사이트의 접속 정보 유출이 없는 네트워크 트래픽 데이터에 대한 순차 패턴 마이닝

김승우 박상현 원정임
연세대학교 컴퓨터과학과
e-mail:{kimsw, sanghyun, jiwon}@cs.yonsei.ac.kr

Privacy Preserving Data Mining of Sequential Patterns for Network Traffic Data

Seungwoo Kim, Sanghyun Park, JungIm Won
Department of Computer Science, Yonsei University

요 약

본 논문에서는 대용량 네트워크 트래픽 데이터를 대상으로 사이트의 프라이버시를 보호하면서 마이닝 결과의 정확성, 실용성 등을 보장할 수 있는 효율적인 순차 패턴 마이닝 기법을 제안한다. 네트워크가 발달함에 따라 네트워크 트래픽 데이터에 대한 마이닝은 네트워크를 통한 통신의 패턴을 찾아내고, 이를 사용하여 침입 탐지, 인터넷 웜의 탐지 등으로 유용하게 쓰이게 되었다. 그러나 네트워크 트래픽 데이터는 네트워크 사용자 개개인의 인터넷 접속 형태, IP 주소 등의 정보를 포함하는 데이터로 네트워크 사용자의 프라이버시를 해칠 수 있다는 문제점이 존재한다. 따라서 이들 네트워크 트래픽 데이터를 대상으로 하는 마이닝 기법에서는 프라이버시 보호를 위하여 각 사이트에 저장되어 있는 네트워크 트래픽 데이터를 공개하지 않으면서도, 의미있는 패턴을 찾을 수 있어야 한다. 본 논문에서는 프라이버시 보호를 위하여 N-저장소 서버 모델을 제안한다. 제안된 모델에서는 데이터를 분할하여 암호화한 후, 이를 복호화할 수 없는 서버에서 집계하는 방식을 사용하여 실제 데이터가 저장되어 있는 각 사이트의 출처 정보를 감추는 방식을 사용한다. 또한, 효율적인 빈번 패턴 생성을 위하여 빈번 항목에 대한 인덱스 구조를 제안하고, 이를 기반으로 한 순차 패턴 마이닝 기법을 보인다.

1. 서론

인터넷이 보급되기 시작한 이후 네트워크 사용자의 급속한 증가에 따라 네트워크에 연결된 컴퓨터의 수와 네트워크를 통해 전송되는 데이터의 양이 점점 증가하고 있다. 최근 이들 네트워크 상에서 발생하는 대용량 네트워크 트래픽 데이터를 자동화된 원격 정보 수집을 통하여 서버에 전송하여 저장, 관리하고, 수집된 데이터를 분석함으로써 유용한 정보를 추출하려는 연구가 활발히 진행되고 있다[1,2,3]. 그 예로 군사적, 상업적 용도 등의 다양한 응용 분야에서 서버로 전송되는 비정상적인 데이터의 흐름을 파악함으로써 외부로부터의 침입이나 인터넷 웜의 동작을 탐지하는 연구를 들 수 있다.

timestamp	source address	source port	destination address	destination port
13:37:11.950966	kimsw	36872	yonsei.ac.kr	www
13:37:11.954474	yonsei.ac.kr	www	kimsw	36872
13:37:22.384472	kimsw	36915	192.168.1.3	telnet
13:37:22.385327	192.168.1.3	telnet	kimsw	36915

<표 1> tcpdump를 사용하여 얻은 네트워크 트래픽 데이터의 예

<표 1>은 tcpdump를 사용하여 얻은 네트워크 트래픽 데이터의 예를 보인다. 각 엔트리는 트래픽이 발생한 시간과 송신 및 수신지에 대한 주소, 포트 번호 등의 정보로 구성된다. 이들 네트워크 트래픽 데이터는 일반적인 데이터와 비교하였을 때 다음과 같은 특성을 지닌다. 첫째, 모든 연결 가능한 네트워크 정보가 트래픽의 대상이 되므로 서로 다른 값을 갖는 항목이 매우 많아진다. 즉, 매우 많은 항목들로 구성된다. 둘째, 네트워크 상에서의 매우 빈번한 송수신으로 인하여 새로운 데이터가 지속적으로 생성되는 대용량 데이터가 된다. 셋째, 네트워크 상에 연결된 다수의 사이트에서 발생되는 데이터로, 데이터가 분산되어 저장되며, 각 사이트는 각 개인의 네트워크 사용에 대한 트래픽 정보를 저장하게 된다.

패턴1:	192.168.1.254로부터의 데이터 수신 -> 192.168.1.254로의 데이터 송신 -> 192.168.1.254로의 데이터 송신 -> 192.168.1.254로의 데이터 송신
패턴2:	yonsei.ac.kr로부터의 데이터 수신 -> yscec.yonsei.ac.kr으로의 데이터 송신

<표 2> 네트워크 트래픽 데이터에서 발견할 수 있는 패턴의 예

이러한 특징을 갖는 대용량 네트워크 트래픽 데이터의 분석을 위해서는 클러스터링이나 연관 규칙 발견과 같은 다양한 데이터 마이닝 기법을 사용할 수 있다. 하지만, 네트워크 상에서 발생한 항목들 간의 의미 있는 시간적 선후 관계를 발견하기 위해서는 순차 패턴 마이닝 기법을 활용해야 한다[1,2]. 네트워크 트래픽 데이터에 대한 순차 패턴의 예를 <표 2>에 보인다. 예를 들어 마이닝 결과로 얻은 패턴 2는 다수의 사이트에서 “yonsei.ac.kr”로부터의 데이터 수신이 발생한 후에는 “yscec.yonsei.ac.kr”로의 데이터 송신이 빈번하게 발생하였음을 의미한다.

그러나 네트워크를 통해 전송되는 데이터는 컴퓨터를 사용하는 개개인들의 인터넷 접속 형태, IP 주소, 접속 포트 등의 개인 프라이버시에 직결되는 정보를 포함하므로, 데이터의 수집 과정에서 데이터의 출처를 은닉하거나 데이터의 변형 등을 통하여 개개인의 프라이버시를 보호하는 등의 추가적 기술이 요구된다. 또한, 은닉 혹은 변형된 형태로 전송되어 수집, 저장된 데이터를 대상으로 마이닝 결과의 정확성을 보장할 수 있는 신뢰성 있는 마이닝 기법의 개발이 수반되어야 한다.

특히 네트워크 트래픽 데이터의 세가지 특성을 고려하였을 때, 마이닝 기법이 가져야 할 조건은 다음과 같다. 첫 번째로 많은 종류의 항목을 가지는 데이터를 처리할 수 있어야 한다. 두 번째로 다수의 사이트에 존재하는 데이터에 대해서 마이닝을 할 수 있어야 한다. 세 번째로 많은 양의 데이터를 대상으로 하기 때문에 효율적으로 마이닝이 이루어져야 한다.

프라이버시를 보장하면서도 마이닝할 수 있는 방법에 대해서 최근 많은 연구가 진행되고 있으나, 이들 대부분은 적은 종류의 항목을 가지는 데이터를 대상으로 하거나 소수의 사이트만을 대상으로 마이닝을 수행하는 방식이다. 따라서 네트워크 트래픽 데이터에 기존의 방식을 그대로 적용할 경우, 마이닝 결과의 부정확성 및 비실용성 등의 문제점을 초래할 수 있다.

본 논문에서는 기존 방법이 가지는 문제점을 해결하며, 위의 세 가지 조건을 만족시키는 1) 대용량 네트워크 트래픽 데이터를 대상으로 사이트의 프라이버시를 보호하면서 마이닝 결과의 정확성, 실용성 등을 보장할 수 있는 효율적인 순차 패턴 마이닝 기법을 제안한다. 제안하는 기법에서는 우선 하나의 마이닝 서버와 같이 작동할 수 있는 2) N-저장소(Repository) 서버 모델을 제안하여 네트워크 상의 각 사이트에서 발생한 빈번 항목들을 저장, 관리한다. N-저장소 서버 모델은 각 사이트가 가지는 데이터를 분할하여 각기 다른 방식으로 암호화한 후, 출처를 감추기 위하여 각

데이터를 복호화를 수행할 수 없는 N개의 서버에 전송하여 집계를 수행한다. 다음, 집계된 정보를 가지고, 복호화를 수행할 수 있는 N개의 서버로 데이터를 재전송하여 빈번 패턴을 생성한다. 이 과정에서 후보 패턴의 실제 발생 여부를 각 사이트에 질의하도록 하고, 그 결과를 이분화된 값으로 제한하면서 기존의 정보 유지 대체 기법(Retention replacement)[4]을 적용하여 마이닝 시에 발생할 수 있는 프라이버시 보장 문제를 해결한다. 또한 3) 빈번 패턴을 효율적으로 생성하기 위하여 각 사이트에서 발생한 길이 1인 빈번 항목에 대한 인덱스 구조를 제안한다. 또한 각 사이트에 존재하지 않는 패턴들에 대한 리스트를 캐쉬에 유지함으로써 패턴의 존재 여부를 보다 빠르고, 신속하게 판단할 수 있도록 한다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구와 관련된 몇 가지 기존 연구를 살펴본다. 3장에서는 본 논문에서 해결하고자 하는 문제를 정의하고 제안하는 기법에 대하여 설명한다. 마지막으로 4장에서는 결론을 내린다.

2. 관련 연구

네트워크 트래픽 데이터를 마이닝 기법을 이용하여 분석함으로써 의미 있는 결과를 얻고자하는 많은 연구들이 진행되고 있다[1,2,3].

Lee 등은 침입 당한 상태와 정상 상태의 네트워크의 데이터를 순차 패턴 마이닝 기법을 이용하여 분석함으로써 침입 시에만 발생하는 패턴을 구하고, 이를 기반으로 침입 탐지 모델을 제안하여 네트워크 트래픽 데이터에 대한 마이닝 결과의 유용성을 보였다[1]. 침입 탐지를 위해 네트워크 트래픽 데이터를 마이닝하는 연구는 이후에도 지속적으로 이루어져왔다[2,3]. 그러나 이러한 연구들은 네트워크 상에서 발생한 모든 데이터를 한 곳에 모아서 저장한 후, 마이닝을 수행하기 때문에 수집된 개개인의 네트워크 정보가 마이닝 시에 노출되어 개인의 프라이버시를 침해할 수 있다는 문제점을 갖는다.

Clifton 등은 마이닝 과정에서 프라이버시의 침해가 발생할 수 있다는 문제를 지적하였으며[5], 이후 이러한 문제점을 해결하기 위하여 프라이버시를 보장하면서 마이닝을 수행하기 위한 많은 연구가 진행되고 있다[4,6,7,8,9,10]. 이들 연구는 크게 두 가지로 분류될 수 있다.

첫 번째는 데이터를 수집하는 과정에서 데이터를 변형하여 프라이버시를 침해하지 못하도록 제약하고, 변형 후의 데이터를 변형 전의 데이터 분포를 갖도록 재구성하여 마이닝을 수행하는 연구이다[4,6,7]. 이 분류에는 사이트는 수치 데이터에 대해서 확률 분포로부터 선택된 임의의 값을 더하여 서버에 전달하여 프라이버시를 지키며, 데이터를 받은 서버가 이 확률 분포를 사용하여 전체 데이터의 실제 분포를 구하여 의사 결정 트리를 마이닝하는 방법이 있다[6]. 정보 유지 대체 기법을 통한 데이터의 변형과 재구성은 0과 1로 양분될 수 있는 데이터 형태에 적용될 수 있는 기법으로, 데이터 수집 과정에서 사이트가 p의 확률로 원래의 데이터를, (1-p)의 확률로 변형된 데이터를 수집하는 측에 전달한다. 데이터를 수집한 측에서는 데이터 0과 1에 대한 발생 빈도를 집계한 후 집계된 빈도와 확률 p를 가지고 실제 0과 1의 분포 정도를 계산한다[4]. 그러나, 이 기법들은 데이터의 값이 특정한 종류로 수치 데이터나, 두 가지 값을 가지는 데이터로 제한된다는 한계점이 있다. 두 가지 방법을 다양한 데이터에 대해 적용하기 위하여 연구가 진행되었으나[7,11], 데이터가 가지는 값의 종류가 커질수록 결과의 정확성이 감소한다는 문제점이 남아있다.

두 번째는 여러 집단의 데이터를 마이닝하기 위하여 각각의 사이트가 마이닝 과정에 참여하여 프라이버시를 침해할 수 있는 데이터는 사이트가 직접 처리하고, 서버가 프라이버시를 침해하지 않는 중간 과정의 결과를 모아 최종 결과를 얻는 기법에 관한 연구이다[8,9,10]. 이중에서 Kantarcioglu 등의 연구는 여러 사이트에서 각 집단이 지니는 개개의 데이터에 대한 프라이버시를 보호하면서 연관 규칙을 찾는 기법[9]으로, 상호 암호화(Commutative encryption) 방식을 사용하여 데이터를 수집한 후, 시큐어 섬(Secure sum) 방식을 사용하여 각 사이트에서의 데이터 발생 빈도를 구함으로써 연관 규칙을 발견한다. 그러나, 이 방식에서 사용되는 두 가지의 연산은 전체 사이트들을 연결하는 사이클을 따라 순차적으로 데이터를 전송해야 하므로 사이트 수가 매우 큰 경우에는 비효율적이라는 한계점이 있다.

기존의 연구들을 대용량 네트워크 트래픽 데이터에 적용할 경우 다음과 같은 문제점이 존재한다. 1) 다양한 종류가 존재하는 네트워크 트래픽 데이터의 특성 때문에 적용하기가 어려우며, 데이터를 원래대로 복구하는 과정에서 정확성이 감소하여 원하는 마이닝 결과를 얻을 수 없다. 2) 네트워크 상에 매우 많은 수의 사이트가 존재하므로 소수의 사이트만을 대상으로 하는 마이닝 기법은 실용성면에서 한계를 갖는다.

3. 제안하는 기법

본 장에서는 본 논문에서 제안하는 대용량 네트워크 트래픽 데이터를 대상으로 개인의 프라이버시를 보호하면서 마이닝 결과의 정확성, 실용성 등을 보장할 수 있는 효율적인 순차 패턴 마이닝 기법에 대해서 설명한다.

3.1 네트워크 트래픽 데이터의 패턴

네트워크 트래픽 데이터는 tcpdump와 같은 tcp/ip 데이터 캡쳐 프로그램을 사용하여 저장된다. tcpdump로부터 얻을 수 있는 정보는 <표 1>에서 살펴본 것과 같이 해당 트래픽이 발생한 시점의 타임스탬프와, 데이터를 전송한 측과 수신한 측에 대한 주소(address) 및 포트(port) 번호를 포함한다. 본 논문에서는 이러한 트래픽 데이터를 대상으로 <표 2>와 같은 패턴을 찾고자 한다. 이를 위하여 우선 <표 1>의 트래픽 데이터를 <표 3>과 같이 재구성한다. 이는 kimsw사이트에서 송수신이 발생한 정보를 표현한다. 여기서, out은 데이터를 송신한 경우를 나타내며, in은 데이터를 수신한 경우를 나타낸다. 또한, 빈번 패턴이 생성될 가능성을 높이기 위하여 패턴을 이루는 데이터에 포트를 포함하는 대신 주소 정보만을 사용한다.

timestamp	in/out	address
13:37:11.950966	out	yonsei.ac.kr
13:37:11.954474	in	yonsei.ac.kr
13:37:22.384472	out	192.168.1.3
13:37:22.385327	in	192.168.1.3

<표 3> 재구성된 네트워크 트래픽 데이터의 예

이렇게 재구성된 네트워크 트래픽 데이터를 대상으로 <표 2>와 같은 의미있는 패턴을 찾기 위해서는 각 데이터 간의 전후 시간 관계를 고려한 순차 패턴 마이닝 기법을 활용하는 것이 유용하다. 그러나, 순차 패턴을 발견하기 위해서는 패턴을 구성할 수 있는 트래픽 데이터를 모두 스캔해야한다. 이 경우 패턴이 될 수 있는 네트워크 트래픽 간의 시간 간격을 무한하게 설정하면 스캔해야 하는 데이터의 양이 지나치게 커지므로 비효율적이다. 또한 트래픽 데이터 간의 시간 간격이 매우 큰 패턴은 신뢰성면에서 무의미할 수 있다. 따라서, 본 논문에서는 이러한 문제점들을 고려하여 보다 효율적이고, 의미있는 패턴의 생성을 위하여 패턴 내의 네트워크 트래픽들 간에 최대 시간 간격(max gap)을 의미를 가지는 적절한 값으로 정하여 사용한다.

3.2 순차 패턴 마이닝 기법의 구성

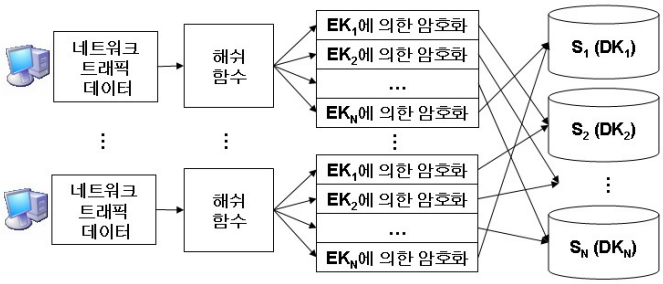
본 절에서는 제안하는 N-저장소(Repository) 서버 모델을 이용하여 길이 1의 빈번 항목을 효율적으로 생성하는 방법과 이를 조합하여 길이 n의 패턴을 생성하는 방법에 대하여 설명한다. N-저장소 서버 모델은 기존 방식이 취하던 사이트 간의 순환 전송으로 인한 비효율성을 해결하기 위하여, 사이트 간의 순환 전송 없이 각 사이트에 저장된 데이

터를 수집하여, 길이 1인 빈번 항목을 효율적으로 생성, 저장, 관리하기 위한 방법이다.

3.2.1 빈번 항목 생성을 위한 N-저장소 서버 모델

N-저장소 서버 모델은 N개의 서버 $\{S_1, S_2, ..., S_N\}$와 N개의 암호화 키와 복호화 키의 쌍 $\{(EK_1, DK_1), (EK_2, DK_2), ..., (EK_N, DK_N)\}$으로 구성된다. 각 사이트는 N개의 암호화 키를 모두 보관하고 있으며, 서버 S_i는 복호화 키 DK_i를 보관하고 있다. 빈번 패턴 생성을 위하여 N-저장소 서버 모델은 다음과 같이 작동한다.

1) 각 사이트 별로 네트워크 트래픽 데이터를 취합한다.
2) 각 사이트는 취합된 트래픽 데이터를 해쉬(hash) 함수를 사용하여 N개의 그룹 $(G_1, G_2, ..., G_N)$으로 분할한다.
3) 각 사이트는 1부터 N까지의 각 i에 대하여 G_i에 속한 각각의 트래픽 데이터를 암호화 키 EK_i를 사용하여 암호화한다.
4) 각 사이트는 1부터 (N-1)까지의 각 i에 대하여 G_i에 속한 각각의 트래픽 데이터를 서버 $S_{(i+1)}$에 전송한다. G_N에 속한 각각의 트래픽 데이터는 서버 S_1에 전송한다. 각각의 서버 S_i는 EK_i로 암호화된 트래픽 데이터만을 복호화 할 수 있으므로 각 사이트의 프라이버시는 보호된다.



<그림 1> 해쉬 함수에 따른 데이터의 분할 및 암호화 과정

5) 서버 S_i는 전송받은 암호화된 데이터로부터 같은 값을 가지는 데이터의 발생 회수를 집계하여 빈번 항목을 생성한다.
6) 생성된 암호화된 빈번 항목을 원래 데이터로 복호화할 수 있도록 서버로 재전송한다. 즉, 2부터 (N)까지의 각 i에 대하여 서버 S_i에 저장되어 있는 패턴을 서버 $S_{(i-1)}$로 재전송한다.서버 S_1에 저장되어 있는 패턴은 서버 S_N으로 재전송한다.
7) 서버 S_i에서 가지고 있는 복호화 키 DK_i로 데이터를 복호화하여 복호화된 빈번 항목을 생성한다.

3.2.2 길이가 2 이상인 빈번 패턴의 생성

N-저장소 서버 모델을 사용하여 발견한 빈번 항목, 즉 길이가 1인 패턴을 조합하여 길이 2 이상의 패턴을 생성해나가는 방법은 다음과 같다.

1) 3.2.1절에서 발견한 길이가 1인 빈번 패턴을 Apriori 알고리즘[12]을 적용하여 길이 2인 후보 패턴을 생성한 후, 후보 패턴의 실제 발생 여부를 판별하기 위한 질의를 각 사이트에 전송한다.
2) 각 사이트는 단계 1)의 질의를 처리하여 결과 값을 구한 후, 정보 유지 대체 기법을 사용하여 변형된 값을 서버로 전송한다. 해당 후보 패턴이 사이트에 존재하는 경우 1, 존재하지 않는 경우 0이 질의의 결과로 구해지며, 이는 0과 1로 양분화 된 값이므로 기존의 정보 유지 대체 기법을 적용하여 변형한 값을 응답함으로써 프라이버시를 보장한다.
3) 서버는 0과 1로 응답한 사이트의 수를 집계한 후 대체 확률 p와 함께 계산하여, 정보 유지 대체 기법에 의하여 변형되기 이전의 결과가 1이었을 사이트의 수를 예측함으로써 패턴이 존재하는 사이트의 수를 구한다.
4) 서버는 단계 3)에서 얻어진 후보 패턴이 존재하는 사이트의 수를 이용하여 각 후보 패턴의 지지도를 계산하고, 특정 지지도 이상이 되는 후보 패턴을 빈번 패턴으로 선택한다.
5) 현재까지 찾은 패턴 중 가장 긴 패턴들을 대상으로 Apriori 알고리즘을 적용하여 새로운 후보 패턴을 생성한다.
6) 더 이상 새로운 패턴이 발견되지 않을 때까지 단계 2)부터 5)까지를 반복한다.

제안된 기법을 통하여 패턴이 발생하는 실제 사이트에 대한 출처 정보를 은닉하여 프라이버시를 보장하면서도 순차 패턴 마이닝을 수행할 수 있다.

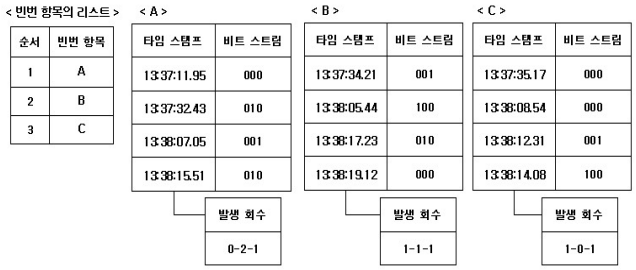
3.3 패턴 발생 여부 판별을 위한 인덱스 구조

3.2절에서 제안된 기법은 빈번 패턴을 생성하기 위하여 각 사이트에 접속하여 후보 패턴의 실제 발생 여부를 판별하는 과정이 필요하다. Apriori 알고리즘에서는 후보 패턴 "A->B->C"이 빈번 패턴이 되기 위해서는 반드시 모든 서브 패턴 "A->B", "A->C", "B->C"가 빈번하여야 하지만, 본 논문에서는 트래픽 데이터 항목 간의 간격이 최대 시간 간격 이하인 경우만을 패턴으로서 생성하므로, 만약 서브 패턴 "A->C"가 최대 시간 간격 보다 큰 값을 갖게 되어 빈번 여부가 성립하지 않더라도 서브 패턴 "A->B", "B->C"가 모두 빈번하면 후보 패턴 "A->B->C"는 빈번 패턴이 된다. 따라서, 기존의 Apriori 알고리즘 보다 더 많은 후보 패턴을 생성하게 되므로, 후보 패턴의 실제 발생 여부를 판별하는 과정에서 많은 오버헤드가 발생한다.

따라서, 본 논문에서는 각 패턴의 발생 여부를 판별하기 위한 질의를 효율적으로 처리하는 IPa(Item pair) 인덱스 구조를 제안하여 사용한다.

3.3.1 IPa 인덱스의 구조

IPa 인덱스는 빈번 항목들을 저장하고 있는 항목 리스트와 각 빈번 항목에 대하여 해당 항목이 발생한 이후에 최대 시간 간격 내에 나타나는 항목에 대한 발생 여부를 비트 스트림으로 표현하는 스트림 리스트로 구성된다. <그림 2>에 구성된 인덱스 구조의 예를 보인다.



<그림 2> IPa 인덱스의 구조

여기서, 타임 스탬프는 항목 리스트내의 각 항목에 대하여 트래픽이 발생한 시각을 의미한다. 비트 스트림에서 각 비트는 해당 항목이 발생 한 이후에 최대 시간 간격 내에 발생하는 항목들에 대한 정보를 표현한다. 즉, 비트 스트림을 구성하는 i번째 위치의 비트 값이 "1"인 경우는 해당 항목이 발생한 이후에 빈번 항목 리스트내의 i번째 위치에 있는 항목이 최대 시간 간격 내에 발생하였음을 의미하며, 발생하지 않은 경우에는 비트 값이 "0"이 된다. 예를 들어 <그림 2>의 "A"에 대한 스트림 리스트에서 타임 스탬프 "13:37:32.43"는 네트워크 상에서 "A"가 발생한 트래픽 시각을 나타내며, 비트 스트림 "010"은 "A"가 발생한 이후에 최대 시간 간격 내에 "A"는 발생하지 않고, "B"는 발생하고, "C"는 발생하지 않음을 의미한다. 이때, 발생된 항목들에 대한 발생 회수를 함께 저장한다. 예를 들어 "A"에 대한 스트림 리스트에서 발생 회수 "0-2-1"은 "A"가 발생한 후 최대 시간 간격 내에 "A"는 한 번도 발생하지 않았고, "B"는 2회, "C"는 1회 발생하였음을 의미한다.

IPa 인덱스는 사이트가 가지고 있는 데이터를 한번 스캔하는 과정에서, 각 항목과 최대 시간 간격 내에 있는 항목들을 검색하고 비트 스트림을 만든 후 다음 항목에 대한 처리를 진행하여 구축할 수 있다. 이렇게 구축된 IPa 인덱스는 각 사이트별로 별도로 저장, 관리되어 원본 데이터를 검색하지 않고도 서버로부터의 질의를 처리할 수 있다.

3.3.2 IPa 인덱스를 이용한 질의 처리

제안된 IPa 인덱스를 사용하여 후보 패턴의 발생 여부를 판별하기 위한 질의를 다음과 같은 과정을 통하여 처리할 수 있다.

1) 질의 패턴을 슬라이딩 방식을 이용하여 길의 2의 서브 패턴들로 분리한다.
2) 분리된 각 서브 패턴에 대하여 서브 패턴내의 첫 번째 항목에 해당하는 스트림 리스트를 검색하여 서브 패턴내의 두 번째 항목을 만족하는 엔트리와 발생 회수를 검색한다. 즉, 비트 스트림내에서 두 번째 항목에 해당하는 비트 값이 1인 엔트리를 검색한다.
3) 단계 2)에서 얻어진 모든 엔트리에 대해서 인접한 두 개의 엔트리간의 시간 간격이 성립하는지의 여부를 판별하기 위하여 조인 연산을 수행한다. 이때, 조인 비용을 감소하기 위하여 발생 회수가 작은 엔트리에 대하여 먼저 조인 연산을 수행한다.
4) 질의 패턴을 구성하는 모든 서브 패턴들에 대하여 조인 연산을 수행한 결과, 시간 간격을 만족하는 하나 이상의 엔트리가 존재하는 경우 해당 질의 패턴이 사이트에 존재함을 의미한다.

예를 들어 최대 시간 간격을 2초로 하였을 때, "A->B->C" 패턴이 각 사이트내에 실제 발생하는지의 여부를 판별하기 위한 질의 처리 과정은 다음과 같다. 우선 패턴을 슬라이딩 방식을 이용하여 길의 2의 서브 패턴 "A->B", "B->C"로 분리한다. 다음 분리된 각 서브 패턴의 첫 번째 항목에 해당하는 스트림 리스트에서 두 번째 항목의 발생 회수를 계산한다. 즉, 서브 패턴 "A->B"를 처리하기 위하여 "A"의 스트림 리스트에서 "B"의 발생 회수 2와 이를 만족하는 엔트리 2와 4를 얻는다. 서브 패턴 "B->C"를 처리하기 위하여 "B"의 스트림 리스트에서 "C"의 발생 회수 1과 이를 만족하는 엔트리 1을 얻는다. 얻어진 각 엔트리 사이의 시간 간격 성립 여부를 순차적으로 검사한 결과, 만족되는 엔트리가 있으므로 패턴 "A->B->C"가 해당 사이트내에서 발생한다고 판단할 수 있다.

3.4 존재하지 않는 패턴에 대한 캐쉬 리스트의 관리

각 사이트에서 후보 패턴이 발생하는지의 여부에 대한 질의는 Apriori 알고리즘에 의해서 생성된 길이 2인 후보 패턴들부터 길이가 1씩 증가되면서 생성된 후보 패턴들에 대하여 이루어진다. 이 과정에서 길이 k인 후보 패턴이 빈번하지 않다면 이 후보 패턴으로부터 생성되는 길이 (k+1)인 후보 패턴은 반드시 빈번하지 않음에도 불구하고, 후보 패턴을 분해한 각 서브 패턴에 대하여 반복적인 질의 처리 과정이 필요하다. 본 논문에서는 이러한 반복적인 질의 처리에 대한 오버헤드를 제거하기 위하여 각 사이트에서 발생하지 않는 패턴에 대하여 별도의 캐쉬 리스트에 저장, 관리하여, 리스트에 있는 패턴으로부터 생성된 후보 패턴의 발생 여부에 대한 질의를 각 사이트에서 판별할 때 즉시 해당 패턴이 각 사이트에서 발생하지 않음을 서버에 알려 불필요한 질의 처리 과정을 제거하도록 한다.

이러한 캐쉬 리스트를 관리하는 방법은 다음과 같다.

1) 길이가 1인 빈번 패턴 중 사이트 내에서 발생하지 않은 패턴을 캐쉬 리스트에 추가한다.
2) 후보 패턴의 발생 여부에 대한 질의가 사이트에 주어졌을 때 후보 패턴에서 첫 번째 항목을 제거한 패턴과 후보 패턴에서 마지막 항목을 제거한 패턴에 대해서 각 패턴이 캐쉬 리스트에 존재하는지의 여부를 검색한다.
3) 단계 2)의 두 가지 패턴에 대하여 캐쉬 리스트에 있으면 해당 후보 패턴은 각 사이트에서 발생하지 않는 패턴이 되며, 해당 후보 패턴을 캐쉬 리스트에 추가한다. 만약, 캐쉬 리스트에 두 가지 패턴이 모두 없는 경우에는 IPa 인덱스를 사용하여 후보 패턴의 발생 유무를 검색하고, 검색 결과 후보 패턴이 사이트에서 발생하지 않는 경우에는 캐쉬 리스트에 해당 후보 패턴을 추가한다.

캐쉬 리스트에서는 길이가 1인 후보 패턴에 대한 질의시, 길이가 (l-1)인 패턴만을 이용하게 되므로, 후보 패턴의 길이가 1이 되는 순간, 캐쉬 리스트 내에 존재하는 길이가 (l-2)인 패턴을 모두 제거함으로써 캐쉬 리스트가 무한히 커지는 것을 방지한다.

4. 결론

본 논문에서는 대용량 네트워크 트래픽 데이터를 대상으로 사이트의 프라이버시를 보호하면서 마이닝 결과의 정확성, 실용성 등을 보장할 수 있는 효율적인 순차 패턴 마이닝 기법을 제안하였다.

제안된 기법을 활용하여 네트워크 트래픽 상의 정상 상태의 패턴과 침입 상황 등과 같은 비정상 상태의 패턴을 발견할 수 있으며, 이를 통하여 본 연구 결과를 침입 탐지, 인터넷 웜의 검출 등의 응용 분야에 활용할 수 있다. 또한 다수의 사용자에 대한 웹 사용 패턴 분석을 통하여 웹 서버에 대한 로드 밸런싱을 처리는데 활용할 수 있다.

본 논문의 공헌은 다음과 같다.

1) 본 논문에서는 대용량 네트워크 트래픽 데이터에 대하여 프라이버시를 보장할 수 있는 마이닝 기법에 대한 연구를 수행하였다.
2) 이를 위하여 해당 패턴이 발생한 각 사이트에 대한 출처를 은닉하여 프라이버시를 보호하는 기법과 N-저장소 서버 모델을 사용한 효율적인 마이닝 기법을 제안하였다.
3) 또한, 각 사이트에서의 후보 패턴 발생 여부를 판별하기 위한 질의를 효율적으로 처리할 수 있는 IPa 인덱스 구조와 캐쉬 리스트를 제안하였다.

참고문헌

[1] W. Lee, S. Stolfo, and K. Mok, "A Data Mining Framework for Building Intrusion Detection Models," IEEE Symposium on Security and Privacy, 1999.

[2] S. Song, Z. Huang, H. Hu, and S. Jin, "A Sequential Pattern Mining Algorithm for Misuse Intrusion Detection," International Workshop on Information Security and Survivability for Grid (GISS2004), 2004.

[3] P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, and P. Tan, "Data Mining for Network Intrusion Detection," Proc. NSF Workshop on Next Generation Data Mining, 2002.

[4] S. Rizvi and J. Haritsa, "Maintaining Data Privacy in Association Rule Mining," In Proceedings of the 28th Conference on Very Large Data Base (VLDB'02), 2002.

[5] C. Clifton and D. Marks, "Security and Privacy Implication of Data Mining," Proceedings of the 1996 ACM Workshop on Data Mining and Knowledge Discovery, 1996.

[6] R. Agrawal and R. Srikant, "Privacy-preserving data mining," Proceedings of the 2000 ACM SIGMOD international conference on Management of data, 2000.

[7] A. Evfimievski, R. Srikant, R. Agrawal, and J. Gehrke "Privacy Preserving Mining of Association Rules," Proc. of 8th ACM SIGKDD Intl. Conf. on Knowledge, 2002.

[8] Y. Lindell and B. Pinkas, "Privacy Preserving Data Mining," Proceedings of the 20th Annual International Cryptology Conference on Advances in Cryptology, August 2000.

[9] M. Kantarcioglu and C. Clifton, "Privacy-preserving distributed mining of association rules on horizontally partitioned data," In The ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, 2002.

[10] J. Zhan, L. Changy, and S. Matwinz, "Privacy-Preserving Collaborative Sequential Pattern Mining," Proceedings of Workshop on Link Analysis, Counter-terrorism and Privacy, 2004.

[11] R. Agrawal and R. Srikant and D. Thomas, "Privacy preserving OLAP," Proceedings of the 2005 ACM SIGMOD international conference on Management of data, 2005.

[12] R. Agrawal and R. Srikant, "Mining sequential patterns," In Proceedings of the 11th International Conference on Data Engineering, 1994.