실시간 네트워크 침입탐지 시스템을 위한 아웃라이어 클러스터 검출 기법☆

An Outlier Cluster Detection Technique for Real-time Network Intrusion Detection Systems

장 재 영* 김 한 준** 박 종 명***
Jae-young Chang Han-joon Kim Jongmyoung Park

요 약

최근의 네트워크 침입탐지 시스템은 기존의 시그너처(또는 패턴) 기반 탐지 기법에 비정상행위 탐지 기법이 새롭게 결합되면서 더욱 발전되고 있다. 일반적으로 시그너처 기반 침입 탐지 시스템들은 기계학습 알고리즘을 활용함에도 불구하고 사전에 이미 알려진 침입 패턴만을 탐지할 수 있었다. 이상적인 네트워크 침입탐지 시스템을 구축하기 위해서는 침입 패턴이 저장된 시그너처 데이터베이스를 항상 최신의 정보로 유지해야 한다. 따라서 시스템은 유입되는 네트워크 데이터를 모니터링하고 분석하는 과정에서 새로운 공격에 대한 시그너처를 생성할 수 있는 기능이 필요하다. 본 논문에서는 이를 위해 밀도(또는 영향력) 함수를 이용한 새로운 아웃라이어 클러스터 검출 알고리즘을 제안한다. 제안된 알고리즘에서는 네트워크 침입 패턴을 하나의 객체가 아닌 유사 인스턴스들의 집합 형태인 아웃라이어 클러스터로 가정하였다. 본 논문에서는 KDD 1999 Cup 침입탐지 데이터 집합을 이용한 실험을 수행하여, 침입이 자주 발생하는 상황에서 본 논문의 방법이 유클리디언 거리를 이용한 기존의 아웃라이어 탐지 기법에 비해서 좋은 성능을 보임을 증명하였다.

Abstract

Intrusion detection system(IDS) has recently evolved while combining signature-based detection approach with anomaly detection approach. Although signature-based IDS tools have been commonly used by utilizing machine learning algorithms, they only detect network intrusions with already known patterns. Ideal IDS tools should always keep the signature database of your detection system up-to-date. The system needs to generate the signatures to detect new possible attacks while monitoring and analyzing incoming network data. In this paper, we propose a new outlier cluster detection algorithm with density (or influence) function. Our method assumes that an outlier is a kind of cluster with similar instances instead of a single object in the context of network intrusion. Through extensive experiments using KDD 1999 Cup Intrusion Detection dataset, we show that the proposed method outperform the conventional outlier detection method using Euclidean distance function, specially when attacks occurs frequently.

☞ keyword : network intrusion system, outlier detection algorithm, cluster, density function, 네트워크 침입탐지 시스템, 아웃라이어 탐지 알고리즘, 클러스터, 밀도 함수

1. 서 론

네트워크 침입(network intrusion)이란 불순의 의도를 가진 해커 또는 크래커(cracker)가 네트워크를 통해 목표 시스템에 접근하여 유용한 정보를 접근, 파괴 및 조작 등을 하는 행위를 말한다. 이러한 보안 사고가 다양해짐에 따라 유해한 침입

* 정 회 원 : 한성대학교 컴퓨터공학과 부교수
jychang@hansung.ac.kr
** 정 회 원 : 서울시립대학교 전자전기컴퓨터공학부 조교수
khj@uos.ac.kr(교신저자)
*** 준 회 원 : 주식회사 위트콤 사원
dustjm@witcom.co.kr
[2007/05/21 투고 - 2007/05/28 심사 - 2007/10/04 심사완료]
☆ 본 연구는 2007년도 한성대학교 교내연구비 지원과제이며, 또한 정보통신부 및 정보통신연구진흥원의 대학IT연구 센터 육성지원사업(IITA-2007-C1090-0701-0031)의 연구결과로 수행되었음.

으로부터 컴퓨터나 네트워크의 안정적 유지를 위한 침입탐지 시스템(intrusion detection system: IDS)의 중요성이 점차 높아지고 있으며, 최근에는 무선네트워크에서 침입탐지를 방지하기 위한 연구가 진행되고 있다[1]. 침입탐지 시스템은 침입탐지 방식에 있어서 기본적으로 시그너처 기반 탐지(signature-based detection)와 비정상행위 탐지(anomaly detection)의 두 가지 방법이 존재한다. 시그너처 기반 탐지는 이미 알려진 침입 행위에 대한 정보를 이용하여 공격을 탐지하는 방식을 말하며, 비정상행위 탐지는 사용자의 정상 행위를 기반으로 정상 패턴에 어긋나는 경우를 침입으로 간주하는 기법을 말한다. 이중에서 기존 대부분의 침입탐지 시스템 도구들은 시그너처 기반 탐지 방법을 사용하고 있다. 그 이유는 비정상행위 탐지 방식은 정상적인 행위임에도 불구하고 침입 행위로 잘못 인식하여 경보를 발령하는 경우가 많아 실용적으로 사용하기에 많은 문제점을 가지고 있기 때문이다 [2][3][4]. 시그너처 기반 탐지 기법에서는 이미 알려진 침입 패턴이 침입 시그너처로 표현되고 저장된 시그너처와 일치되는 침입 패턴이 발견되면 경보를 발령하게 된다. 최근에는 의사결정트리(Decision Trees), 인공신경망(Artificial Neural Networks), Support Vector Machine (SVM)과 같은 다양한 기계학습 알고리즘(machine learning algorithm)을 이용하여 사전에 알려진 침입 행위를 학습함으로써 자동으로 침입 시그너처를 생성하기도 한다[3][5][6][7].

이미 알려진 침입 패턴만을 탐지할 수 있는 시그너처 기반 탐지 기법은 정확한 탐지가 가능하다는 장점이 있지만, 이는 반대로 심각한 문제점이 될 수도 있다. 그 이유는 새로운 형태의 침입을 탐지하기 위해서는 그와 대응되는 침입 시그너처를 생성하고 침입탐지 시스템의 데이터베이스에 추가되어야만 가능한데, 이러한 새로운 침입 시그너처가 추가되기 전까지는 동일한 형태의 침입에 대응할 방법이 없기 때문이다. 특히 이런 방식은 새로운 형태의 공격이 자주 발생하는 네트워크 환경에서는 실효성을 거두기가 어려울 수밖에 없다.

이상적인 침입탐지 시스템 도구는 침입탐지를 위한 정보를 담고 있는 시그너처 데이터베이스를 최신의 상태로 유지해야한다. 따라서 최근 침입탐지 시스템은 현재 유입되고 있는 네트워크 데이터를 모니터링하고 분석하여 새로운 공격 패턴을 검출하여 실시간으로 시그너처를 생성하는 기능을 갖고 있다[8][9]. 실제로 네트워크 데이터에서 새로운 침입 패턴은 아웃라이어 데이터들의 모임 형태로 나타나며 이는 아웃라이어 탐지 알고리즘(outlier detection algorithm)을 이용하여 검출해 낼 수 있다[10]. 문제는 기존의 아웃라이어 탐지 기법들은 단일 인스턴스(instance)로서의 아웃라이어를 탐지하는데 중점을 두었다는 것이다. 반면에 네트워크 침입을 암시한 데이터들은 네트워크 데이터 상에서 유사한 인스턴스들의 '군집' 형태로 나타난다. 따라서 기존 아웃라이어 탐지 알고리즘을 이러한 집합을 탐지하는데 직접적으로 적용하기에는 많은 문제점이 존재한다.

이러한 점을 감안하여, 본 논문에서는 그룹 형태의 침입데이터 집합을 효율적으로 탐지하기위해 밀도 함수(density function) - 또는 영향력 함수(influence function) - 에 기반한 새로운 아웃라이어 탐지 기법을 제안한다. 본 논문에서는 아웃라이어를 하나의 인스턴스가 아닌 유사한 인스턴스들로 구성된 클러스터(cluster)로 가정하였다. 즉, 정상적인 클러스터들과는 다른 '아웃라이어 클러스터(outlier cluster)'를 검출하여 이 클러스터에 포함된 인스턴스들을 네트워크 침입을 위한 패킷들로 판단하는 방법을 사용하였다. 기존의 아웃라이어 클러스터를 검출하는 방법으로는, 중심 클러스터와의 유클리디어거리(Euclidean distance)를 이용한 방법이 있었으나[11], 이는 중심 클러스터와 멀리 떨어진 작고 희소한 클러스터만을 검출하는 단점이 있다. 본 논문에서는 실제 침입패턴 데이터의 군집성을 감안하여 중심 클러스터와의 거리뿐만 아니라 주변 클러스터들의 영향력을 고려한

함수를 정의하였다. 이 함수를 이용하여 클러스터링 과정에서 생성되는 클러스터의 집합에서 아웃라이어 클러스터를 효과적으로 검출하는데 활용하였다. 특히 이 함수는 짧은 시간에 다량의 침입 패킷으로 형성된 밀도가 높은 아웃라이어 클러스터를 탐지하는데 유용하게 사용할 수 있었다. 본 논문에서는 제안된 기법의 성능을 평가하기 위해서 KDD CUP 1999 데이터 집합[16]을 이용한 실험을 실시하였다. 평가 기준은 정확도(precision)와 재현율(recall)을 결합한 F1-측정치를 이용하였으며, 유클리디언 거리 함수를 이용한 기존의 기법보다 우수함을 실험적으로 증명하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문에서 가정한 기본적인 침입탐지 시스템의 아키텍처와 관련된 주요 이슈를 소개한다. 3장에서는 본 논문에서 제안하는 밀도 함수와 아웃라이어 탐지 기법을 제시한다. 4장에서는 실험 결과를 제시하고 마지막으로 5장에서는 결론 및 향후 연구 과제를 기술한다.

2. 침입탐지 시스템 모델

본 장에서는 제안하는 침입탐지 기법의 모델이 되는 실시간 침입탐지 시스템의 기본 아키텍처를 소개한다. 제안된 기본 시스템 아키텍처는 그림 1과 같으며 다음과 같은 구성 요소를 갖는다.

우선 **패킷수집 모듈(Packet Capture Module)**은 네트워크상에서 TCP/IP 혹은 기타 프로토콜에 의해 전송중인 패킷(packet)들을 수집하며, 수집된 패킷들을 **패킷 저장소(Packet Storage)**에 저장된다. 본 연구에서는 promiscuous 모드를 지원하는 네트워크 디바이스를 사용하여 패킷 데이터를 수집하였다. 수집된 데이터는 저장소에 저장되기 전에 **클러스터링 모듈(Clustering Module)**에서 처리될 수 있도록 미리 정의된 형식으로 변환된다. **아웃라이어 탐지 모듈(Outlier Detection Module)**은 **클러스터링 모듈**에서 생성된 클러스터들 중에서

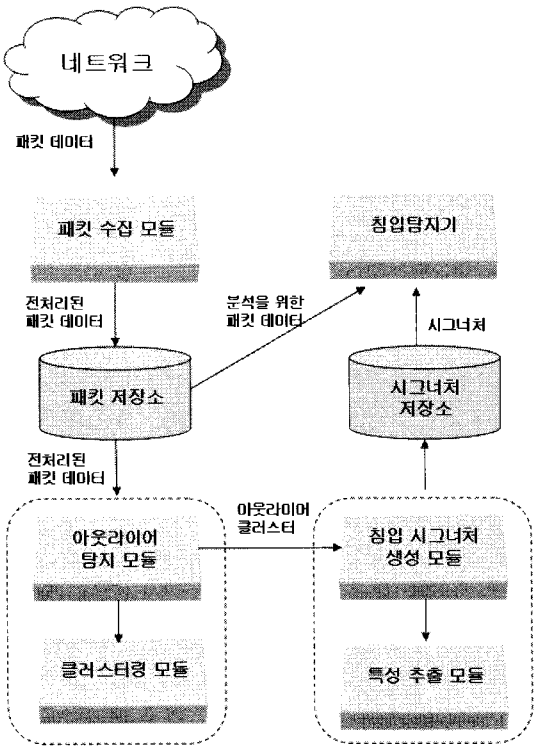


그림 2 실시간 네트워크 침입탐지 시스템의 기본 아키텍처

네트워크 침입으로 규정할 수 있는 아웃라이어 클러스터를 검출하는 역할을 한다. 앞서 언급한 바와 같이 네트워크 침입은 침입 인스턴스들의 집합으로 나타난다. 따라서 우선 **클러스터링 모듈**이 패킷 데이터들로부터 클러스터 집합을 생성하고, **아웃라이어 탐지 모듈**은 클러스터 집합에서 밀도함수에 기반하여 아웃라이어 클러스터를 검출하는 역할을 담당한다. 본 연구에서는 **클러스터링 모듈**에서 네트워크 패킷데이터들을 클러스터링하기 위해 K-means 알고리즘[11]을 사용한다.

침입 시그너처 생성 모듈(Intrusion Signature Generation Module)은 발견된 아웃라이어 클러스터를 구별해 낼 수 있는 특징들을 정형화된 형식으로 표현하는 침입 시그너처(또는 규칙)를 생성한다. 예를 들어, 침입 시그

너처는 다음과 같은 IF-THEN 형태로 표현될 수 있다.

IF protocol = 'icmp' AND port = 1380 AND length <= 211 THEN 'ATTACK'

이 시그너처의 의미는 protocol값이 'icmp'이고 port번호가 1380이고, 패킷길이가 211이하이면, 네트워크 침입으로 판별하는 규칙이다. 이러한 침입 규칙들은 **특성추출 모듈(Feature Extraction Module)**이 아웃라이어 클러스터의 주요 특성을 추출함으로써 생성할 수 있다. 침입 규칙의 각 특성은 특성이름(혹은 속성)과 값의 쌍으로 표현되는데, 예를 들면, 아웃라이어 클러스터는 protocol = 'icmp' AND port = 1380 AND length <= 211 와 같은 특성을 갖는 것이다. 이 모듈을 위해 본 논문에서는 의사결정 트리를 이용한 기계학습 알고리즘을 활용하였다[6][13]. 이와 같은 과정으로 생성된 침입 시그너처들은 **시그너처 저장소(Signature Storage)**에 저장되어 침입 여부를 판별하는 기준으로 활용된다. 마지막으로 **침입탐지기(Intrusion Detector)**는 **패킷 저장소**에 저장된 네트워크 패킷 데이터를 **시그너처 저장 모듈**에 저장된 침입 시그너처와 실시간 대조하여 침입 여부를 분석, 판단하여 경보를 발령하거나 보고하는 역할을 한다.

본 논문의 제안 기법을 실제로 적용한 **아웃라이어 탐지 모듈**의 적정성 여부와 성능실험을 위해 그림 1의 아키텍처를 구현한 프로토타입 시스템을 개발하였으며, 시스템의 구성요소 중에서 본 논문에서 제안하는 아웃라이어 클러스터 검출 기법은 **아웃라이어 탐지 모듈**에서의 주요 구성요소로 활용된다.

3. 아웃라이어 클러스터 탐지기법

3.1 네트워크 침입 패턴으로서의 아웃라이어 클러스터

아웃라이어란 대부분의 일반적인 객체(또는 값)들과 구별되는, 비정상적이고 특이한 객체를 의미한다. 따라서 아웃라이어를 검출한다는 것은 일반적 객체들과는 상이한 객체를 찾는 행위를 의미한다. 일반적으로 아웃라이어 탐지를 위한 방법으로 통계기반(statistics-based) 기법, 거리기반(distance-based) 기법 그리고 모델기반(model-based) 기법 등의 세 가지 접근 방법이 존재한다[14]. 그러나 이 기법들은 아웃라이어를 어느 클러스터에도 속하지 못한 하나의 객체(또는 인스턴스)로서 정의한다.

많은 경우에 네트워크 침입은 유사한 인스턴스들의 집합 형태로 나타나는데, 실제로 유사하거나 동일한 네트워크 침입들은 일정 시간동안 집중적으로 발생하는 경향이 있다. 결국 이런 침입 네트워크 패킷들은 유사한 데이터들의 클러스터 형태를 갖게 된다. 클러스터라는 것은 서로 강하게 연관된 유사한 객체들의 그룹을 의미한다. 예를 들어, **'SYN flooding attack'**에 해당하는 네트워크 패킷들은 밀집된 형태의 클러스터를 형성할 수 있으며, 특히 **'SYN flooding attack'**의 한 종류인 **'Denial-of-Service (DoS) attack'**의 경우 많은 양의 네트워크 트래픽을 동반하므로 다소 큰 형태의 클러스터를 형성하게 된다. 따라서 네트워크 환경에서 아웃라이어는 하나의 인스턴스가 아닌 여러 인스턴스들의 집합형태인 클러스터로 확장하는 것이 바람직하다. 이러한 아웃라이어 클러스터는 보통 정상적 데이터를 담고 있는 주요 클러스터로부터 거리가 크면서 밀도가 큰 클러스터로 나타나게 된다. 결국 네트워크 침입탐지는 패킷 데이타로부터 아웃라이어 클러스터를 검출하는 것으로 침입여부를 판단 할 수 있다.

아웃라이어 클러스터를 검출하기 위해서 [11]에서는 간단히 유클리디언 거리함수를 이용한 방법을 제안하였다. 그 과정은 다음과 같다. 첫 단

계는 K-means 클러스터링 기법을 이용하여 주어진 데이터에 대해 클러스터링을 수행한다. 둘째 단계에서는 생성된 클러스터 중에서 정상적 데이터를 담고 있는 것으로 추정되는 주요 (또는 중심) 클러스터를 결정한다. 마지막 단계에서는 둘째 단계에서 판별된 주요 클러스터와 가장 멀리 떨어진 클러스터를 아웃라이어 클러스터로 판별한다. 특히 특정 임계값을 정해두고 이 값을 초과하여 멀리 떨어진 하나 이상의 클러스터들을 아웃라이어 클러스터들로 판별할 수도 있다. 여기서 두 클러스터간의 거리는 각 클러스터의 중심 간의 거리로 측정한다.

[11]에서 제시한 방법은 비교적 간단하게 아웃라이어 클러스터를 검출할 수 있다는 장점이 있으나 클러스터의 다른 특성들(예를 들어 크기나 밀도)등을 고려하지 않고 단순히 거리만으로 아웃라이어 클러스터를 판별하므로 만족할 만한 정확도를 보이지 못한다. 예를 들어, 주변의 인접한 클러스터가 많더라도 중심 클러스터와 멀리 떨어져 있다면 아웃라이어 클러스터로 결정될 수 있으며, 주변에 인접한 클러스터가 없더라도 오로지 중심 클러스터와의 거리가 가깝다는 이유만으로 아웃라이어 클러스터로 검출되지 않을 수도 있기 때문이다.

이 문제를 해결하기 위해 본 논문에서는 주변의 다른 클러스터로 부터의 밀도(혹은 영향력)를 고려한 방법을 이용하였다. 이전 연구에서는 밀도 개념을 데이터 포인트(또는 클러스터) 주위에 있는 데이터 포인트들의 수로서 정의하였는데 [15][16], 본 연구에서는 클러스터의 밀도를 주위 클러스터의 영향력 정도를 합산한 값으로 정의하였다.

3.2 밀도 기반 아웃라이어 클러스터 검출 기법

본 절에서는 아웃라이어 클러스터를 검출하는데 필요한 밀도 함수를 제안한다. 밀도 함수는 영향력 함수를 기반으로 하는데, 영향력 함수는 주어진 객체그룹 내에서 특정 객체가 이웃 객체들에 미치는 영향 또는 효과의 정도를 표현하는데 사용된다. 주어진 객체집합 S에 대해서 객체 $o_y \in S$에 대한 객체 $o_x \in S$의 영향력 ($\Omega^{o_x}(o_y)$)은 가우시안(Gaussian) 확률분포와 유사한 다음과 같은 수식으로 표현 할 수 있다.

$$\Omega^{o_x}(o_y) = e^{-\frac{|o_x - o_y|^2}{2\sigma^2}} \quad (1)$$

여기서 σ는 밀도 함수의 형태를 결정하는 제어 인자(control parameter)이고, $|o_x - o_y|$는 o_x 객체 o_y와 간의 유클리디언 거리이다. 가우시안 영향력 함수는 DENCLUE 클러스터링 알고리즘 [17]에서 제안되었는데, 이 함수는 두 객체간의 거리가 가까워질수록 영향력은 급격히 강해지며, 역으로 거리가 멀어질수록 영향력 정도가 현저하게 떨러지는 특징을 갖는다. 따라서 특정 객체에 대한 밀도를 계산할 때 그 객체와 근거리에 존재하는 객체만을 고려해도 전체적인 밀도 함수 값과 차이는 거의 없다.

이제 식(1)을 이용하여 밀도 함수를 정의할 수 있다. 다음 식 (2)에서 보는 바와 같이 주어진 객체 집합 $S=\{o_1, o_2, \ldots, o_i, \ldots\}$에 대해서, 객체 o_y에 대한 S의 밀도($\Omega^S(o_y)$)는 S에 존재하는 각 객체 o_i로부터 o_y대한 영향력의 합으로 정의한다.

$$\Omega^S(o_y) = \Sigma_{o_i \in S} \Omega^{o_i}(o_y) = \Sigma_{o_i \in S} e^{-\frac{|o_x - o_i|^2}{2\sigma^2}} \quad (2)$$

본 논문에서는 유사한 객체들로 구성된 아웃라이어 클러스터를 판별하는 것이 목적이므로 하나

의 객체에 대한 밀도수가 아닌 클러스터에 대한 밀도 함수 - 즉, 클러스터 밀도 함수 - 를 계산해야한다. 클러스터 밀도 함수는 클러스터에 존재하는 모든 객체들에 대한 밀도 함수 값의 합으로 계산된다. 즉, 객체 집합 S를 파티션한 클러스터의 집합 $C = \{c_1, c_2, \ldots, c_i, \ldots\}$가 있다고 가정할 때, 데이터 공간 S에서 클러스터 c_i에 대한 클러스터 밀도 함수는 다음과 같이 정의한다.

$$\Omega^S(c_i) = \Sigma_{o_x \in c_i} \Omega^S(o_x) \quad (3)$$

이 함수는 클러스터간의 거리뿐만 아니라 각 개체의 밀도까지 모두 고려한 것으로 아웃라이어 클러스터를 검출하는데 효율적으로 활용될 수 있다. 즉, 모든 클러스터들 중에서 아웃라이어 클러스터는 클러스터 밀도 함수 값을 기준으로 하여 가장 작은 영향력을 갖는 것으로 판정하는 것이다. 아웃라이어 클러스터를 검출하는 과정은 [11]에서 제안한 방법과 거의 유사하다. 차이점은 마지막 단계에서 [11]는 유클리디언 거리함수를 사용한다는 것이고 본 논문에서 제안한 방법에서는 클러스터 밀도 함수를 사용한다는 점이다.

4. 성능 분석

3장에서 제안된 방법에 대한 성능을 평가하기 위해서 본 논문에서는 침입탐지 시스템 시스템을 평가하는데 가장 일반적으로 이용되는 KDD 1999 Cup Intrusion Detection 데이터 집합[12]을 사용하였다. 이 데이터 집합은 1998년 DARPA Intrusion Detection Evaluation Program에서 사용된 데이터를 토대로 몇 가지 속성을 추가하여 생성한 데이터 집합이다. 본 실험에서 사용한 DARPA 테스트 데이터는 2주간의 네트워크 트래픽으로 구성된 TCP Dump 데이터로서, 약 2,000,000개의 데이터 인스턴스로 구성되어 있다. 그리고 이는 DoS(Denial-of-Service), R2L(Remote-to-Local), U2R(User-to-Root), Probing[1] 등을 포함한 잘 알려진 24종류의 침입 패턴을 포함한다. 본 실험에서는 전체 데이터 패킷 중에서 침입 패킷이 적은 경우와 많은 경우에 대해서 실시하였으며, 각 경우의 대표적 침입 패킷의 비율을 1%와 23%가 되도록 생성하였다.

본 실험을 위해서, 데이터베이스 관리 시스템으로 MySQL을 사용하였으며, 아웃라이어 탐지 모듈을 구현하기 위해서 Weka 데이터마이닝 워크벤치[2][13]에 포함된 SimpleKMeans 클러스터링 클래스를 활용하였다. 텍스트 형태의 DARPA 테스트 데이터는 속성별로 파싱(parsing)하여 하나의 관계형 테이블에 저장한다. 데이터 입력을 위해서 데이터베이스에 저장된 테스트 데이터를 Weka엔진에 입력되는 ARFF 포맷 파일[3] 형태로 변환이 필요하며, 이 파일을 아웃라이어 탐지 모듈에 입력하게 된다. 아웃라이어 클러스터를 결정하기 위한 거리 임계값은 적정한 범위 내에서 고정하며, 이 값을 기준으로 클러스터 개수에 따라 아웃라이어 탐지 실험이 반복적으로 이루어진다.

제안기법의 성능은 두 가지 관점에서 평가될 수 있다. 첫째는 제안된 기법에 의해 검출된 아웃라이어 클러스터 내에서 실제로 얼마나 많은 침입 데이터들이 포함되었느냐를 평가하며, 둘째는 전체 패킷 중에 얼마나 많은 실제 침입 패킷들이 제안된 알고리즘에 의해 검출되느냐를 평가한다. 흔히 전자를 정확도(precision)라 하고 후자를 재현율(recall)이라 한다. 이는 정보검색 시스템의 성

1) DoS는 침입거부 공격, R2L은 원격지에서의 인가되지 않은 접근, U2R은 Administrator 또는 root로의 인가되지 않은 접근, Probing은 시스템의 취약점의 탐사를 의미한다.

2) 호주 Waikato대학에서 개발한 데이터마이닝 알고리즘 라이브러리로서, 다양한 알고리즘을 체계적으로 구성해놓았으며, 관련 사이트는 http://www.cs.waikato.ac.nz/ml/weka/ 임

3) ARFF 파일은 Weka 데이터마이닝 워크벤치에서 사용되는 특수 포맷의 텍스트 파일로서, 입력 데이터의 상단에 입력내용의 메타정보 (예: 컬럼명, 컬럼타입, 컬럼개수 등)가 포함되어 있다.

능평가 척도로 활용되는 것으로서, 다음과 같이 정의한다.

$$Precision = \frac{N_{OutlierIntrusions}}{N_{OutlierPackets}} \quad (4)$$

$$Recall = \frac{N_{OutlierIntrusions}}{N_{TotalIntrusions}} \quad (5)$$

이 식에서 $N_{OutlierIntrusions}$는 아웃라이어 클러스터내의 실제 침입 패킷의 수를 나타내며, $N_{TotalIntrusions}$은 전체 테스트 데이터 중에서 침입 패킷의 수를 나타낸다. 그리고 $N_{OutlierPackets}$는 아웃라이어 클러스터 내의 전체 패킷 수를 나타낸다. 본 논문에서는 단일 측정치를 제시하기 위해 위에서 기술한 재현율과 정확도를 하나로 통합한 F-측정치(F-measure)를 사용하였으며, 이는 아래와 같이 정의한다.

$$F = \frac{(1+\beta^2)\times recall \times precision}{\beta^2(recall + precision)} \quad (6)$$

F-측정치에서 재현율과 정확도의 중요도는 β 값을 통해 조절할 수 있으며, 본 실험에서는 두 인자간의 중요도를 균등하게 설정하여 β값을 1로 설정한다. 이는 결국 재현율과 정확도의 조화평균값을 의미한다. 이 측정치는 0부터 1의 값을 가지며, 1에 가까울수록 고성능을 의미한다.

본 논문에서는 아웃라이어 클러스터 검출을 위해 기본 클러스터링 알고리즘을 K-means[11]를 사용하였다. 이 알고리즘은 시간뿐만 아니라 공간 측면에서도 선형(linear) 내지는 선형에 근접한 복잡도를 갖고 있어 클러스터링 과정을 매우 효율적으로 처리한다. 그러나 이는 클러스터링 전에 최종 클러스터 개수를 자체적으로 결정하지는 못하기 때문에 본 실험에서 10개에서 30개 사이에서 클러스터들의 수를 변화시켜 가면서 반복적인 실험을 수행하였다.

하지만 실제의 침입탐지 시스템을 구동하기 위해서는 자동으로 클러스터 개수를 결정하는 것이 요구된다. 클러스터의 개수를 결정하는 기본적인 원칙은 클러스터의 질(quality)이 가장 높을 때의 개수를 취하는 것이다. 클러스터의 질을 측정하는 여러 방안 중에서 직관적인 방법은 클러스터 내부의 각 인스턴스들간의 평균 거리와 클러스터간의 평균거리를 계산하는 것이다[18]. K-means 알고리즘의 경우, 클러스터 개수를 달리 하면서 클러스터링을 수행할 때, 각 수행에서 생성된 클러스터들의 질을 평가하여 임계점을 초과할 때를 적정 개수로 볼 수 있다. 중요한 점은 클러스터의 질을 정량화하는 여러 방안 중에서 침입탐지율과 상관성이 큰 것을 사용해야 하는 것이다. 이에 대한 구체적인 방안은 향후 해결해야할 연구과제중 하나이다.

아웃라이어 클러스터의 판별 기준은 중심 클러스터와의 거리 임계값의 변화시키면서 아웃라이어 클러스터를 결정하였으며, 최소 하나의 아웃라이어 클러스터는 존재한다는 가정하에 실험을 실

표 1. 알고리즘의 성능 비교(침입패킷 비율이 1%일 경우)

알고리즘 구분 \ 클러스터 개수		10	12	14	16	18	20	22	24	26	28	30
유클리디언 거리 기반 알고리즘	Outlier 개수	212	541	186	541	206	206	202	202	185	202	202
	Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Recall	0.71	0.28	0.81	0.28	0.73	0.73	0.74	0.74	0.81	0.74	0.74
	F1값	0.83	0.44	0.90	0.44	0.84	0.84	0.85	0.85	0.90	0.85	0.85
밀도함수기반 알고리즘 (제안알고리즘)	Outlier 개수	212	541	186	541	206	206	202	202	185	202	202
	Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Recall	0.71	0.28	0.81	0.28	0.73	0.73	0.74	0.74	0.81	0.74	0.74
	F1값	0.83	0.44	0.90	0.44	0.84	0.84	0.85	0.85	0.90	0.85	0.85

표 2. 알고리즘의 성능 비교 (침입패킷 비율이 23%일 경우)

알고리즘 구분 \ 클러스터 개수		10	12	14	16	18	20	22	24	26	28	30
유클리디언 거리 기반 알고리즘	Outlier 개수	2701	2701	2366	2701	359	719	320	318	2701	238	320
	Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Recall	0.82	0.82	0.72	0.82	0.11	0.22	0.10	0.10	0.82	0.07	0.10
	F1값	0.90	0.90	0.84	0.90	0.20	0.36	0.18	0.18	0.90	0.14	0.18
밀도함수기반 알고리즘 (제안알고리즘)	outlier 개수	2701	2701	2366	2701	1351	1982	724	727	2701	1652	724
	Precision	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Recall	0.82	0.82	0.72	0.82	0.41	0.60	0.22	0.22	0.82	0.22	0.22
	F1값	0.90	0.90	0.84	0.90	0.58	0.75	0.36	0.36	0.90	0.36	0.36

시하였다. 그리고 테스트 데이터를 대상으로 아웃라이어 클러스터의 검출 성능을 비교 평가하기 위해서, 본 논문에서 제안한 클러스터 밀도 함수와 유클리디언 거리 함수를 이용한 성능을 비교하였다. 본 실험의 목적은 K-means 클러스터링 알고리즘을 기반으로, 밀도함수 기반 기법이 유클리디언 거리함수 기반 기법에 비해 아웃라이어 클러스터 탐지 효과가 얼마나 향상되는지를 확인하는데 있다.

표 1과 2는 두 기법을 테스트한 결과를 나타낸 것이다. 표 1은 침입 패킷이 적은 경우에 해당하고, 표 2는 침입 패킷이 많은 경우에 해당한다. 우선 정확도의 측면에서 두 기법 모두가 클러스터의 개수에 따라 변함없이 1.0의 값을 나타낸다. 이는 탐지 알고리즘에 의해 선정된 아웃라이어 클러스터 내부에 포함된 모든 인스턴스 데이터는 침입 패킷임을 의미한다. 이는 본 논문에서 침입 패킷을 찾기 위한 목적으로 '아웃라이어 클러스터'를 탐색하는 아이디어가 주효함을 증명하는 결과이다. 반면에 재현율의 경우에 0과 1사이의 값을 가지면서 클러스터의 개수에 따라 그 값의 변화가 크다. 결국 본 실험에서 결과적으로 측정하고자 하는 F1-측정치는 재현율에 의해 결정되며, 제안한 기법의 관건은 실험 데이터에 존재하는 침입 패킷을 얼마나 많이 아웃라이어 클러스터에 포함시킬 수 있느냐 하는 것이다. 클러스터 개수가 18개 이상인 경우, 유클리디언 거리함수 기반 기법에 비해 재현율 및 F1-측정치의 값이 높게 나타나고 있다. 대신 유클리디언 거리함수 기반 기법이 최대값을 가지는 경우, 두 기법이 같은 성능을 보이고 있다. 앞서 기술한 바와 같이 밀도 함수 기반 기법은 서로 가까이 존재하는 주변 클러스터와의 영향력을 근거로 아웃라이어 클러스터를 판별한다. 그러므로 클러스터 개수가 많을수록 그들 사이의 영향력이 민감하게 반응하여 아웃라이어 클러스터의 판별력이 높아지는 것으로 분석한다. 그리고 실제로 네트워크 침입 패턴들은 그 크기가 크거나 밀집된 형태의 아웃라이어 클러스터를 구성할 수 있으며, 유클리디언 거리함수로는 그러한 형태의 아웃라이어 클러스터를 탐지하기 어렵다.

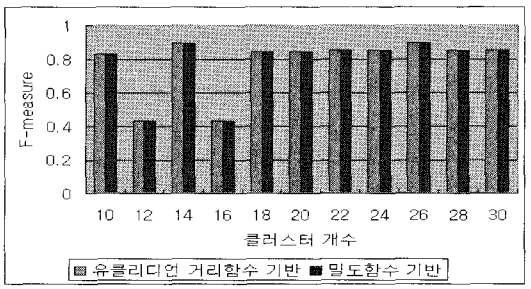


그림 2. 침입패킷 비율이 1%일 경우의 F-측정치

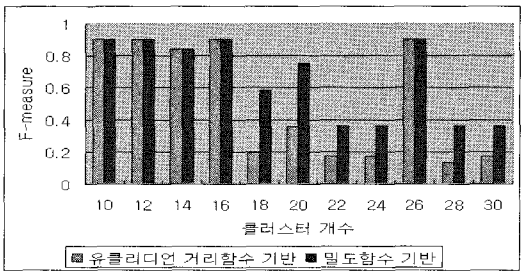


그림 3. 침입패킷 비율이 23%일 경우의 F-측정치

그림 2와 3은 두 기법의 성능을 쉽게 파악할 수 있도록 생성되는 클러스터의 수에 따라서 단일 측정값인 F1-측정치를 보여준다. 이 그림에서 보는 바와 같이, 침입 패킷의 비율이 낮은 상황(그림 2)에서는 기존의 방법과 거의 동일한 성능을 보여주고 있지만, 침입이 빈번히 발생되는 상황(그림 3)에서는 본 논문에서 제안한 밀도 함수를 이용한 방법이 기존의 유클리디언 거리 함수를 이용한 기존의 방법에 비해 좋은 성능을 보여주고 있다.

5. 결론

본 논문은 실시간 네트워크 침입 탐지 시스템에 적용하기 위해 밀도 함수를 이용한 새로운 아웃라이어 클러스터 검출 기법을 제안하였다. 기존 유클리디언 거리함수와 데이터포인트 수량 기반 밀도함수를 이용한 아웃라이어 탐지는 단일 인스턴스로서의 네트워크 침입을 판별하는 것이지만, 제안한 방법은 네트워크 침입이 일반적으로 하나의 인스턴트가 아닌 유사한 인스턴트의 집합 형태로 나타난다는 점에 착안하여, 아웃라이어 인스턴스들의 군집인 '아웃라이어 클러스터'를 검출하는 기법을 제안한 것이다. 이러한 아웃라이어 클러스터를 검출하기 위해 특정 클러스터의 거리뿐만 아니라 주변 클러스터들로부터의 영향력 정도를 고려한 밀도 함수를 정의하였다. 이 함수는 침입이 자주 발생하는 상황에서 그 효과가 큼을 실험을 통하여 증명하였다. 향후에는 본 논문에서 제안된 방법에 의해 탐색된 아웃라이어 클러스터로부터 침입 시그너처를 자동 생성하는 연구를 진행할 것이며, 이는 의사결정트리(Decision Trees) 학습알고리즘 또는 퍼지집합 이론을 활용하여 아웃라이어 클러스터로부터 주요 특성을 자동 추출하는 과정을 포함할 수 있다[19]. 또한 본 논문의 실험결과를 토대로 침입탐지율과 크게 상관성을 가지는 클러스터 질의 측정치를 제시함과 더불어, 그 측정치를 이용하여 클러스터 개수를 동적으로 결정하는 기법을 연구할 것이다.

참 고 문 헌

[1] S. Zhong, T.M. Khoshgoftaar, "A Clustering Approach to Wireless Network Intrusion Detection", *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, pp. 190-196, 2005.

[2] A. Lazarevic, A. Ozgur, L. Ertoz, J. Srivastava, and V. Kumar, "A comparative study of anomaly detection schemes in network intrusion detection", *Proceedings of SIAM International Conference on Data Minig*, 2003.

[3] W. Lee, S.J. Stolfo, K.W. Mok, "A data mining framework for building intrusion detection models", *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, 1999.

[4] G. Vigna and R.A. Kemmerer, "NetSTAT: A Network-based Intrusion Detection Approach", *Proceedings of the 14th Annual Computer Security Conference*, 1998.

[5] S. Chris, P. Lyn and M. Sara, "An Application of Machine Learning to Network Intrusion Detection", *Proceedings of the 54th Annual Computer Security applications Conference*, p.371-377, 1999.

[6] C. Krugel, and T. Toth, "Using Decision Trees to Improve Signature-Based Intrusion Detection", *Proceedings of Recent Advances in Intrusion Detection*, pp.173-191, 2003.

[7] J. Mill, and A. Inoue, "Support Vector Classifiers and. Network Intrusion Detection", *Proceedings of FUZZ-IEEE 2004*, Budapest, Hungary, pp. 1-211, 2004.

[8] C. Kruegel and G. Vigna. Anomaly detection of web-based attacks, *ACM Proceedings of the*

10th ACM conference on Computer and communications security, pp. 251-261, 2003.

[9] J. Oldmeadow, S. Ravinutala, and C. Leckie, "Adaptive Clustering for Network Intrusion Detection", *Lecture Notes in Computer Science*, Vol. 3056, pp. 25-259, 2004.

[10] M. I. Petrovskiy, "Outlier Detection Algorithms in Data Mining Systems", *Programming and Computing Software*, Vol.29, No.4, pp.228-237, 2003.

[11] S. Zhong, T. M. Khoshgoftaar, and N. Seliya, "Evaluating Clustering Techniques for Network Intrusion Detection", *Proceedings of the 10th ISSAT International Conference on Reliability and Quality Design*, pp. 149-155, 2004.

[12] MIT Lincoln Labs, DARPA intrusion detection evaluation. 1999, available in http://www.ll.mit.edu/IST/ideval/index.html.

[13] Witten I.H. and Frank E, "Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations", Morgan Kaufmann, 2000.

[14] Victoria Hodge, Jim Austin, "A Survey of Outlier Detection Methodologies", *Artificial Intelligence Review*, Vol.22, No.2, pp.85-126, 2004.

[15] M. Breunig, H. Kriegel, R.T. Ng, J. Sander, "LOF: Identifying Density-Based Local Outliers", *Proceedings of ACM SIGMOD International Conference on Management of Data*, pp.93-104, 2000.

[16] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

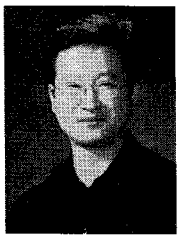
[17] A. Hinneburg and D. A. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise", *Proceedings of the 4th International Conference of Knowledge Discovery and Data Mining*, pp.58-65, 1998.

[18] J. Han and M. Kamber, "Cluster Analysis", in *Data Mining: Concepts and Techniques*, Morgan Kaufmann Publishers, pp. 335-394, 2000.

[19] E. Leon, O. Nasraoui, J. Gomez, "Anomaly Detection Based on Unsupervised Niche Clustering with Application to Network Intrusion", *Evolutionary Computation 2004 (CEC2004)*, pp. 502-508, 2004.

◐ 저 자 소 개 ◑

장 재 영(Jae-young Chang)
1992년 서울대학교 계산통계학과 졸업(학사)
1994년 서울대학교 대학원 계산통계학과 졸업(석사)
1999년 서울대학교 대학원 계산통계학과 전산과학전공 졸업(박사)
2000~현재 한성대학교 컴퓨터공학과 부교수
관심분야 : 데이터베이스, 데이터마이닝
E-mail : jychang@hansung.ac.kr



김 한 준(Han-joon Kim)
1994년 서울대학교 계산통계학과 졸업(학사)
1996년 서울대학교 대학원 전산과학과 졸업(석사)
2002년 서울대학교 대학원 컴퓨터공학과 졸업(박사)
2002~현재 서울시립대학교 전자전기컴퓨터공학부 조교수
관심분야 : 데이터베이스, 데이터마이닝, 정보검색
E-mail : khj@uos.ac.kr

박 종 명(Jongmyoung Park)
2004년 서울시립대학교 전자전기공학부 졸업(학사)
2006년 서울시립대학교 대학원 전자전기컴퓨터공학부 졸업(석사)
2006~현재 주식회사 위트콤 사원
관심분야 : 데이터베이스, 정보보호
E-mail : dustjm@witcom.co.kr