진화신경망을 이용한 효과적인 침입탐지
(Effective Intrusion Detection using Evolutionary Neural Networks)

한 상 준 † 조 성 배 ††
(Sang-Jun Han) (Sung-Bae Cho)

요 약 시스템 호출 감사자료기반 기계학습기법을 사용한 프로그램 행위 학습방법은 효과적인 호스트 기반 침입탐지 방법이며, 규칙 학습, 신경망, 통계적 방법, 은닉 마크로프 모델 등의 방법이 대표적이다. 그 중에서 신경망은 시스템 호출 시퀀스를 학습하는데 있어 적합하다고 알려져 있는데, 실제 문제에 적용하여 좋은 성능을 내기 위해서는 그 구조를 결정하는 것이 중요하다. 하지만 보통의 신경망은 그 구조를 찾기 위한 방법이 알려져 있지 않아 침입탐지에 효과적인 구조를 찾기 위해서는 많은 시간이 요구된다. 본 논문에서는 기존 신경망 기반 침입탐지시스템의 단점을 보완하고 성능을 향상시키기 위해 진화신경망을 이용한 방법을 제안한다. 진화 신경망은 신경망의 구조와 가중치를 동시에 학습하기 때문에 일반 신경망보다 빠른 시간에 더 좋은 성능의 신경망을 얻을 수 있다는 장점이 있다. 1999년의 DARPA IDEVAL 자료로 실험한 결과 기존의 연구보다 좋은 탐지율을 보여 진화신경망이 침입탐지에 효과적임을 확인할 수 있었다.

키워드 : 침입탐지시스템, 비정상행위 탐지, 유전자 알고리즘, 진화 신경망

Abstract Learning program's behavior using machine learning techniques based on system call audit data is an effective intrusion detection method. Rule learning, neural network, statistical technique, and hidden Markov model are representative methods for intrusion detection. Among them neural networks are known for its good performance in learning system call sequences. In order to apply it to real world problems successfully, it is important to determine their structure. However, finding appropriate structure requires very long time because there are no formal solutions for determining the structure of networks. In this paper, a novel intrusion detection technique using evolutionary neural networks is proposed. Evolutionary neural networks have the advantage that superior neural networks can be obtained in shorter time than the conventional neural networks because it learns the structure and weights of neural network simultaneously. Experimental results against 1999 DARPA IDEVAL data confirm that evolutionary neural networks are effective for intrusion detection.

Key words : intrusion detection system, anomaly detection, genetic algorithm, evolutionary neural networks

1. 서 론

정보통신 기술의 발전으로 정보보호의 중요성이 점점 커지고 있는 가운데 침입탐지시스템은 시스템 보안을 위한 필수적인 도구로 자리 잡고 있다. 침입탐지 기법에는 미리 알려진 공격행위에 대한 정보를 구축하고 이를 이용해 침입을 판정하는 오용탐지 방법과 사용자나 프로그램의 정상행위에 대한 정보를 구축하고 이를 이용하는 비정상행위 탐지 방법의 두 가지가 있다. 대부분의 상업용 침입탐지 시스템은 오용탐지 기법을 적용한 규칙기반 침입탐지 시스템이기 때문에 새로운 공격에 대한 탐지가 힘들다. 이런 오용탐지 기법의 단점보완을 위해 최근에는 비정상행위기반의 침입탐지 시스템에 관한 연구가 활발한데, 비정상행위기법의 경우에도 정상행위를 제대로 학습하지 않을 경우 많은 수의 false-positive 오류가 생긴다는 단점이 있다[1].

· 본 연구는 과학기술부가 지원한 뇌과학 연구 프로그램에 의해 지원되었음
† 학생회원 : 연세대학교 컴퓨터과학과
sangjunhan@yonsei.ac.kr
†† 종신회원 : 연세대학교 컴퓨터과학과 교수
sbcho@yonsei.ac.kr
논문접수 : 2004년 2월 6일
심사완료 : 2005년 2월 15일

호스트기반 비정상행위 탐지에는 프로그램의 행동을 분석하는 방법이 많이 사용되고 있다. 정상 프로그램의 행동을 학습하고 이와는 상이한 행동을 침입으로 잡아내는 문제는 일반적인 인공지능의 이진분류문제로 바꾸어 볼 수 있어 규칙 학습, 신경망, 통계적 방법, 은닉 마크로프 모델 등의 기계학습 방법이 많이 사용되어 좋은 성능을 보였다. 그중에서도 신경망은 호스트기반 침입탐지 방법 중 가장 좋은 성능을 보였다. 하지만 감사자료의 특성상 학습데이타의 크기가 매우 크고 대부분의 기계 학습 알고리즘이 계산량이 많기 때문에 정상행위 모델링 과정에 있어서 많은 시간을 필요로 한다. 또한 분류기의 구조와 설정에 따라 성능이 좌우되기 때문에 실제 문제에 신경망을 적용하여 좋은 성능을 얻기 위해서는 각 응용분야에 적합한 가중치, 은닉 노드 수, 위상 구조 등을 설정하는 것이 매우 중요하다. 하지만 적합한 구조에 대한 많은 연구가 진행되었음에도 불구하고 이를 결정하는 정형화된 방법은 알려져 있지 않다[2]. 따라서 대부분의 경우 유사한 응용 분야에서 사용했던 경험에 기반하여 시도와 오류 과정의 반복을 통해서 설계되기 때문에, 기계학습 기법을 이용한 침입탐지는 학습단계에서 매우 많은 시간이 소요되는 단점이 있다.

본 논문에서는 기존 기계학습 기반 침입탐지기법의 단점을 극복하기 위하여 진화신경망을 사용하였다. 진화신경망은 분류기의 구조와 내부의 가중치를 동시에 학습하기 때문에 구조를 결정하기위한 시행착오 과정이 필요 없을뿐만 아니라 문제에 최적화된 구조의 신경망을 자동으로 얻을 수 있다는 장점이 있다. 따라서 일반적인 기계 학습방법보다 빠른 시간내에 좋은 성능의 분류기를 얻을 수 있는데, 실제 감사자료를 사용한 실험과 다른 연구와의 비교를 통해 이를 검증한다.

본 논문의 나머지 부분은 다음과 같이 구성된다. 2장은 프로그램의 행동학습을 통한 침입탐지 방법에 대해 소개하고, 3장은 진화 신경망과 제안하는 기법에 대해 설명한다. 4장은 실험 과정과 결과를 설명하고, 5장은 논문의 결론과 향후연구에 대해 언급한다.

2. 관련연구

많은 유형의 침입은 프로그램의 버그를 이용해 오동작을 유도함으로써 이루어지기 때문에 공격을 받은 프로그램은 정상적인 실행과는 다른 행동을 하게 된다. 따라서 정상적인 프로그램을 학습한 후 이와 다른 행동을 보이는 프로그램을 침입으로 간주하는 방식은 효과적인 비정상행위 탐지 방법이 될 수 있다. 프로그램의 행동을 관찰하는 방법에는 여러 가지가 있지만 호스트기반 침입탐지시스템에서는 주로 프로그램이 사용한 시스템 호출을 기록한 감사자료가 많이 사용되며, 정상적인 프로그램 실행에서 생성된 일정크기의 시스템 호출 시퀀스 데이타를 수집하여 이와는 다른 시스템 호출 시퀀스를 보이는 프로그램을 침입으로 간주한다. 즉 프로그램 행동 학습은 시간순서의 시퀀스 데이타를 학습하고 분류하는 문제로 다시 정의될 수 있으며 전통적으로 이런 문제에 좋은 성능을 보였던 규칙기반방법, 신경망, HMM-(Hidden Markov Model) 등 많은 기계학습방법들이 활발히 적용되어 왔다.

프로그램 행동학습과 시스템 호출 시퀀스 데이타를 통한 침입탐지는 다음과 같이 정리될 수 있다. 어떤 프로그램이 N개의 시스템 호출 이벤트를 사용하였을 때 모든 이벤트의 집합을 P라고 하고, 시간 t에서 길이 L의 크기로 P를 윈도윙(windowing) 하여 만들어낸 시퀀스의 집합을 S_t라 할 때 P와 S_t는 다음과 같이 표현된다.

$$P = (s_1, s_2, \dots, s_N)$$

$$S_t = (s_{t+1}, s_{t+2}, \dots, s_{t+L}), t \le N - L$$

이때 R_t 를 S_t에 대한 시퀀스 평가함수 eval을 통해 나온 결과 값 이라 할 때 $R_t = eval(S_t)$ 의 값이 정해진 임계값보다 높은 경우 현재 프로세스는 침입으로 판단된다.

$$alarm\ (R_t) = \begin{cases} normal & \text{if } R_t \ge threshold \\ attack & \text{if } R_t < threshold \end{cases}$$

이와 같은 방법을 사용한 대표적인 연구로는 표 1과 같은 것들이 있다.

표 1 대표적인 프로그램 행동학습 기반 침입탐지 기법

기관	년도	기법	데이타
Univ. of New Mexico	1996-1999	Equality Matching, stide, t-stide, RIPPER, HMM	UNM
Columbia Univ.	1997-2001	RIPPER, Sparse Markov tree	1998 IDEVAL, UNM
Arizona State Univ.	2001	Decision tree, Multivariate test, Markov chain	1998 IDEVAL
Reliable Software Technologies	1999-2000	Neural network, Elman network	1998, 1999 IDEVAL

S. Forrest 등은 프로그램 행동학습을 통한 침입탐지 방법을 최초로 제안하였는데 컴퓨터 면역 시스템의 개념을 적용한 침입탐지의 한 방법으로 시스템 호출과 동등 매칭기법을 사용한 침입탐지방법을 제시하였다[3, 4]. 정상적인 프로그램의 모든 시스템 호출 시퀀스를 저장한 후 모니터 되는 프로그램이 발생 시킨 시스템 호출 자료 중 부정합된 시퀀스가 일정 빈도이상을 넘어설 경

우 침입으로 탐지하는 방법을 사용하였다. 그 후 [5]에서는 해밍 거리를 사용한 방법을 제시하였다. [6]에서는 단순한 나열 방법, 상대적 빈도 비교방법, 규칙 학습방법, 은닉 마르코프 모델(HMM)의 성능을 비교하였다. 실험 결과 HMM이 가장 좋은 성능을 보였으나 학습과정에 시간이 많이 소요되는 단점이 있었다.

J. Stolfo 등은 규칙 학습방법인 RIPPER[7]를 프로그램 행동학습에 적용하였다[8]. 정상적인 시퀀스와 비정상적인 시퀀스를 분류하는 규칙을 생성한 후 새로운 시퀀스에 이 규칙을 적용하여 침입을 탐지하였다. Sendmail 프로그램이 발생시킨 시스템 호출 시퀀스를 이용하여 그 가능성을 검증하였다. 또한 [9]에서는 동적인 윈도우 길이를 사용하는 방법을 제안하였다. MIT Lincoln Lab의 1999년 DARPA IDEVAL(Intrusion Detection Evaluation) 데이타와 UNM(Univsersity of New Mexico) 데이타를 이용하여 기존의 고정 윈도우 길이 방법보다 성능이 더 좋음을 입증하였다.

N. Ye 등은 확률기반의 학습방법을 이용하여 프로그램의 행동을 학습하였다[10]. 시스템 호출 시퀀스를 학습하는데 이용 가능한 빈도적 특징과 순서적 특징을 제시하고 각 특징에 적합한 학습 방법을 제안하였다. 의사결정트리와 chi-square 다변량 검증은 시스템 호출 시퀀스의 빈도적 특징을 학습하는데, 사용하였고 Markov chain 모델로는 시스템 호출의 순서적인 특징을 학습하였는데 1998년 DARPA IDEVAL감사자료를 사용하여 성능을 비교한 결과 순서적 특징을 사용한 Markov chain 모델이 가장 좋은 성능을 보였다. 이는 시스템 호출 시퀀스에서는 각 이벤트간의 순서가 침입탐지에 중요한 정보를 제공함을 의미한다.

A.K. Ghosh 등은 프로그램 행동학습에 신경망을 적용하였다[11,12]. 전통적인 전방향 오류 역전파 알고리즘을 사용하는 다층 신경망과 Elman recurrent 신경망을 사용하였다. 신경망은 불완전한 데이타에 대해서도 일반화하는 성능이 좋기 때문에 시퀀스 데이타 분류에 좋은 성능이 기대된다. 또한 일반적인 신경망과는 달리 Elman recurrent 신경망은 연결구조가 입력 시퀀스 사이의 상태정보를 저장하는 특성이 있기 때문에 시퀀스 정보를 학습하는데 더 적합하다. 1998년과 1999년의 DARPA IDEVAL 데이타를 사용하여 실험한 결과 Elman 신경망의 성능이 더 뛰어나 신경망의 구조가 침입탐지 성능에 큰 영향을 미치는 것을 확인할 수 있었다.

3. 진화신경망기반 침입탐지

프로그램 행동학습에 기계학습기법이 성공적으로 사용되어 왔으나 이를 실제 문제에 적용하기 위해서는 분류기의 구조, 학습에 필요한 여러 가지 변수 등을 응용분야에 맞게 설정해주는 과정이 필수적이다. 그러나, 이와 같은 문제는 표준화된 해결책이 없어 주로 시행착오에 의한 방법이 사용되어 왔다. 기계학습을 사용한 연구 중 가장 좋은 성능을 보인 A.K. Ghosh 등의 연구에서는 10, 15, 20, 25, 30, 35, 40, 50, 60개의 은닉 노드를 가지는 신경망을 각 10개씩 학습시켜 각 프로그램마다 모두 90개의 신경망을 생성하고 그중 가장 좋은 성능을 보인 신경망을 선택하는 방식을 취하였다[11]. 또한 시퀀스데이타 학습에 좋다고 알려진 HMM을 사용할 경우에도 상태수와 상태들 사이의 연결 관계가 성능에 큰 영향을 미친다. 하지만 이러한 기계학습 기법의 학습알고리즘은 매우 많은 계산량을 요하고, 특히 침입탐지분야의 경우 감사자료의 크기가 방대하기 때문에 많은 시간이 필요하게 된다. 또한 프로그램기반 학습의 경우에는 사용자기반 학습방법보다 많은 수의 신경망을 필요로 하기 때문에 반복적인 시행착오를 통해 경험적으로 결정하는 방법은 더욱 힘들게 된다. 그렇기 때문에 신경망의 가중치뿐만 아니라 구조까지 자동으로 설계해주는 방법이 필요하다.

초기 신경망 자동 설계에 관한 연구에서는 다양한 컨스트럭티브(constructive) 알고리즘과 가지치기(pruning) 알고리즘을 사용하였다[2]. 컨스트럭티브 알고리즘은 최소의 은닉층과 은닉 노드 및 연결 정보를 가진 신경망에서 시작해서 학습하면서 필요에 따라 새로운 은닉층과 노드 및 연결 정보를 추가하는 방법이다. 가지치기 알고리즘은 반대로 불필요한 은닉층, 노드, 연결 정보 등을 제거하면서 신경망을 설계하는 방식이다. 하지만 두 방식 모두 신경망의 전체적인 구조 영역을 검색하는 것이 아니라 주어진 환경에서 제한된 영역만을 탐색하므로 최적화된 신경망을 찾기 어렵다.

이런 한계를 극복하기 위하여 진화 알고리즘이 도입되었다. 진화 알고리즘은 일반적인 모든 탐색 문제에 적용될 수 있으며, 초기 조건에 덜 민감한 전역 탐색 능력을 가진다. 진화 신경망은 진화 알고리즘을 신경망 설계 과정에 도입해서 자동으로 신경망을 결정하는 방법으로 신경망의 가중치, 위상 구조, 은닉 노드 수, 학습 알고리즘 등 신경망 학습 시 결정해야 하는 인자들을 여러 세대 진화를 통해 찾아서 최적의 신경망을 결정한다.

본 논문에서는 기존 신경망 기반 침입탐지시스템의 단점을 보완하고 성능을 향상시키기 위해 진화신경망을 이용한 방법을 제안한다. 진화 신경망은 학습에 구조를 결정하는 과정이 포함되기 때문에 시행착오를 반복할 과정이 필요가 없다. 따라서 빠른 시간에 적은 노력으로 높은 성능의 신경망을 만들 수 있다. 또한 신경망의 구조에 제약이 없어 정형화된 구조를 사용하는 신경망보다 응용분야에 더 최적화된 구조를 찾아낼 수 있다. 그

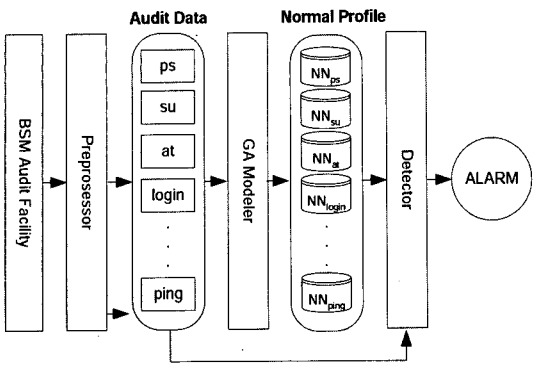

그림 1 제안하는 방법의 구조

림 1은 진화신경망을 사용한 침입탐지 시스템의 개요를 보여준다.

본 논문에서는 솔라리스 운영체제의 BSM(Basic Security Module)에서 제공하는 시스템 호출 감사자료를 사용하였다. 전처리 모듈에서는 지정된 프로그램의 실행 여부를 감사하여 각 프로그램별로 시스템 호출을 분리시키고 시퀀스 데이타를 만든다. 여기서 나온 학습데이타를 이용하여 각 프로그램의 정상적인 행동을 신경망으로 학습하여 정상행위 프로파일을 생성한다. 감사대상으로 지정된 프로그램의 수만큼의 신경망이 사용되며 새로운 데이타가 들어온 경우 만들어진 신경망 중 해당하는 것에 입력한 후 정규화 과정을 거쳐 임계값보다 높은 경우 경보를 발생시킨다.

3.1 정상행위 모델링

그림 2는 전체적인 정상행위 모델링 과정을 보여준다. 먼저 임의의 가중치로 초기화한 신경망 집단을 생성한 후 각 개체를 역전파 알고리즘으로 부분 학습시켜서 진화 알고리즘이 주어진 문제에 대해 최적 구조를 찾는 것을 돕는다. 그 후 각 신경망의 적합도를 계산하고 선택 및 유전연산자를 적용해 다음 세대를 생성한다. 이 과정을 종료조건이 만족될 때까지 반복한다. 본 논문에서는 100세대에 도달할 때까지 진화 과정을 반복하였다.

시스템 호출 감사자료를 학습하기 위해 사용된 신경망의 구조는 그림 3과 같다. 신경망의 입력으로는 시간 t에 길이 L의 윈도우로 추출한 시스템 호출 시퀀스 S_t가 사용되기 때문에 입력층은 L개의 노드를 가지며 출력층은 각각 침입, 정상을 나타내는 2개의 출력노드로 이루어져 있다. 본 논문에서는 윈도우 길이를 10으로 사용하였으므로 10개의 입력노드를 가진다. 은닉 노드는 모두 15개가 사용되었는데 노드들 사이의 연결 관계는 진화 알고리즘을 이용해서 설정된다. 비정상행위 탐지는 정상행위만을 모델링하지만 신경망 학습에는 침입, 정상 두 가지 클래스의 데이타가 모두 필요하기 때문에, 침입

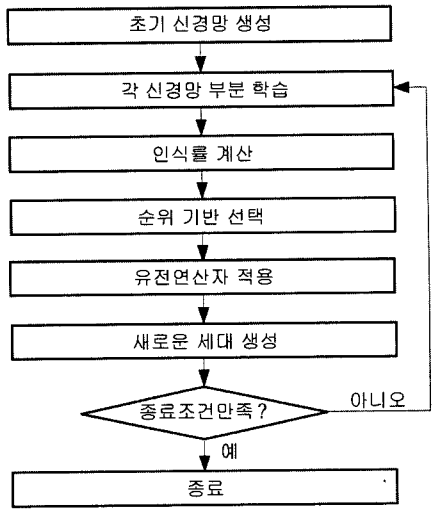

그림 2 정상행위 모델링과정

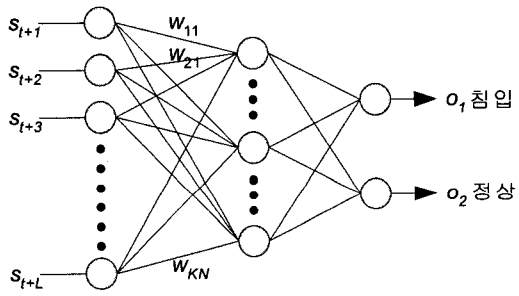

그림 3 사용된 신경망의 구조

클래스의 데이타를 만들기 위해 랜덤하게 시퀀스를 생성하였다. 학습데이타에는 정상적인 프로그램의 시스템 호출 시퀀스와 침입 클래스로 간주되는 랜덤하게 생성된 시퀀스가 1:2의 비율로 존재하며, 이로써 주어진 정상적인 시스템 호출 시퀀스이외의 것은 침입으로 탐지하는 신경망을 만들 수 있다.

일반적으로 진화 알고리즘을 실제 문제에 적용하기 위해서 해결해야할 문제들은 3가지로 요약 될 수 있다. 먼저 풀어야할 문제를 어떻게 유전자형으로 표현할 것인가에 대한 표현의 문제, 표현방법과 문제에 적합한 유전연산자 정의, 마지막으로 각 개체에 대한 적합도 평가 방법이다. 본 논문에서는 다음과 같은 방식으로 진화알고리즘을 신경망 학습에 적용하였다.

1) 표현

행렬기반의 신경망 표현방법을 사용하였다. N개의 노드를 가진 신경망은 $N \times N$ 크기의 정방행렬에 연결 정

보와 가중치를 동시에 표시하여 나타내진다. 행렬의 우상단은 노드간 연결 정보를 1과 0으로 표시하고, 각 연결 정보에 대칭하는 좌하단은 가중치를 나타낸다. 그림 4는 4개의 노드와 4개의 연결로 구성된 신경망의 행렬기반 표현의 예를 보여준다. 행렬기반 표현방식은 간단하면서도 N개의 노드를 가지는 모든 경우의 신경망을 표현할 수 있고 유전연산자를 적용하기 쉬운 장점이 있다.

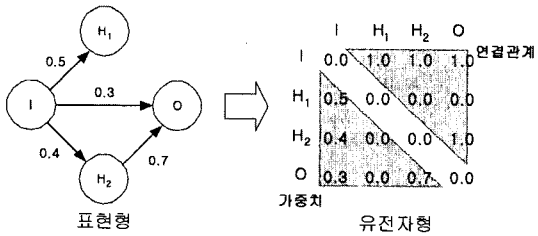

그림 4 신경망의 표현

2) 유전연산자

진화 알고리즘은 다양한 선택방법을 제공하고 있다. 일반적으로 많이 사용되는 방법은 적합도의 크기에 비례하여 선택확률을 부여하는 것이다. 이 방법은 개체 사이의 적합도 차이가 지나치게 클 경우 하나의 해로만 빠르게 수렴해 버리는 문제점이 있다. 이러한 문제점을 해결하기 위해 순위기반 선택 방법을 사용했다.

교차연산은 두개의 신경망을 교차하여 새로운 자손을 만들어 내는 방법이다. 그림 5는 교차연산의 예를 보여준다. 임의의 은닉노드를 하나 선택한 후 그 노드를 중심으로 두 신경망의 구조를 교환하는 것이다. 그림 5에서 회색노드를 중심으로 연결 구조가 교환된 것을 볼 수 있다.

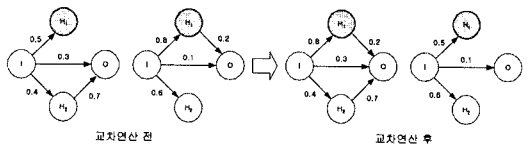

그림 5 교차 연산

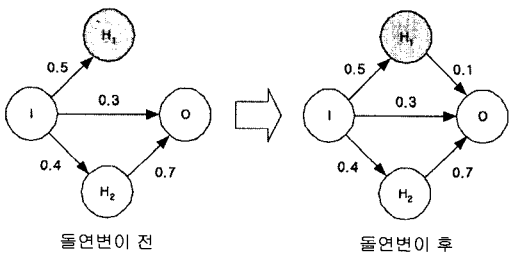

그림 6 돌연변이 연산

돌연변이 연산은 추가/삭제의 두 가지 형태로 이루어진다. 임의로 선택한 연결이 이미 존재하는 경우 그 연결은 삭제된다. 연결이 존재하지 않는 경우에는 새로운 연결을 생성하고 가중치를 0과 1사이의 임의의 실수로 결정한다. 그림 6은 돌연변이 연산의 예를 보여준다.

3) 적합도 평가

각 신경망을 학습 데이타로 테스트해 옳게 분류한 샘플이 많은 신경망이 높은 적합도를 가지도록 인식률을 사용하여 적합도를 계산하였다.

3.2 비정상행위 탐지

신경망은 한 개의 시퀀스 S_t에 대한 정상, 비정상 여부를 제공하지만 프로세스 전체의 침입여부, 즉 P의 정상, 비정상 여부를 판단해야한다. 그림 7은 시간 60에서 루트권한을 얻은 침입에 대한 신경망 평가값의 변화를 나타낸다. 침입이 성공하는 시간 60 근처의 (c)구간에서 비정상노드의 값이 급격히 증가하고 한동안 지속되는 것을 관찰할 수 있다. (a), (b), (d)구간에서도 비정상 출력 노드의 값이 정상 출력 노드의 값보다 잠시 커지지만 이것만으로 침입여부를 판단하기에는 부족하다. 따라서 (c)구간과 같은 연속적인 비정상적인 시퀀스를 탐지하는 것이 중요하다. 그러기 위해서는 현재 시퀀스의 평가 값 뿐만 아니라 이전 시퀀스의 평가 값도 같이 반영하는 것이 필요한데 본 논문에서는 이를 위해 다음과 같은 방법으로 시퀀스의 평가값을 결정하였다. o_t^1은 침입을 나타내는 출력 노드의 값, o_t^2는 정상을 나타내는 출력 노드의 값, w_1, w_2, w_3는 각 값의 가중치를 나타낼 때 시간 t의 시퀀스의 평가 값 r_t는 다음과 같은 식에 의해 결정된다.

$$r_t = w_1 \cdot r_{t-1} + w_2 \cdot o_t^1 + w_3 \cdot o_t^2$$

이 식에 의해 과거 시퀀스의 평가 값의 영향은 점점 감소되고 비교적 최근의 것이 더 많은 영향을 주게되며 또한 침입노드의 출력 값이 높고 정상노드의 출력 값이 낮을 수록 높은 평가 값을 얻는다. 이렇게 하면 (c)와 같은 지역적이며 연속적인 평가 값 변화 추이를 일시적인 변화와 구별해 낼 수 있게 된다.

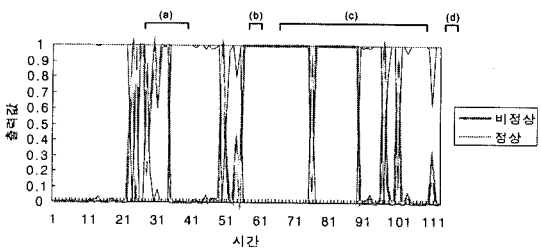

그림 7 신경망의 평가값 변화

최종적인 침입여부의 판단은 r_t값에 대한 임계값을 설정하여 임계값 이상일 경우 침입으로 결정해야한다. 하지만 각 프로그램별로 다른 신경망이 사용되므로 입력값의 변화에 따른 평가값의 민감도가 달라 침입과 정상행위의 결정경계가 모두 다르다. 따라서 하나의 임계값을 전체 신경망 집합에 사용하는 것은 문제가 있는데 본 논문에서는 이를 해결하기 위하여 통계적인 방법을 사용하여 각 신경망의 평가 값을 정규화하였다. 먼저 각 신경망의 평가 값은 정규분포를 따른다고 가정하고 학습 데이타에 대한 평가 값의 평균과 표준편차를 구한 후 테스트 데이타의 평가 값을 표준정규분포상의 값으로 변환하였다. 변환된 값이 정해진 임계치를 넘을 경우 그 프로세스는 침입으로 판단된다. m은 학습데이타 평가 값의 평균, d는 표준 편차라 할 때 정규화 된 평가값 R_t는 다음과 같이 계산된다.

$$R_t = eval\ (S_t) = \frac{R_t - m}{d}$$

이렇게 계산된 R_t'값이 정해진 임계값을 넘으면 해당 프로세스는 침입으로 간주된다.

4. 실험 및 결과

4.1 실험 환경

BSM 감사자료에는 약 280여개의 시스템 호출 이벤트가 들어있다. 하지만 이를 모두 다 사용할 경우 문제의 복잡도가 너무 커지므로 46개의 시스템 호출로 축약하여 사용하였다. 자주 사용되는 시스템 호출 45개에 0부터 44까지의 값을 할당하고 나머지 시스템 호출은 모두 45로 할당하였다. 이렇게 지정된 0~45까지의 값은 0~1사이로 정규화되어 신경망의 입력으로 사용된다. 사용한 45개의 시스템 호출은 표 2와 같다.

제안하는 기법의 성능을 시험하기 위해 MIT Lincoln Lab에서 제공하는 1999년 DARPA IDEVAL데이타를 사용하였다[13]. 이 데이타는 Denial of Service, probe, Remove-to-local (R2L), User-to-root (U2R)의 4가지 종류의 공격을 담고 있는데 본 논문에서는 프로그램의 오동작을 일으켜 비정상적인 프로그램 행동을 유도하는 U2R공격을 탐지하는데 초점을 두어 실험하였다. 따라서 본 논문에서는 U2R공격의 주된 공격 대상이 되는 SETUID권한을 가지는 프로그램만의 실행을 모니터링 하였다. 1999년 IDEVAL 감사자료에 사용된 호스트의 파일시스템에서 추출된 SETUID 프로그램의 목록은 표 3과 같다.

표 2 사용한 시스템 호출 이벤트

exit	fcntl	ioctl
fork	rename	pipe
creat	mkdir	setuid
unlink	fchdir	utime
chdir	pathconf	setgid
chown	open -read	mmap
access	open - write	audit
stat	open - write,creat	munmap
lstat	open - write,trunc	seteuid
readlink	open - write,creat,trunc	putmsg
execve	open - read,write	getmsg
vfork	open - read,write,crea	auditon
setgroups	close	memcntl
setpgrp	getaudit	sysinfo

표 3 SETUID 프로그램 목록

at	rsh	sendmail	deallocate
atq	su	utmp_update	list_devices
atm	uptime	accton	ffbconfig
chkey	w	xlock	ptree
crontab	yppasswd	ff.core	pwait
eject	volcheck	kcms_configure	ssh
fdformat	ct	kcms_calibrate	sulogin
login	nispasswd	mkcookie	admintool
newgrp	top	allocate	sulogin
passwd	quota	mkdevalloc	whodo
ps	ufsdump	mkdevmaps	pt_chmod
rcp	ufsrestore	ping	rlogin
rdist	exrecover	sacadm	

1999년 IDEVAL 데이타는 총 5주 분량의 감사자료를 제공하는데 그중 1-3주는 학습 데이타이고 4-5주는 테스트 데이타이다. 본 논문에서는 침입이 들어 있지 않은 1, 3주 데이타를 신경망 학습을 위해 사용하였고 4, 5주 데이타로 그 성능을 시험하였다. 테스트 데이타에는 4가지종류의 U2R공격이 11번 수행되었다. 표 4는 테스트 데이타에 포함된 공격의 종류를 보여준다.

유전자 알고리즘의 변수로 신경망 집단의 크기는 20, 교차 확률은 0.3, 돌연변이 확률은 0.08을 사용하였다. 100세대까지 진화시킨 후 가장 높은 적합도를 가지는 신경망을 사용해 테스트 데이타에 적용하였다.

표 4 테스트 데이타에 포함된 공격

이름	설명	횟수
eject	exploiting buffer overflow in the 'eject' program	2
ffbconfig	exploiting buffer overflow in the 'ffbconifg' program	2
fdformat	exploiting buffer overflow in the 'fdformat' program	3
ps	race condition attack in 'ps' program	4

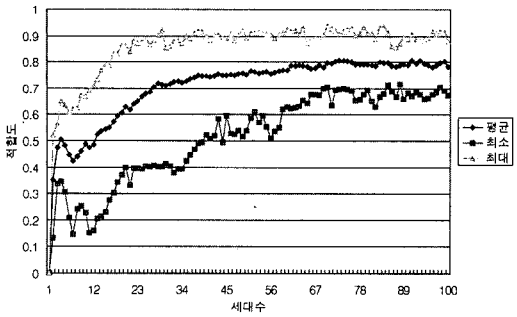

그림 8 신경망의 진화 과정

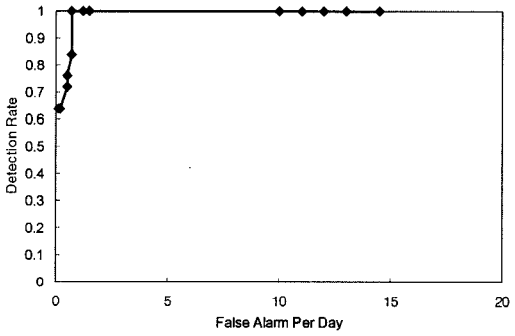

그림 9 제안한 탐지 방법의 성능

4.2 실험 결과

그림 8은 진화신경망의 진화과정을 보여준다. 세대가 거듭될수록 신경망의 적합도가 증가하여 진화알고리즘이 좀더 좋은 신경망의 구조를 찾아낼 수 있음을 확인하였다. 최대 적합도는 0.9에서 수렴하여 학습데이타를 90%정도 인식률로 분류할 수 있음을 알 수 있었다.

일반적인 다층 신경망(MLP, Multi Layer Perceptron)과 진화신경망의 학습시간을 비교하여 보았다. 실험은 Intel Pentium Zeon 2.4GHz Dual 프로세서, 1GB RAM의 하드웨어와 솔라리스9 운영체제 하에서 10번 실행한 후 평균값을 취하였다. MLP의 경우 은닉노드수를 10부터 60개까지 변화시키며 5000세대까지 학습시켰고 진화신경망의 경우 15개의 은닉노드를 가지는 20개체의 신경망을 100세대까지 진화시켜보았다. 사용한 데이타는 login프로그램의 학습데이타로 총 1905개의 시퀀스로 이루어져 있다.

표 5는 실험결과를 나타낸다. 기존의 방법과 같이 각 신경망을 10개씩 학습시킨 후 그중 가장 좋은 것을 선택하는 방법을 쓸 경우 약 17시간 50분이 걸리게 된다. 그러나 진화신경망을 사용한 경우는 약 1시간 14분밖에 걸리지 않았다. 진화알고리즘을 통한방법이 구조를 최적화 시킬 수 있는 장점을 가지면서도 학습 시간면에서도 기존의 방법보다 나은 것을 알 수 있었다.

표 5 진화신경망과 일반적인 MLP의 학습시간 비교

종류	은닉노드 수	소요시간(초)
MLP	10	235.5
	15	263.4
	20	454.2
	25	482
	30	603.6
	35	700
	40	853.6
	50	1216
	60	1615
진화신경망	15	4460

표 6 진화신경망과 MLP의 구조 비교

(a) 진화 신경망

FROM\TO	입력	은닉	출력
입력	0	86	15
은닉	0	67	19
출력	0	0	0

(b) MLP

FROM\TO	입력	은닉	출력
입력	0	150	0
은닉	0	0	30
출력	0	0	0

그림 9에서는 테스트 결과를 1999년 DARPA IDEVAL에서 사용된 탐지/오경보 그래프로 나타내었다. 성능은 상당히 좋았는데 100% 탐지율에서 하루에 평균 0.7개의 false alarm을 나타내었다. 1999년 DARPA IDEVAL 데이타를 사용한 연구 결과 중 U2R공격 탐지에 가장 좋은 성능을 보인 것은 A.K. Ghosh 등의 시스템 호출 감사자료와 Elman 신경망을 사용한 방법이었다[14]. 이 탐지 시스템은 100% 탐지율에서 하루에 3개의 false alarm을 보였다[12]. 이와 비교해 보았을 때 진화 알고리즘을 이용해 신경망의 구조를 결정하는 것이 시행착오를 통해서 결정하는 것보다 더 짧은 시간 안에 효과적인 구조를 찾아낼 수 있음을 확인하였다.

그림 10은 ps프로그램의 행동 학습에 가장 좋은 성능을 보인 진화 신경망의 구조를 보여준다. 일반적인 다층 신경망보다 더 복잡한 구조를 보임을 알 수 있다. 진화신경망은 구조에 제약을 두지 않기 때문에 이처럼 복잡하지만 침입탐지에 최적의 성능을 보이는 구조를 찾아 성능을 향상시켰다.

표 6은 ps프로그램의 행동학습을 위해 100세대 동안 진화시킨 신경망과 일반적인 MLP와 연결 가중치 수를 비교한 결과이다. 각각의 신경망은 모두 10개의 입력 노

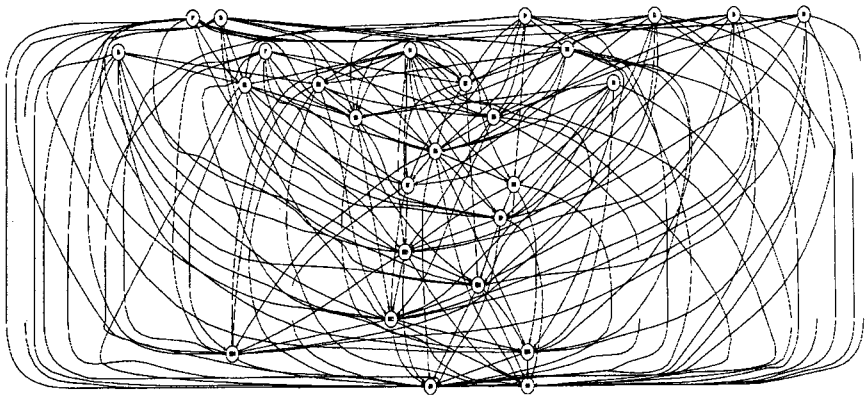

그림 10 가장 좋은 성능을 보인 신경망의 구조

드, 15개의 은닉 노드, 2개의 출력노드를 가진다. 진화 신경망은 총 187개의 연결 가중치를 가지고 MLP는 180개의 연결 가중치를 가져 연결 가중치의 수는 크게 차이나지 않는다. 그러나 진화신경망의 경우는 MLP에는 존재하지 않는 입력노드에서 출력노드로의 연결, 은닉노드에서 은닉노드로의 연결 등 더 다양한 종류의 연결 가중치가 추가되어 더 복잡한 신경망의 구조를 형성한다.

A.K. Ghosh 등의 연구[11,12]에서는 시스템 호출 감사 자료가 시간순서의 시퀀스임을 이용하여 recurrent 연결 관계를 설정하여 샘플간의 상태 정보를 유지함으로써 성능향상을 꾀하였다. 하지만 진화신경망은 이처럼 불규칙적으로 복잡한 연결 구조를 설정함으로써 학습 가능한 시퀀스 종류를 늘려 보다 정확하게 정상행위를 학습할 수 있었다.

5. 결론 및 향후연구

본 논문에서는 프로그램 행위 학습기반 비정상행위탐지를 위하여 진화 신경망을 사용하는 방법을 제안하였다. 제안한 침입탐지 방법은 분류기의 구조와 가중치가 진화 알고리즘에 의해 동시에 학습되므로 기존의 고정된 구조를 사용하는 방법보다 더 좋은 성능을 기대할 수 있다. 1999년의 DARPA IDEVAL 데이타로 실험한 결과 100% 침입탐지율에서 하루에 0.7개의 false-alram을 보여 진화 신경망이 기존의 연구보다 더 좋은 성능을 보임을 확인하였다. 향후 연구로는 진화된 신경망의 구조를 분석하여 침입탐지에 좋은 구조가 어떤 것인지 밝혀내는 작업이 필요하겠다. 또한 종분화 방법에 의해 진화된 상호보완적인 다중 신경망을 결합하는 방법을 이용하면 좀 더 좋은 성능을 기대할 수 있을 것이다.

참 고 문 헌

[1] T. F. Lunt, "A Survey of Intrusion Detection Techniques," *Computers & Security*, vol. 12, no. 4, pp. 405-418, June 1993.

[2] X. Yao, "Evolving Artificial Neural Networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423-1447, 1999.

[3] S. Forrest, S. A. Hofmeyr, and A. Somayaji, "A Sense of Self for Unix Processes," *Proceedinges of the 1996 IEEE Symposium on Research in Security and Privacy*, pp. 120-128, 1996.

[4] S. Forrest, S. A. Hofmeyr, and A. Somayaji, "Computer Immunology," *Communications of the ACM*, vol. 40, no. 10, pp. 88-96, 1997.

[5] S. A. Hofmeyr, A. Somayaji, and S. Forrest, "Intrusion Detection Using Sequences of System Calls," *Journal of Computer Security*, vol. 6, pp 151-180, 1998.

[6] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting Intrusions Using System Calls: Alternative Data Models," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 133 - 145, Oakland, CA, May 1999.

[7] W. W. Cohen, "Fast Effective Rule Induction," *Proceedings of the 12th International Conference on Machine Learning*, pp. 115-123, Tahoe City, CA, 1995.

[8] W. Lee, S. Stolfo, and P.K. Chan, "Learning Patterns from Unix Process Execution Traces for Intrusion Detection," *Proceedings of AAAI97 Workshop on AI Methods in Fraud and Risk Management*, pp. 50-56, 1997.

[9] E. Eskin, W. Lee and S. J. Stolfo, "Modeling System Calls for Intrusion Detection with Dynamic Window Sizes," *Proceedings of DARPA Information Survivability Conference and Exposition*(DISCEX II), pp. 165-175, 2001.

[10] N. Ye, X. Li, Q. Chen, S. M. Emran, and M. Xu, "Probabilistic Techniques for Intrusion Detection based on Computer Audit Data," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 31, no. 4,

pp. 266-274, 2001.
[11] A. K. Ghosh, A. Schwartzbard, and M. Schatz, "Learning Program Behavior Profiles for Intrusion Detection," *Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring*, pp. 51-62, Santa Clara, CA, April, 1999.
[12] A. K. Ghosh, C. C. Michael, and M. A. Schatz, "A Real-Time Intrusion Detection System Based on Learning Program Behavior," *Proceedings of the Third International Workshop Recent Advances in Intrusion Detection*, pp. 93-109, 2000.
[13] MIT Lincoln Laboratory, "DARPA Intrusion Detection Evaluation", Available from <http://www.-ll.mit.edu/IST/ideval/index.html>
[14] R. Lippmann, J. Haines, D. Fried, J. Korba, and K. Das, "The 1999 DARPA Off-Line Intrusion Detection Evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579-595, 2000.

한 상 준

2002년 8월 연세대학교 컴퓨터과학과(학사). 2004년 8월 연세대학교 컴퓨터과학과(석사). 현재 삼성전자주식회사 재직중 관심분야는 인공지능, 지능형 에이전트, 유비쿼터스 컴퓨팅

조 성 배

1988년 연세대학교 전산과학과(학사) 1990년 한국과학기술원 전산학과(석사) 1993년 한국과학기술원 전산학과(박사) 1993년~1995년 일본 ATR 인간정보통신연구소 객원 연구원. 1998년 호주 Univ. of New South Wales 초청연구원. 1995년~현재 연세대학교 컴퓨터과학과 정교수. 관심분야는 신경망, 패턴인식, 지능정보처리