침입 탐지를 위한 효율적인 퍼지 분류 규칙 생성
(Generation of Efficient Fuzzy Classification Rules for Intrusion Detection)

김 성 은 † 길 아 라 †† 김 명 원 ††
(Sung Eun Kim) (Ara Khil) (Myung Won Kim)

요 약 본 논문에서는 효율적인 침입 탐지를 위해 퍼지 규칙을 이용하는 방법을 제안한다. 제안한 방법은 퍼지 의사결정 트리의 생성을 통해 침입 탐지를 위한 퍼지 규칙을 생성하고 진화 알고리즘을 사용하여 최적화한다. 진화 알고리즘의 효율적인 수행을 위해 지도 군집화를 사용하여 퍼지 규칙을 위한 초기 소속함수를 생성한다. 제안한 방법의 진화 알고리즘은 적합도 평가시 퍼지 규칙(퍼지 의사결정 트리)의 성능과 복잡성을 고려하여 평가한다. 또한 데이타 분할을 이용한 평가와 퍼지 의사결정 트리의 생성과 평가 시간을 줄이는 방법으로 소속정도 캐싱과 zero-pruning을 사용한다. 제안한 방법의 성능 평가를 위해 KDD'99 Cup의 침입 탐지 데이타로 실험하여 기존 방법보다 성능이 향상된 것을 확인하였다. 특히, KDD'99 Cup 우승자에 비해 정확도가 1.54% 향상되고 탐지 비용은 20.8% 절감되었다.

키워드 : 침입 탐지, 퍼지 분류 규칙, 진화알고리즘, 분할 진화 방법, 지도 군집화

Abstract In this paper, we investigate the use of fuzzy rules for efficient intrusion detection. We use evolutionary algorithm to optimize the set of fuzzy rules for intrusion detection by constructing fuzzy decision trees. For efficient execution of evolutionary algorithm we use supervised clustering to generate an initial set of membership functions for fuzzy rules. In our method both performance and complexity of fuzzy rules (or fuzzy decision trees) are taken into account in fitness evaluation. We also use evaluation with data partition, membership degree caching and zero-pruning to reduce time for construction and evaluation of fuzzy decision trees.

For performance evaluation, we experimented with our method over the intrusion detection data of KDD'99 Cup, and confirmed that our method outperformed the existing methods. Compared with the KDD'99 Cup winner, the accuracy was increased by 1.54% while the cost was reduced by 20.8%.

Key words : intrusion detection, fuzzy classification rule, evolutionary algorithm, partition evolutionary method, supervised clustering

1. 서 론

최근 인터넷 서비스 사용자가 급증하고 전자상거래와 인터넷 공공서비스 등과 같은 네트워크 서비스가 보편화 되면서 컴퓨터 시스템에 대한 침입 유형이 다양해지고 침입에 노출될 가능성이 증가하고 있다. 그에 따라 개인정보, 기업정보를 불법적으로 획득하여 오용하는 사례가 발생하고 있다. 실제로 침입에 대한 피해 사례는 2002년 이후 해마다 이전 년도 대비 7%씩 증가하고 있다[1]. 따라서 정보자산의 보호를 위한 보안 시스템이 필요하고 침입에 대하여 사전에 경보하고 대응할 수 있는 침입 탐지 시스템(intrusion detection system)에 대한 요구가 급증하고 있다.

기존에 개발된 침입 탐지 시스템은 탐지 방법에 따라 오용탐지(misuse detection)와 비정상 행위 탐지(anomaly detection)로 나뉜다. 오용탐지의 경우 일반적으로 많이 사용하는 방법으로 공격의 특징들을 기술한 시그너쳐(signature)를 사용하여 네트워크의 패킷 중 시그너쳐와 비슷한 형태의 패턴이 발생하면 침입을 알리는 탐지방법이다. 하지만 사전에 모든 발생 가능한 공격 유형에 대하여 해당 분야의 전문가에 의해 시그너쳐를 정의해야 하고 새로운 공격 유형이 발생할 때마다 시그너쳐를 업데이트해야 하는 문제가 있다.

· 본 연구는 한국학술진흥재단 선도연구자지원사업에 의해 수행되었습니다. (과제번호: KRF-2005-041-D00707)
† 정 회 원 : (주)퓨쳐시스템 정보통신연구소 주임연구원
babystep@naver.com
†† 종신회원 : 숭실대학교 컴퓨터학부 교수
ara@comp.ssu.ac.kr
mkim@comp.ssu.ac.kr
논문접수 : 2006년 8월 16일
심사완료 : 2007년 4월 10일

이에 비해 비정상 행위 탐지는 평소에 발생하는 정상적인 데이타를 분석하여 모델을 생성하고 모델의 기준과 상대적으로 급격한 변화를 일으켜서 임계치(threshold) 이상 차이가 나는 경우 침입으로 통보한다. 전문가에 의한 오용탐지에 비해 유지보수 비용이 적고 비교적 새로운 공격 유형에 대한 탐지도 할 수 있다. 하지만 적절한 임계치를 결정하기가 어렵고 오탐지(false alarm)의 경우가 많다.

이와 같은 기존의 침입 탐지의 문제를 해결하기 위해 인공지능 분야에서는 데이타 마이닝의 분류(classification) 기법을 적용한 연구가 진행되었다[2-5]. 분류 기법은 생성되는 규칙의 형태에 따라 일반 규칙과 퍼지 규칙으로 나뉜다. 일반 규칙을 사용하는 침입 탐지 시스템은 퍼지 규칙에 비해 탐지 속도는 빠르지만 정상적인 사용자의 접속에서 정상적인 패턴에도 불구하고 침입 경고를 통보하는 경우와 침입을 탐지하지 못하여 침입에 노출되는 경우가 많다[3]. 이러한 문제점을 개선하기 위해 일반 규칙 보다 간결한 규칙으로 데이타의 모호성을 표현하고 근사 추론을 통해 처리할 수 있는 특성을 갖는 퍼지 규칙을 사용한 방법들이 연구되고 있다[3-5].

본 논문에서는 침입 탐지를 위한 퍼지 규칙을 자동으로 생성하는 방법을 제안한다. 첫 번째로 네트워크에서 추출된 패킷 데이타를 지도 군집화(supervised clustering)를 사용하여 침입 유형별 분포를 고려해 퍼지 소속함수를 자동으로 생성한다. 이때 최적의 군집 개수를 결정하기 위해 진화 알고리즘을 사용한다. 두 번째로 정확하고 간결성이 높은 침입 탐지 규칙 생성을 위해 진화 알고리즘을 사용하여 규칙 최적화를 수행한다. 최적화 알고리즘인 진화 알고리즘의 수행 시간이 오래 걸리는 단점을 개선하기 위해 데이타 분할 평가 방법을 사용한다. 그리고 침입 탐지 데이타가 대용량임을 고려하여 소속함수 최적화 진화 알고리즘의 개체 평가시 사용하는 퍼지 의사결정 트리의 생성과 평가 시간을 줄이는 방법인 소속정도 캐싱과 zero-pruning을 사용한다.

제안한 방법을 이용한 침입 탐지 시스템은 침입 유형별 분포를 고려하여 침입 탐지에 보다 효과적인 초기 소속함수를 생성한다. 또, 최적화를 통해 정확하고 간결성이 높은 탐지 규칙을 생성함으로써 오탐지와 미탐지(miss detection)를 줄일 수 있다. 그 결과 네트워크 서버 관리자에게 정상적인 접속에 대한 잘못된 침입 경보의 횟수가 감소하여 시스템 관리 비용을 줄일 수 있으며, 침입에 노출되는 횟수를 줄임으로써 침입에 대한 피해로 발생되는 비용을 감소시킨다. 효율적이고 정확한 침입 탐지 시스템을 구축하여 침입으로부터 사용자 또는 기업의 자원과 정보를 보호할 수 있다.

본 논문의 구성은 다음과 같다. 2절에서 본 논문의 기본 개념인 퍼지 규칙, 퍼지 의사결정 트리 및 진화알고리즘에 대하여 간략히 설명하고 3절에서는 침입 탐지 시스템을 위한 퍼지 분류 규칙을 생성하는 관련 연구를 소개한다. 4절에서는 제안한 침입 탐지를 위한 퍼지 분류 규칙의 생성 방법에 대하여 설명하고 또 진화 알고리즘의 계산 시간을 줄이기 위한 분할 평가 방법, zero-pruning과 소속정도 캐싱에 대하여 설명한다. 5절에서는 제안한 방법의 타당성 검증을 위해 KDD'99 침입 탐지 데이타를 사용하여 기존 방법과 성능을 비교한 실험에 대하여 기술한다. 마지막으로 6절에서는 결론 및 향후 연구를 제시한다.

2. 기본 개념: 퍼지 규칙, 퍼지 의사결정 트리, 진화 알고리즘

본 논문에서 사용하는 퍼지 규칙은 식 (1)과 같이 여러 개의 조건항이 논리적AND (logical conjunction)로 연결된 형태의 조건절 (IF 부분)과 클래스 분류를 나타내는 결론절(THEN 부분)을 갖고 해당 규칙에 대한 확신도(CF : Certainty Factor)로 구성되어 있다. 식 (1)에서 A_k는 데이타의 속성을 나타내며 U_k는 속성 A_k의 값을 나타내는 언어항으로서 퍼지집합의 소속함수로 표현된다. 본 논문에서는 퍼지 규칙의 추론 방법으로 Mamdani의 min-max 방법을 사용한다[10]. 즉, 각 조건항의 만족도를 해당 조건항의 속성 값에 대응하는 퍼지 집합의 소속정도로 하고 각 조건항들의 만족도의 최소값을 조건절의 만족도로 하고 이 값을 해당 규칙의 확신도와 곱한 값을 결론의 신뢰도로 하여 결론을 도출한다.

$$\text{규칙 : } IF\ A_1\ is\ U_1\ AND\ A_2\ is\ U_2 \cdots \quad (1)$$
$$AND\ A_m\ is\ U_m\ THEN\ Class\ is\ k\ (CF)$$

예를 들면, '*IF* 접속시간 is 짧다 *AND* 패킷 용량 is 크다 *THEN* Class is 침입 (0.9)'에서 짧다, 크다 등과 같은 퍼지 언어항은 퍼지집합으로 표현된다. 실제로 추론할 경우 조건항 '접속시간 is 짧다'에 대하여 실제 접속시간이 주어지면 그 값이 '짧다'에 대응하는 퍼지집합의 소속정도가 해당 조건항의 만족도가 된다. 조건절에 속한 모든 조건항의 만족도의 최소값을 조건절의 만족도로 한다. 예를 들면, '접속시간 is 짧다'와 '패킷 용량 is 크다'에 대한 만족도를 각각 0.8, 0.9라 하면 두 수의 최소값 0.8이 조건절의 만족도가 되고 최종적으로 '침입'의 결론을 도출하는데 그 신뢰도는 0.8×0.9=0.72가 된다. 일반적으로 동일한 데이타에 대하여 여러 개의 퍼지 규칙이 동시에 적용될 수 있는데, 이 경우 각 규칙으로부터 도출된 결론 중 최대의 신뢰도를 갖는 결론을 최종 결론으로 취하게 된다.

소속함수는 퍼지 규칙에서 매우 중요한 요소이다. 소

속함수는 퍼지 규칙의 성능과 이해성에 영향을 준다. 일반적으로 소속함수는 삼각형, 사다리꼴, 가우시안 함수 형태의 소속함수가 많이 사용된다. 본 논문에서는 그림 1과 같이 삼각형 소속함수를 사용한다. 삼각형 소속함수는 (l, c, r)로 표현하며 여기서 l, c, r은 그림 1에서와 같이 삼각형 소속함수의 왼쪽, 가운데, 오른쪽 꼭지점의 x좌표를 각각 나타낸다. 또 $\mu_A(x)$는 x의 퍼지집합 A에 대한 소속정도를 나타내는 함수이다.

본 논문에서는 최대한 간결한 퍼지규칙을 생성하기 위하여 퍼지 의사결정 트리를 먼저 생성하고 이로부터 퍼지규칙을 추출한다. 퍼지 의사결정 트리는 중간 노드, 단말노드, 링크로 구성되어 있다. 그림 2에서 '접속시간', '패킷용량', '점유파일 개수'는 중간노드이고 '정상', '침입'은 단말 노드이며 '짧다', '보통', '길다'는 링크를 나타낸다. 중간노드는 분할을 위한 속성을 가지고 있으며, 단말노드는 클래스와 CF를 가지고 있다. 노드와 노드 사이를 연결하는 링크는 속성에 대한 소속함수를 나타낸다. 그림 2는 '접속시간', '패킷용량', '점유파일 개수'와 같은 세 가지의 연속형 속성들을 분석하여 컴퓨터 시스템의 침입 여부를 예측하는 퍼지 의사결정 트리를 예시한 것이다. 그림에서와 같이 루트(root)에서부터 단말노드까지의 경로가 한 개의 퍼지 규칙에 해당된다.

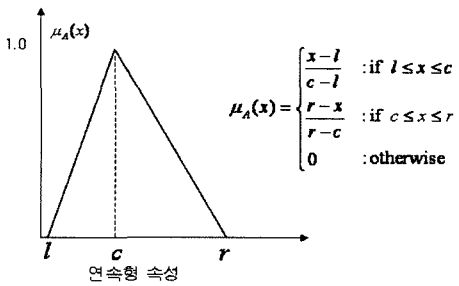


그림 1 삼각형 퍼지 소속함수 예

진화 알고리즘은 자연계의 생물의 진화 메커니즘을 모방한 확률적 탐색 알고리즘으로서 주로 최적화 문제에 많이 응용되고 있다. 주어진 문제의 잠재적 해를 개체 (individual)로 인코딩하여 초기 개체 집단(세대)을 만든다. 그 다음 각 개체가 주어진 문제의 제약 조건을 얼마나 잘 만족하는가 또는 목적함수를 평가하여 각 개체의 적합도(fitness score)를 계산한다. 적합도가 상대적으로 우수한 개체들을 선택(재생산)한 후 교배(cross-over), 돌연변이(mutation) 등의 진화 연산을 적용하여 새로운 개체를 생성하고 이 새로운 개체들이 새로운 세대를 이룬다. 이와 같은 개체의 평가, 진화 연산 적용 등의 과정을 종료 조건이 만족될 때까지 반복적으로 수행함으로써 개체를 진화시켜서 문제에 대한 최적의 해를 구하게 된다. 그림 3은 진화 알고리즘을 나타내며 그림에서 P_t는 t 세대의 개체 집단을 나타낸다.

```
procedure EA
  begin
    t = 0
    초기화(P_t)
    while not 종료조건 do
      begin
        평가(P_t)
        t = t+1
        P_t = 재생산(P_t-1)
        교배(P_t)
        돌연변이(P_t)
      end
  end
```

그림 3 진화 알고리즘

3. 관련 연구

본 절에서는 침입 탐지 문제를 해결하기 위해 퍼지

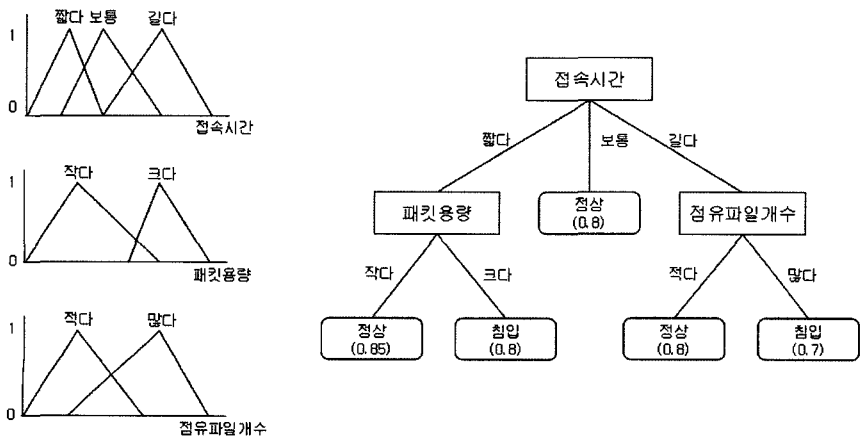


그림 2 퍼지 의사결정 트리의 예

분류 규칙을 생성하는 연구로 소속함수를 사전에 정하고 퍼지 규칙을 생성하는 방법인 EFRID와 소속함수를 자동으로 생성하여 최적화하는 방법인 MOGFIDS를 살펴본다.

3.1 EFRID

Jonatan Gomez[3]는 정상일 때와 침입일 때, 각각에 대해 진화 알고리즘을 사용하여 침입 탐지 규칙을 생성하는 EFRID(Evolving Fuzzy Rules for Intrusion Detection)를 제안하였다. 두 개의 진화 알고리즘은 각각 정상일 때의 규칙과 침입일 때의 규칙을 생성한다. 규칙은 조건절(condition part)과 정상 또는 침입으로 분류하기 위한 결론절(consequent part), 그리고 규칙 신뢰도(confidence) 항목으로 구성된다. 이때 조건절은 데이타를 분류하기 위한 조건들을 포함하고, 결론절은 진화 알고리즘의 목적에 따라 '정상' 또는 '침입'을 갖는다. 규칙에서 결론절과 규칙 신뢰도 항목을 제외하고 조건절만 진화 알고리즘의 개체에 인코딩한다. 조건절을 진화 알고리즘의 개체로 인코딩하기 위해 완전 수식 트리(complete expression tree)를 사용하여 괄호가 제거된 형식으로 변환한 후 인코딩한다[4]. 진화연산은 일반적인 진화 알고리즘의 교배(crossover)와 돌연변이(mutation) 연산을 사용하지 않고 추가와 삭제 연산을 사용한다[4]. 진화 알고리즘을 사용함으로써 최적화된 규칙을 생성할 수 있는 장점이 있다. 하지만 퍼지 규칙을 사용하기 위해 사전에 사용자에 의해 퍼지 소속함수를 정해야 하는 제약이 있다. EFRID에서는 0에서 1사이의 범위로 정규화된 영역을 다섯 구간으로 퍼지화한 표준 삼각형 소속함수 집합을 사용한다.

퍼지 소속함수의 경우 표현하고자 하는 문제의 배경지식을 잘 표현해야 정확도가 높은 퍼지 분류 규칙을 생성할 수 있다. 하지만 EFRID에서는 고정된 표준 소속함수를 사용하고 진화 알고리즘을 이용하여 퍼지 규칙만을 최적화한다. 따라서 좀 더 효과적인 퍼지 분류 규칙 생성을 위해 퍼지 소속함수를 최적화하여 정확도 높은 퍼지 분류 규칙을 생성하는 방법이 필요하다.

3.2 MOGFIDS

Chi-Ho Tsang[5]은 퍼지 소속함수의 집합을 갖는 퍼지 집합 에이전트들을 사용하여 침입 탐지에 적합한 퍼지 규칙을 생성하는 MOGFIDS(Multi-Objective Genetic Fuzzy Intrusion Detection System)를 제안하였다. MOGFIDS는 다수의 퍼지 집합 에이전트(fuzzy set agent)와 한 개의 중재자 에이전트(arbitrator agent)로 구성된 에이전트 기반의 진화 프레임워크이다. 각각의 퍼지 집합 에이전트들은 세 가지의 주된 전략을 갖고 그것을 바탕으로 퍼지 규칙을 생성하고 진화하는 과정을 자동으로 수행한다.

각 퍼지 집합 에이전트는 퍼지 집합 분배 전략(fuzzy sets distribution strategy)을 사용하여 초기화한다. 초기화된 퍼지 집합 에이전트들은 이해성 기반의 정형화 전략(interpretability-based regulation strategy)과 퍼지규칙 생성 전략(the generation strategy of fuzzy rules)을 통하여 이해성이 높은 퍼지규칙을 생성한다. 퍼지 집합 에이전트들은 계층적 구조를 갖는 염색체의 형태로 인코딩 되고, 매 세대마다 교배와 돌연변이 연산을 적용하여 퍼지 집합 에이전트들이 갖는 퍼지 집합의 정보를 협력적으로 교환함으로써 다음 세대의 자식 에이전트를 생성한다. 이와 같은 방법으로 퍼지 집합 에이전트들은 매 세대마다 퍼지 집합의 정보를 교환하면서 진화과정을 수행한다.

각각의 퍼지 집합 에이전트들은 생성한 퍼지규칙에 대해 정확성과 이해성을 기준으로 평가하고 자신의 적합도(fitness value)를 중재자 에이전트에게 전달한다. 중재자 에이전트는 매 세대의 평가 결과를 사용하여 적합도가 높은 에이전트는 유지하고 반면에 적합도가 낮은 에이전트는 제거한다. 이와 같은 과정을 수행하여 최종적으로 침입 탐지에 최적화된 퍼지규칙을 생성한다.

에이전트 기반 진화 프레임워크를 사용하여 침입 탐지에 적합한 퍼지 소속함수를 자동으로 생성하고, 진화를 통해 퍼지 규칙을 생성함으로써 정확도와 이해성이 높은 규칙을 생성하는 장점이 있다. 하지만 진화 프레임워크에서 퍼지 소속함수를 갖는 초기 퍼지 집합 에이전트를 생성할 때 별도의 사전 정보를 사용하지 않고 임의로 퍼지 소속함수를 생성함으로써 수렴하는 해의 최적성이 떨어지거나 최적의 해에 수렴하는 시간이 증가한다. 따라서 사전 정보를 활용하여 효과적으로 초기 소속함수를 생성하는 방법이 필요하다.

4. 제안한 침입 탐지를 위한 퍼지 분류 규칙 생성 방법

본 논문은 네트워크에서 수집된 트래픽 데이타를 사용하여 불법 사용자의 침입을 탐지할 수 있는 퍼지 분류 규칙을 생성하고 이를 이용한 침입 탐지 방법을 제안한다. 그림 4는 제안한 침입 탐지 규칙의 생성을 위한 프로세스를 도식화한 것이다.

제안한 방법은 진화 알고리즘의 평가시간 단축을 위해 그림 4와 같이 전체데이타를 n개의 부분 학습데이타로 분할하는 단계와 클래스 분포를 고려한 초기 소속함수 생성 단계[6], 정확도 높은 침입 탐지 규칙 생성을 위한 소속함수 최적화 단계[7]로 구성되어 있다.

4.1 초기 소속함수 생성을 위한 군집화

지도 군집화는 학습 데이타의 클래스를 사용하여 군집화 하는 방법이다. 지도 군집화는 클래스를 이용하여 군

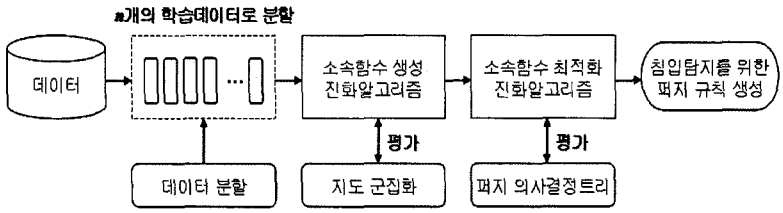


그림 4 침입 탐지 규칙 생성 프로세스

집하므로 침입과 정상 클래스의 분포를 고려하여 군집화한다. 지도 군집화는 k-means와 수행 과정은 같고 단지, 군집 중심에 클래스가 명시되고 데이타를 가장 가까운 군집에 할당할 때 데이타의 클래스와 군집 중심의 클래스가 같은 경우에만 유사도를 계산하여 할당한다.

본 논문에서는 지도 군집화의 군집 개수를 결정하기 위해 진화 알고리즘을 사용한다. 지도 군집화를 위한 진화 알고리즘에서 개체는 실수형 가변 길이로 인코딩한다. 유전인자는 군집의 중심 좌표를 나타내고 유전인자 개수는 군집의 개수를 나타낸다. 다음의 그림 5는 지도 군집화를 위한 진화 알고리즘에서 개체를 인코딩한 예이다.

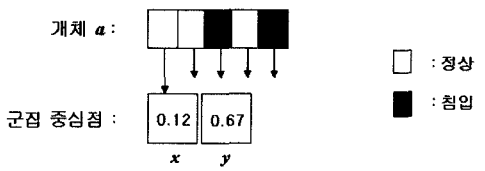


그림 5 소속함수 생성 진화 알고리즘의 개체 인코딩

그림 5는 차원이 2차원(x, y)이고 클래스가 2종류('정상', '침입')인 학습 데이타를 사용하는 경우의 개체 인코딩 예이다. 이 경우 개체의 길이는 5이고 각 유전인자는 군집의 중심점의 값을 갖는다. 초기 개체 집단을 생성할 때 개체의 길이는 $[|CLASS|, 2\times|CLASS|]$ 범위에서 임의로 생성하고 유전인자는 클래스 별로 데이타에서 임의로 선택한다. 여기서 $CLASS$는 클래스 집합을 나타낸다.

개체 평가 단계에서 개체에 표현된 군집의 정보를 가지고 지도 군집화를 수행한다. 생성된 군집화 결과를 이용하여 개체의 적합도를 식 (2)의 함수를 사용하여 계산한다. 적합도 함수는 군집의 동질성(homogeneity)과 군집화 복잡도(clustering complexity) 항목으로 나타낸다. 동질성($Hom(C)$)은 식 (3)과 같이 군집의 평균 순도로 계산하고 군집의 순도는 식 (4)와 같이 군집 중심의 클래스와 동일한 클래스의 데이타들의 비율로 나타낸다.

$$CFit(C) = vHom(C) - (1-v)Ccpx(C),\ 0 \le v \le 1 \quad (2)$$

$$Hom(C) = \frac{\sum_{i=1}^{|C|} Pur(c_i)}{|C|},\ c_i \in C \quad (3)$$

$$Pur(c_i) = \frac{\sum_{m \in N^{k_i}} \mu_{im}}{\sum_{l \in N} \mu_{il}} \quad (4)$$

$$Ccpx(C) = \frac{|C|}{|CLASS|} \quad (5)$$

또 군집화 복잡도(Ccpx)는 군집의 개수를 클래스의 개수로 나눈 것으로 군집이 얼마나 많이 생성되는가를 나타내며 군집의 개수가 초기 소속함수의 개수를 결정하므로 가능한 한 적은 군집 개수로 최대의 동질성을 얻기 위해 식 (2)에서 군집화율은 패널티로서 작용한다. N, C는 각각 전체 데이타 집합, 생성된 군집의 집합을 나타낸다. k_i는 군집 i의 중심 c_i의 클래스이고, N^{k_i}는 k_i와 동일한 클래스를 갖는 데이타 집합이다. μ_{il}는 데이타 l이 군집 i에 속하는 소속정도를 의미한다. v는 적합도 함수에서 동질성과 군집화율의 비중을 제어하는 가중치이다.

자식 개체를 생성할 때, 부모 개체에게 사용되는 진화 연산은 클래스별 한 점 교배 연산, 가우시안 돌연변이 연산을 사용한다[8]. 클래스별 한 점 교배 연산은 두 개의 클래스가 있을 때 각각 클래스별로 임의의 한 점을 선택하고 선택된 점을 기준으로 교배한다.

위와 같이 생성된 군집을 각 차원에 투영하여 삼각형 소속함수를 생성하는데 이 때 군집의 중심점이 삼각형 소속함수의 가운데 점(소속정도 1.0)이 되게 하고 좌우 끝점은 랜덤하게 정하되 이웃하는 소속함수의 경우 중간 부분이 서로 겹치게 끝점을 생성한다. 이 경우 각 차원에서 생성될 수 있는 소속함수의 최대 개수는 생성된 군집의 개수가 된다.

4.2 소속함수 최적화

클래스 분포를 고려하여 생성된 초기 소속함수를 이용하여 퍼지 의사결정 트리[9]를 생성한다. 생성된 퍼지 의사결정 트리로부터 퍼지 분류 규칙을 생성한다. 이때 정확도와 간결성이 높은 퍼지 분류 규칙을 생성하기 위해 진화 알고리즘을 사용하여 최적화를 수행한다.

소속함수의 최적화를 위한 진화 알고리즘의 초기 개체 집단에서 모든 개체의 길이는 클래스 분포를 고려하여 생성된 각 차원의 소속함수 개수의 합과 같다. 그리고 개체 유전인자인 삼각형 소속함수에서 소속정도가 1.0인 중앙값도 클래스 분포를 고려하여 생성된 좌표값과 같다. 소속함수 생성 단계에서 지도 군집화를 사용하여 생성된 소속함수를 초기 개체 집단으로 설정한다. 개체의 길이는 데이타의 속성 개수와 같다. 각각의 유전인자는 해당하는 속성을 표현하는 퍼지 소속함수의 집합을 나타내고 각 속성의 소속함수 집합 내에는 삼각형 소속함수의 좌표를 나타내는 세 개의 실수값으로 된 triple을 포함하고 있다. 그림 6은 소속함수 최적화 진화 알고리즘에서 사용한 개체 인코딩 예이다.

개체 x는 속성이 5개인 데이타를 나타내는 소속함수를 인코딩한 예이다. 각각의 유전인자는 해당 속성의 소속함수 집합을 갖고 소속함수 집합의 각 원소는 3개의 실수 좌표값(l, c, r)을 포함한다.

개체 집단에 표현된 소속함수를 사용하여 퍼지 의사결정트리를 생성하여 트리로부터 규칙을 추출한다. 추출된 규칙을 사용하여 개체 평가 과정을 수행한다. 해당 개체의 적합도를 계산하기 위한 적합도 함수는 식 (6)과 같다. w는 적합도 함수에서 정확성과 간결성의 비중을 나타내는 가중치이다.

$$TFit_i = wAcc(\tau_i) - (1-w)Tcpx(\tau_i),\ 0 \le w \le 1 \quad (6)$$

$$Tcpx(\tau_i) = \frac{R(\tau_i)}{|CLASS|} \quad (7)$$

τ_i는 개체 i에 대응하는 퍼지 의사결정 트리를 의미하고, $Acc(\tau_i)$는 개체 i에 의해 생성된 퍼지 의사결정 트리의 정확도(accuracy)를 나타낸다. $Tcpx(\tau_i)$는 퍼지 의사결정 트리 τ_i의 복잡도(complexity)를 나타내고 식 (7)을 사용하여 계산한다. $R(\tau_i)$는 개체 i의 퍼지 의사결정 트리 τ_i로부터 생성된 규칙의 개수를 나타내고 $CLASS$는 클래스의 집합을 나타낸다.

가변길이 실수 표현으로 인코딩한 개체에 대해 진화 연산은 가우시안 돌연변이 연산과 전체 산술 교배(whole arithmetic crossover), 휴리스틱 교배(heuristic crossover) 그리고 속성 교체 교배(attribute change crossover)로 정의한다[8]. 각 진화 연산은 개체 내에 표현된 소속함수를 조정하고 진화 과정을 반복하면서 소최적의 성능을 갖는 탐지 규칙을 생성한다.

4.2.1 소속정도 캐싱

퍼지 의사결정 트리 생성시 소속정도 캐싱 방법은 트리를 생성할 때 소속정도의 값을 캐싱하여 소속정도의 중복된 계산을 제거함으로 트리 생성에 소요되는 시간을 단축시킨다.

일반적으로 의사결정 트리를 생성하는 과정은 데이타를 분할하기 위해 각 속성의 무질서도(entropy)를 계산하여 분할할 속성을 선택한다. 무질서도를 계산하는 과정에서 일반 의사결정 트리의 경우 해당 속성에 포함된 각 클래스별 아이템 집합의 비율을 사용하여 계산한다. 하지만 퍼지 의사결정 트리의 경우 아이템 집합의 비율 대신 해당 속성이 갖는 퍼지 집합에 대하여 각 클래스별 퍼지 소속정도 비율을 사용하여 계산한다[9]. 이 과정에서 매 노드마다 무질서도를 계산하기 위해 소속정도 계산의 중복이 발생한다. 소속정도 계산의 중복을 없애기 위해 트리 생성 전에 각 속성이 갖는 퍼지 집합에 대해 소속정도를 계산하여 캐시에 저장한다. 매 노드에서 소속정도를 계산하지 않고 속성 ID와 소속함수 ID를 사용하여 캐시에 저장된 값을 사용함으로 중복 계산 시 소요되는 시간을 절약할 수 있다. 캐시를 사용하여 중복 계산 시 소요되는 시간을 줄임으로써 트리 생성에 소요되는 시간을 단축시킬 수 있다.

4.2.2 Zero-pruning

Zero-pruning은 퍼지 의사결정 트리의 평가 시 트리의 모든 규칙을 평가하지 않고, 부분적으로 의미가 있는 규칙만 이용하여 평가함으로써 트리 평가 시간을 단축시키는 방법이다. 의사결정 트리의 규칙은 트리의 루트 노드부터 단말 노드까지의 경로로 표현된다. 퍼지 의사결정 트리의 규칙도 일반 의사결정 트리와 같은 방법으로 표현된다. 하지만 트리 평가 시 일반 의사결정 트리는 평가할 데이타에 대하여 현재 노드가 갖는 조건을 기준으로 분류하여 하위 노드의 어느 한쪽에만 포함된다. 퍼지 의사결정 트리의 경우 조건에 사용된 퍼지 소

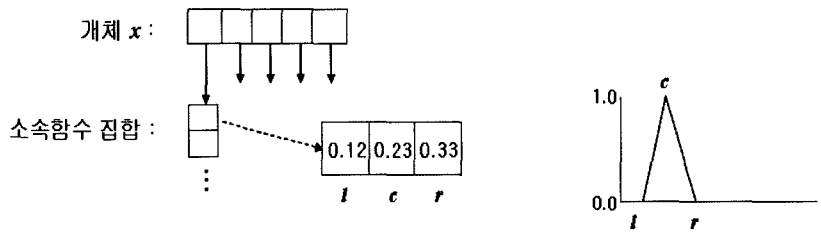


그림 6 소속함수 최적화 진화 알고리즘의 개체 인코딩

속합수의 소속정도에 따라 하위 노드에 포함되는 정도가 결정된다. 즉 동일한 데이타가 복수의 하위 노드에 포함 될 수 있다.

본 논문에서는 퍼지 의사결정 트리 평가 시 Mamdani의 min-max추론 방법을 사용한다[10]. 일반적인 min-max추론의 경우 평가할 데이타에 대해 트리의 모든 단말 노드까지 최소의 소속정도에 규칙 신뢰도를 곱하고, 단말 노드 중에 최대의 소속정도를 갖는 규칙의 클래스로 분류한다. 이때 본 논문에서는 zero-pruning 방법을 적용하여 중간 노드가 갖는 조건의 소속정도가 0.0인 경우 해당 하위 노드들은 더 이상 평가하지 않는다. 루트 노드부터 단말 노드까지 하나의 규칙으로 표현할 때 min-max추론의 경우 루트 노드에서 단말 노드까지의 조건들의 소속정도 중 최소값을 취한다. 이때 특정 노드의 소속정도가 0.0이 되는 경우 해당 노드를 포함하는 경로에서 최소값은 0.0이 되므로 해당 규칙은 더 이상 하위 노드를 평가할 의미가 없다. 이와 같이 트리 평가 시 트리의 전체 노드를 평가하지 않고 의미 있는 부분만 사용하여 평가하므로 트리 평가 시간을 단축시킬 수 있다.

4.3 분할 평가 방법

진화 알고리즘을 사용하여 퍼지 규칙과 같은 모델을 최적화 할 경우 개체 평가에 사용되는 데이타의 크기는 수행 시간과 밀접한 관련이 있다. 특히 진화 알고리즘의 개체 평가 단계의 수행 시간은 모델 생성과 모델 평가 단계를 수행하는 시간이다.분할 평가 방법은 개체 평가 과정에서 전체 데이타를 사용하는 대신 n개로 나누어진 부분 데이타를 이용하여 개체를 평가함으로써 수행 시간을 단축시키는 방법이다. 데이타를 분할할 때는 클래스별 계통추출 샘플링(systematic sampling)방법[11]을 사용하여 부분 학습 데이타를 생성한다. 만약 분할된 부분 데이타가 전체 데이타의 특성을 충분히 나타내지 못한다면 왜곡된 모델로 진화될 수 있기 때문에 이를 보완하기 위하여 개체 평가할 때마다 사전에 n개로 분할된 부분 데이타를 임의로 또는 일정한 순서로 선택하여 평가한다.

5. 실 험

제안한 방법의 성능 평가를 위해 사용된 실험데이타와 평가 방법을 기술하고 정확성과 효율성을 확인하기 위한 실험을 한다. 실험을 통해 제안한 방법과 기존에 제안된 방법과 비교 평가한다.

5.1 실험 데이타 및 평가 방법

본 논문에서는 침입 탐지 분야의 연구에서 가장 많이 사용되고 있는 1999년도 KDD(Knowledge Discovery and Data-mining) Cup 침입 탐지 데이타를 사용하여 실험한다[12]. KDD'99 침입 탐지 데이타는 MIT 링컨 연구소에서 1998 DARPA 침입 탐지 평가 프로그램(intrusion detection evaluation program)에 의해 침입 탐지 연구의 평가를 목적으로 미국 군사 네트워크(military network)에서 9주에 걸친 시뮬레이션(simulation)을 통해 제작되었다. 각 연결 기록은 41개의 독립적인 속성과 클래스로 구성되어 있다. 클래스는 연결기록에 대한 구분으로 정상 또는 공격 유형(attack type)을 갖고 있다. 학습 데이타와 평가 데이타의 구성은 표 1과 같다.

표 1 KDD'99 침입 탐지 데이타의 구성

구 분	학습 데이타	평가 데이타
속성 개수	41	41
클래스 개수	24	38
레코드 개수	494,021	311,029

표 1을 보면 학습 데이타와 평가 데이타의 클래스 개수가 다른 것을 확인할 수 있다. 평가 데이타에만 존재하는 14개의 클래스는 기존에 알려지지 않은 새로운 공격 유형으로 간주한다. 총 38개의 클래스는 KDD'99 Cup에서 제공한 클래스 범주화 스크립트를 사용하여 'Normal', 'Probe', 'DOS', 'U2R', 'R2L'로 대분류 5개의 클래스로 범주화한다.

본 논문에서는 KDD'99 Cup 데이타의 41개 속성 중에 카테고리(category)형을 갖는 3개의 속성 'protocol', 'service', 'flag'를 제외하고 나머지 38개의 수치(numeric)형을 갖는 속성을 사용한다. 퍼지 규칙은 연속적이고 애매한 값을 표현할 때 매우 유용하고 효과적이다. 카테고리 형을 갖는 속성을 제외한 것은 침입 탐지에서 퍼지 규칙 표현 시 의미가 없음을 나타낸다. 제안한 방법에 의해 생성된 침입 탐지를 위한 퍼지 분류 규칙의 성능을 평가하기 위해 정확도와 탐지 비용, 오탐지율, 미탐지율의 항목을 사용한다. 정확도는 생성된 퍼지 규칙의 분류 정확도를 의미하고 전체 평가 데이타 중에 정확히 분류한 백분율로 나타낸다. 탐지 비용 항목은 KDD'99 Cup의 탐지 비용 행렬(cost matrix)을 사용하여 계산한다[13]. KDD'99 Cup의 탐지 비용 행렬은 침입 탐지에 대한 비용을 산정하기 위하여 공격 유형에 대한 위험 등급을 정하고 그에 따른 비용을 정의한 행렬을 사용하여 대회 참가자들이 제안한 방법의 결과에 대한 비용을 계산하였다. 표 2는 탐지 비용 행렬이다.

탐지 비용 행렬은 $|CLASS|\times|CLASS|$의 크기로 구성되고, 행은 평가 데이타가 갖는 실제 클래스를 나타내고 열은 퍼지 분류 규칙에 의해 예측된 클래스를 나타낸다. 각 원소들은 평가 데이타의 실제 클래스와 예측된 클래

표 2 KDD'99에서 사용한 침입 등급에 따른 탐지 비용 행렬

		예측 클래스				
		Normal	Probe	DOS	U2R	R2L
실제 클래스	Normal	0	1	2	2	2
	Probe	1	0	2	2	2
	DOS	2	1	0	2	2
	U2R	3	2	2	0	2
	R2L	4	2	2	2	0

스에 해당하는 탐지 비용을 가지고 있다.

정상을 공격으로 분류하는 경우를 오탐지라 하고 공격을 정상으로 분류하는 경우를 미탐지라 한다. 침입 탐지 규칙을 평가하는 항목으로 오탐지와 미탐지에 대해 백분율로 나타낸 오탐지율과 미탐지율을 사용한다. 각 항목에 대한 계산식은 식 (8), 식 (9)에 의해 계산된다. 침입 탐지에서 FPR(false positive ratio)는 오탐지율 즉, 정상 데이타를 침입으로 분류한 비율을, FNR(false negative ratio)은 미탐지율, 즉 침입 데이타를 정상으로 분류한 비율을 나타낸다.

$$FPR = \frac{\text{정상을 침입으로 분류한 개수}}{\text{정상 데이타 개수}} \times 100 \quad (8)$$

$$FNR = \frac{\text{침입을 정상으로 분류한 개수}}{\text{침입 데이타 개수}} \times 100 \quad (9)$$

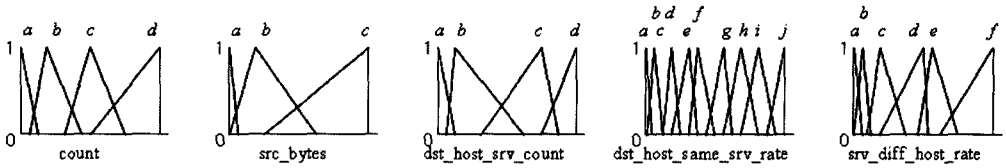
5.2 실험 및 성능분석

본 절에서는 제안한 방법에서 사용하고 있는 소속함수 최적화 진화 알고리즘의 적합도 함수의 가중치(w) 변화에 따른 성능 차이를 조사한다. 가중치는 1에 가까우면 정확성이 높은 퍼지 분류 규칙으로 진화하게 되고, 반대로 0에 가까우면 간결한 규칙으로 진화하게 된다. 또한 소속함수 최적화 진화 알고리즘을 수행하기 위한 기본 매개 변수는 개체 집단 크기와 교배 확률과 돌연변이 확률이다. 그리고 개체 평가시 사용하는 퍼지 의사결정 트리에서 사용하는 매개 변수로 임계값 θ_d, θ_e가 있다. θ_d는 해당 노드의 주 클래스 비율에 대한 임계값을 의미하고, θ_e는 전체 데이타 중에 해당 노드에 소속되는 소속정도 비율의 임계값을 의미한다. 본 실험에서는 개체 집단 크기를 20으로 설정하였고 교배 확률은 0.4, 돌연변이 확률은 0.1로 설정하였다. 퍼지 의사결정 트리의 임계값 θ_d와 θ_e는 각각 0.85와 0.01로 설정하였다. 소속함수 최적화 진화 알고리즘의 진화 종료 조건으로는 엘리트 개체의 정확도가 95%이상 이거나 엘리트 개체의 적합도가 30세대 동안 변하지 않을 경우에 종료하고 최대 100세대까지 진화하도록 설정하였다. 실험은 학습 데이타와 평가 데이타에서 각각 3,000건을 샘플링하여 진화 과정에서 사용하고, 진화 종료 후 311,029건의 평가 데이타로 평가하였다. 이때 샘플링 방법은 클래스별 계통추출 방법[11]을 사용하였다. 실험은 랜덤 초기값을 0에서 9까지 변경하면서 10회 반복 수행하여 평균을 구했다. 실험에서 사용한 시스템 환경은 Intel Pentium4 1.8Ghz 프로세서를 사용하고, 운영체제는 Microsoft Windows XP SP2, 주메모리 1,024MB를 사용하였다. 표 3는 분할 평가 알고리즘의 분할 개수와 가중치 변화에 따른 침입 탐지 성능의 변화를 보여준다.

표 3의 실험 결과를 통해 가중치가 증가하면서 평균 정확도와 평균 규칙 개수가 증가함을 확인할 수 있다. 가중치를 증가 시킬 때 마다 평균 정확도의 평균 증가율은 약 1%이지만, 평균 규칙의 개수와 조건항의 개수는 약 2배 증가하였다. 3,000개의 전체 데이타를 실험에 사용하였을 경우에 비해 10분할 진화를 사용한 경우는 가중치 0.9의 경우를 제외하고 전체 데이타를 사용한 것과 비슷한 정확도를 갖는 규칙이 생성되었고 평균적으로 규칙개수는 약 15% 감소하였다. 또 정확도가 높아짐에 따라 평가 데이타를 정확하게 분류한 개수가 증가하므로 탐지 비용이 감소하는 것을 확인할 수 있다. 오탐지율 FPR의 경우 가중치 0.1일 때 4.9로 가장 낮고 그 외의 경우 동일한 결과를 보인다. 하지만, 미탐지율 FNR의 경우 가중치가 증가함에 따라 감소하는 특성이 있고 특히 0.9일 때 낮은 미탐지율을 보인다. 이때 생성되는 규칙의 특성은 가중치가 증가함에 따라 정상 데이타 보다는 침입 데이타를 잘 분류하는 특징을 갖는다. 이것은 가중치가 증가함에 따라 정확성의 비중을 크게 하여 진화하게 되고 그 결과 평가 시 사용된 데이타가 침입에 비해 정상 데이타의 빈도는 상대적으로 낮기 때

표 3 데이타 분할 개수와 가중치 변화에 따른 성능 변화

데이타 분할	가중치	정확도 (%)	규칙수	조건항수	비용	FPR (%)	FNR (%)	평가 시간(초)
전체	0.1	92.04	69.2	224.0	0.2500	4.9	7.8	2.47
	0.5	92.96	71.6	184.2	0.2281	5.4	6.7	2.78
	0.9	94.25	178.4	513.3	0.1847	5.4	4.9	3.28
10분할	0.1	92.12	63.5	166.8	0.2440	5.7	7.3	0.75
	0.5	93.09	65.6	158.9	0.2221	5.8	6.1	0.76
	0.9	93.54	144.7	399.5	0.1960	8.4	4.8	0.95



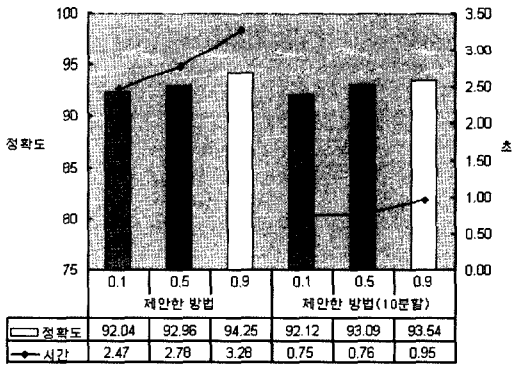
if count(*a*) *and* src_bytes(*b*) *then* R2L (*0.8*)
if count(*a*) *and* src_bytes(*c*) *then* Normal (*1.0*)
...
if count(*b*) *and* src_bytes(*b*) *and* dst_host_srv_count(*a*) then U2R (*0.4*)
...
if count(*c*) *and* src_bytes(*c*) *and* dst_host_same_srv_rate(*j*) *and* srv_diff_host_rate(*a*) *then* Normal (*0.7*)
if count(*c*) *and* src_bytes(*c*) *and* dst_host_same_srv_rate(*j*) *and* srv_diff_host_rate(*b*) *then* DOS (*0.5*)
if count(*c*) *and* src_bytes(*c*) *and* dst_host_same_srv_rate(*j*) *and* srv_diff_host_rate(*c*) *then* R2L (*0.4*)
if count(*c*) *and* src_bytes(*c*) *and* dst_host_same_srv_rate(*j*) *and* srv_diff_host_rate(*d*) *then* Normal (*0.6*)
...
if count(*d*) *and* dst_host_srv_count(*a*) *then* DOS (0.9)
if count(*d*) *and* dst_host_srv_count(*b*) *then* Normal (1.0)
if count(*d*) *and* dst_host_srv_count(*c*) *then* DOS (0.8)
if count(*d*) *and* dst_host_srv_count(*d*) *then* Probe (1.0)

그림 7 제안한 방법에서 생성된 퍼지 소속함수와 퍼지 규칙

문에 침입을 잘 분류하는 규칙이 생성된다. 그림 7은 제안한 방법을 사용하여 자동으로 생성된 퍼지 소속함수와 퍼지 규칙의 일부이다. 그림 7의 소속함수와 규칙은 3,000개 데이타를 사용하여 가중치 0.9, 랜덤 초기값 0.0으로 설정하여 최종 생성된 165개의 규칙중 일부이다.

각 속성의 명칭은 KDD'99 데이타의 고유 명칭으로 표기하였다. count는 2초 동안 동일한 호스트에서의 연결 횟수, src_bytes는 소스 호스트에서 대상 호스트로 보낸 패킷의 누적 용량, dst_host_srv_count는 대상 호스트에서 2초 동안 동일한 서비스의 연결 횟수, dst_host_same_srv_rate는 대상 호스트에서 동일한 서비스의 비율, srv_diff_host_rate는 소스 호스트에서 서로 다른 호스트에 연결한 비율을 나타낸다. 각 규칙의 조건절은 해당 속성의 소속함수로 구성되고 규칙의 결론절과 규칙신뢰도 값으로 구성된다. 학습데이타에 포함 빈도가 높은 Normal, DOS는 신뢰도가 높으나 상대적으로 빈도가 낮은 R2L, U2R에 대해서는 신뢰도가 낮은 특징이 있다.

표 3의 평가 시간 항목은 진화 알고리즘 수행 시 개체 평가 단계에서 한 개체의 평가에 소요되는 시간을 의미한다. 가중치가 증가함에 따라 생성되는 규칙 개수가 많아지고 개체 평가 시간이 증가하게 된다. 개체 평가 시 규칙 개수가 많은 경우 평가해야 할 규칙이 많아지고 그만큼 시간이 증가한다. 그림 8은 전체 데이타를 이용한 방법과 분할 평가 방법을 적용한 실험의 소요된 시간과 정확도를 비교한 결과이다. 그림 8에서 막대 그래프는 정확도를 나타내고 꺾은선은 개체 당 평가 시

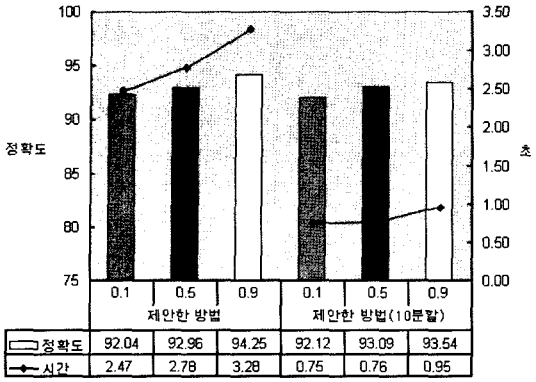


	제안한 방법			제안한 방법(10분할)		
정확도	92.04	92.96	94.25	92.12	93.09	93.54
시간	2.47	2.78	3.26	0.75	0.76	0.95

그림 8 분할 진화 알고리즘의 효율성 실험 결과

소요된 시간을 나타낸다. 분할 평가 방법을 적용하여 개체 평가 시간이 약 70% 감소하고 탐지 성능도 전체 데이타를 사용한 것과 유사한 효과를 얻을 수 있다.

소속함수 최적화 진화 알고리즘에서 개체 평가 시 사용되는 퍼지 의사결정 트리의 생성과 평가 시간을 단축시키기 위한 방법으로 소속정도 캐싱과 zero-pruning을 제안하였다. 그림 9는 소속정도 캐싱과 zero-pruning의 적용에 따른 트리 생성과 평가 시간이다. 실험 시 사용한 실험 변수는 표 3의 실험 변수와 동일하고 가중치와 랜덤 초기값만 각각 0.9, 0.0으로 설정하여 실험하였다. 실험한 결과 하나의 개체가 갖는 트리의 생성과 평가시간이 7.78초에서 2.53초로 약 67% 감소하였다. 일반적인 퍼지 의사결정 트리의 생성과정에 캐싱를 적용하여 트리생성 시간이 약 25% 감소하고 zero-pruning을 적용하여 트리(또는 개체)평가에 소요되는 시간이 약 90% 감소하였다.



	제안한 방법			제안한 방법(10분할)		
정확도	92.04	92.96	94.25	92.12	93.09	93.54
시간	2.47	2.78	3.28	0.75	0.76	0.95

그림 9 소속정도 캐싱과 zero-pruning에 따른 트리 생성 시간

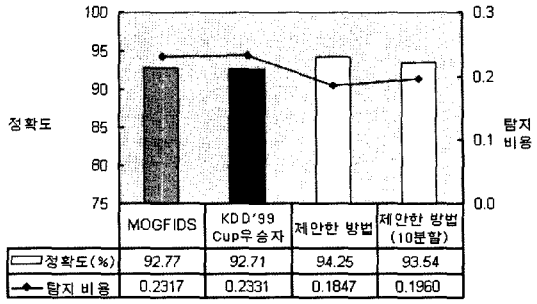
5.3 기존 방법과 성능 비교

표 4는 제안한 방법을 기존 방법과 비교한 결과이다. 실험 결과 제안한 방법이 기존 방법보다 생성된 침입 탐지 규칙이 평균적으로 더 정확하고 탐지 비용이 감소된 것을 알 수 있다. 또한 생성된 규칙 개수에 대해서도 제안한 방법이 MOGFIDS보다 평균적으로 감소한 것을 확인할 수 있다.

표 4 MOGFIDS와 KDD'99 Cup우승자, 제안한 방법의 성능 비교

구 분	가중치	정확도(%)	규칙 개수	탐지 비용
MOGFIDS[5]	-	92.77	148.0	0.2317
KDD'99 Cup 우승자[13]	-	92.71	-	0.2331
제안한 방법	0.1	92.04	69.2	0.2500
	0.5	92.96	71.6	0.2281
	0.9	94.25	178.4	0.1847
제안한 방법 (10분할)	0.1	92.12	63.5	0.2440
	0.5	93.09	65.6	0.2221
	0.9	93.54	144.7	0.1960

임의로 초기 소속함수를 생성[5]하는 것 보다 데이타 특성을 고려하여 생성한 소속함수로 퍼지 분류 규칙을 생성하는 것이 보다 정확하고 간결한 규칙을 생성한다는 것을 알 수 있다. 그림 10은 기존 방법과 제안한 방법의 정확도, 탐지 비용을 그래프로 나타낸 것이다. 제안한 방법의 정확도와 탐지 비용은 표 4에서 실험한 가중치 0.9일 때의 값이다. 정확도가 높아짐에 따라 상대적으로 탐지 비용은 감소하는 것을 알 수 있다. 5.1절의 탐지 비용 행렬에서 확인한 바와 같이 침입 여부를 정확이 분류할 경우 탐지에 소요되는 비용을 절감할 수 있다. 최적화를 통해 제안한 방법에서 가중치가 0.9일 때 기존에 제안된 MOGFIDS보다 정확도는 1.48% 높아



	MOGFIDS	KDD'99 Cup우승자	제안한 방법	제안한 방법 (10분할)
정확도(%)	92.77	92.71	94.25	93.54
탐지 비용	0.2317	0.2331	0.1847	0.1960

그림 10 기존 방법과 제안한 방법과 정확도, 탐지 비용 비교

지고 탐지 비용은 20.3% 절감되었다. KDD'99 Cup 우승자의 경우 정확도는 1.54% 높아지고 탐지 비용은 20.8% 절감되었다.

6. 결론 및 향후 연구

본 논문은 퍼지 소속함수의 개수와 형태를 고정하지 않은 상태에서 침입 탐지에 적합한 정확하고 간결한 퍼지 분류 규칙을 생성할 수 있는 방법을 제안하였다. 제안한 방법은 지도 군집화로 클래스 분포를 고려하여 초기 소속함수를 생성한다. 생성된 초기 소속함수는 정확도가 높고 간결한 침입 탐지 규칙의 생성을 위해 진화 알고리즘을 사용하여 최적화한다. 진화 알고리즘은 최적화 문제에 효과적이지만 수행 시간에 대한 효율성이 낮다. 특히 개체 평가 단계에서 많은 시간이 소요된다. 본 논문에서는 진화 알고리즘의 수행 시간을 단축시키기 위해 분할 평가 방법을 제안하였다. 또한 소속함수 최적화 진화 알고리즘의 개체 평가 단계에서 사용되는 퍼지 의사결정 트리의 생성과 평가 시간을 단축시키는 방법으로 소속정도 캐싱과 zero-pruning을 제안하였다.

제안한 방법은 KDD'99 Cup에서 사용한 침입 탐지 벤치마크 데이타를 사용하여 기존에 제안된 방법들과 비교한 결과 정확도가 1.48% 향상 되었다. 또한 진화 알고리즘의 수행 시간을 단축시키기 위해 분할 평가 방법을 적용하여 평균적으로 70%의 시간을 단축시켰다. 그리고 zero-pruning과 소속정도 캐싱 방법을 적용하여 트리의 생성과 평가 각각에 대해 45%, 90%의 소요시간을 단축시킴으로써 진화 알고리즘의 시간적 효율성을 향상시켰다. 제안한 방법은 확률적 기반의 교배 연산과 돌연변이 연산만을 사용하기 때문에 휴리스틱 연산 사용에 비해 진화 수렴 속도가 느리다.

향후 연구로는 수렴 속도를 효율적으로 개선함으로써 진화 알고리즘의 수행시간을 단축시킬 수 있는 휴리스틱 진화 연산에 대한 연구가 요구된다. 또한 학습 데이타에서 상대적으로 빈도가 낮은 U2R이나 R2L과 같은

공격 유형에 대한 효과적인 탐지 방법에 대한 연구가 필요하다. U2R과 R2L의 경우 발생 빈도는 낮지만 다른 유형의 공격에 비해 피해를 입었을 경우 발생하는 비용이 높다. 마지막으로 제안한 방법을 실제 침입 탐지 시스템에 적용함으로써 보다 일반화된 타당성을 검증하는 것이 필요하다.

참 고 문 헌

[1] 이홍섭, 2005 정보 보호 실태 조사, 한국정보보호진흥원, 2005.

[2] C. Xiang, S. M. Lim, "Design of multiple-level hybrid classifier for intrusion detection system," Machine Learning for Signal Processing, IEEE, pp.117-122, 2005.

[3] J. Gomez, D. Dasgupta, "Evolving fuzzy classifiers for intrusion detection," International Proceedings of the IEEE Workshop on Information Assurance, 2002.

[4] J. Gomez, D. Dasgupta, O. Nasraoui, F. Gonzalez, "Complete expression trees for evolving fuzzy classifier systems with genetic algorithms and application to network intrusion detection," Fuzzy Information Processing Society, IEEE, pp.469-474, 2002.

[5] Chi-Ho Tsang, S. Kwong and H. Wang, "Anomaly intrusion detection using multi-objective genetic fuzzy system and agent-based evolutionary computation framework," International Conference on Data Mining, IEEE, pp.789-792, 2005.

[6] 김성은, 류정우, 김명원, "효율적인 지도 퍼지 군집화를 위한 휴리스틱 분할 진화알고리즘", 한국종합 컴퓨터 학술대회 논문집, 한국정보과학회 제32권 제1호(B), pp.667-669, 2005.

[7] 류정우, 김성은, 김명원, "효율적인 진화알고리즘을 이용한 적응형 퍼지 분류 규칙 생성", 한국정보과학회 추계학술대회 논문집 제32권 2호, pp.769-771, 2005.

[8] J. Roubos, M. Setnes, J. Abonyi, "Learning fuzzy classification rules from labeled data," International Journal of Information Sciences, pp.77-93, 2003.

[9] M. W. Kim, J. G. Lee, "Classification Fuzzy Rule Generation Based on Fuzzy Decision Tree," The Journal of Electrical Engineering and Information Science, Vol.5, NO.3, pp.264-272, 2000.

[10] J.-S. R. Jang, C.-T. Sun, E. Mizutani, Neuro-Fuzzy and Soft Computing, Prentice-Hall International, Inc., 1997.

[11] R. L. Scheaffer, W. Mendenhall III, R. L. Ott, Elementary Survey Sampling, 5th edition, Duxbury Press, 1996.

[12] Kdd cup 1999 data set, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, The UCI KDD Archive, University of California.

[13] C. Elkan, "Results of the KDD'99 classifier learning," ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp.63-64, 2000.

김 성 은

1999년 3월~2003년 2월 동양대학교 소프트웨어공학과(공학사). 2004년 9월~2006년 8월 2006 숭실대학교 일반대학원 컴퓨터학과(공학석사). 2006년 8월~현재 (주)퓨쳐시스템 정보통신연구소 주임연구원. 관심분야는 데이타마이닝, 퍼지추론, 진화알고리즘, 침입탐지 규칙생성

길 아 라

정보과학회논문지 : 소프트웨어 및 응용 제 34 권 제 1 호 참조

김 명 원

정보과학회논문지 : 소프트웨어 및 응용 제 34 권 제 1 호 참조