdot\sum_{i=1}^{n} diff(X^i, o^k)$$

III. 실험 결과

본 논문에서 제안된 방법의 성능을 실지 환경에서 평가하기 위해서 1998년 DARPA에서 수집한 로그 데이터 [7]을 이용하였다. 클러스터링을 위한 특징 값들은 BSM (Basic Security Module) [8]을 이용하여 로그 데이터로부터 추출하였다. 특징 값들을 추출하기 위해서 Solaris 2.6의 BSM을 이용하였다. 추출되는 특징들은 유닉스 명령어가 실행될 때 발생되는 시스템 콜들 중에서 84개를 주요 특징들로 이용하였다. 여다. 실험을 위해서 두 종류의 로그 데이터, 즉 프로그래머가 사용한 로그 기록과 시스템 관리자가 사용한 로그 기록을 사용하였다. 프로그래머의 로그 기록에는 에는 C 코드를 작성하기 위한 vi 에디터, 컴파일러, 이메일 및 각종 기본 유닉스 명령어들이 포함된다. 시스템 관리자가 사용한 로그 기록에는 권한 부여된 명령어들이 포함된다.

[그림 1]에서는 제안된 방법과 스트림 데이터 클러스터링 기법인 LSEARCH 및 통계적 침임 탐지 시스템인 NIDES간의 false alarm rate 및 탐지율의 비교결과를 보여준다. 이 실험에서 LSEARCH와 NIDES의 false alarm rates 가 제안된 방법에 비해서 높게 나타났다. 한편, NIDES의 탐지 결과는 제안된 방법과 LSEARCH에 비해서 상대적으로 낮게 나타났다. 결과적으로 제안된 방법의 성능이 기존의 LSEARCH 및 NIDES보다 효율적임을 알 수 있다.



▶▶ 그림 1. 침입 탐지 결과

IV. 결 론

본 논문에서는 데이터 스트림 클러스터링을 이용한 침입 탐지 방법을 제안하였다. 제안된 방법에서는 각 특징에 대해서 데이터 저장 없이 클러스터를 효과적으로 탐색한다. 이를 위해서, 분할/병합을 이용하여 클러스터들을 생성한다. 결과적으로, 제안된 방법이 기존의 클러스터링 방법보다 정확한 클러스터를 탐색하게 된다. 침입탐지를 위해서 새로 생성된 데이터가 클러스터 및 프로파일에 계속적으로 반영된다. 따라서 부가적인 처리 없이 침입 탐지를 수행할 수 있다.

▌참 고 문 헌▐

[1] 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.

[2] 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.

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

[4] Guha, S., Meyerson, A., Mishra, N., Motwani, R., and O'Callaghan, L. "Clustering data streams: Theory and practice," IEEE Trans. Knowl. Data Eng 15, 3(2003), 515--528.

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

[6] Tian Zhang, Raghu Ramakrishnan, and Miron Livny, "Birch: An Efficient data clustering method for very large databases," Proceedings for the ACM SIGMOD Conference on Management of Data, Montreal, Canada, June 1996.

[7] http://www.ll.mit.edu/IST/ideval/index.html

[8] Sun Microsystems. SunShield Basic Security Module Guide.