98-I-01-03-A-104

국제공동연구개발사업

인공지능 기법을 이용한 전자상거래 신용 평가 및 Frauds Detection 시스템 개발

상명대학교

과학기술부

제출문

과학기술부 장관 귀하

본 보고서를 " 인공지능기법을 이용한 전자상거래 신용평가 및 Frauds Detection 시스템 개발" 과제의 보고서로 제출합니다.

2000. 10.

주관연구기관명	: 상명대학교
주관연구책임자	: 최종욱
연구원	: 유세근
"	: 이원하
"	: 조정석
"	: 신우철
"	: 전태식
"	: 이한호
"	: 김민철
공동연구기관명	: UCL (영국)
공동연구책임자	: Philip Teleaven
연구원	: 김정원
참여기업명	: 시큐리티테크놀로지
참여기업 책임자	: 최환석

요약문

I. 제목

인공지능기법을 이용한 전자상거래 신용평가 및 Frauds Detection 시스템개발

II. 연구개발의 목적 및 필요성

○ 인터넷의 확산에 따르는 전자상거래가 향후 우리나라의 중요한 사회적 변화로 떠오를 전망이나 전자상거래가 이루어지기 위해 필요한 고객의 신용평가와 인증, 신용카드나 네트워크 접속 사용자 번호(ID Number)를 불법으로 사용하는 Frauds의 발견 기술을 확보하고 있지 않다. 이들 3가지 기술은 향후 전자상거래가 활성화하기 위한 필수조건이라고 할 수 있다. 우선 판매자의 입장에서는 타인의 신용카드나 네트워크 접속 번호를 불법으로 사용하는 고객 Fraud를 방지할 수 있는 통신망 보호대책이 필요하다. 한편 고객의 입장에서는 위장 거래업체를 차려 놓고 고객을 유혹하는 사기업체로부터의 보호를 받을 필요가 있으며 자신의 신용카드나 인터넷 접속번호를 도용한 상거래를 방지하기 위한 Frauds Detection 기술이 필요하다. 물건이나 서비스를 구매하려는 소비자와 이를 제공하는 공급자의 신용도와 신분확인은 어느 쪽에서도 필요한 정보이며 기술이다. 최근 이러한 Frauds방지 문제는 각국이 인터넷 홈페이지에 대한 인증(e-Trust Mark)을 실시하여 방지하고 있다. 또한 각국에서는 소비자 보호원과 같은 소비자 단체들이 고객 불만과 홈페이지 감시를 통해 해결해 나가고 있다.

○ 전자상거래 시스템의 활성화를 위해 필요한 이러한 기술들은 국내에서 개발되거나 축적된 경험이 없어 이러한 기술이 이미 확보되어 있거나 개발 중인 선진국의 연구기관과 손잡고 이를 공동연구로 수행할 필요가 있다. 전자상거래의 전제조건인 소비자/공급자의 신용평가 및 인증기술은 이미 선진국에서는 안정적으로 사용하고 있는 기술이나 이를 국내의 소비자나 단체의 거래특성에 맞도록 수정 보완하여 도입할 필요가 있다. 그리나 본 연구를 공동으로 진행하고 영국의 University College London(UCL)에서는 초기의 internet frauds detection’에서 연구 주제를 ‘intruder detection’으로 바꾸도록 요청하였으므로 본 연구에서는 ’intruder detection system(IDS)의 연구를 집중적으로 진행하였다.

○ 본 연구개발에서는 오랫동안 전자상거래 분야의 기술을 축적해 온 University College London(UCL)과 손잡고 안전한 전자상거래에 필요한 기술인 financial fraud detection and network intrusion detection에 관한 기술을 공동 연구 개발하였다. 특히 본 연구에서는 기존의 인공지능 알고리즘으로는 해결이 어려운 것으로 알려진 데이터의 전처리문제, 알고리즘의 일반화 문제를 다루고, 개발된 시스템에 의한 예측 결과의 이해도를 높이며, 분산형에 적합한 새로운 구조 등을 개발하는데 그 연구의 중점을 두었다.

Ⅲ. 연구개발의 내용 및 범위

년 도	목 표	개 발 내 용
1차년도 (1997-1998)	개인 신용평가 모델(Financial Fraud Detection) 초기 알고리즘 개발 침입 탐지 시스템 (Intrusion Detection System) 현장조사 및 데이터 수집	1. 개인 신용평가 모델 초기 알고리즘 개발 - 클러스터링 알고리즘을 이용한 개인 신용카드 데이터 수집 및 전처리 - 퍼지 전문가 시스템 구축 - 유전자 알고리즘을 이용한 퍼지 규칙 학습 알고리즘 구현 2. 침입 탐지 시스템(Intrusion Detection System) 현장조사 및 데이터 수집 - 현존 침입 탐지 시스템 분석 - 네트워크 데이터 수집 및 전처리 - Automated Profiler 개발
2차년도 (1998-1999)	개인 신용평가 모델(Financial Fraud Detection) 알고리즘 개발 새로운 네트워크 침입 탐지시스템(Network-Based Intrusion Detection System) 프레임 웍 개발	1. 개인 신용평가 모델(Financial Fraud Detection) 알고리즘 개발 - 다양한 새 퍼지 멤버쉽 함수 개발 - 노이즈 데이터 처리를 위한 multi-objective fitness function 개발 - 다양한 변수의 자동 조절을 위한 committee members 개발 2. 새로운 침입 탐지 시스템(Network-Based Intrusion Detection System) 프레임 워크 개발 - 인간 면역 시스템의 연구 - 새로운 침입 탐지 시스템 모델인 인공 면역 시스템 개발 - Negative Selection Algorithm 개발
3차년도 (1999-2000)	시스템 통합, 성능 테스트	1. 개인 신용평가 모델(Financial Fraud Detection) 평가 - 실 데이터를 이용한 시스템 평가 2. 인공 면역 시스템 평가 및 추가 개발 - Negative Selection Algorithm 평가 - Clonal Selection Algorithm 개발

Ⅳ. 연구개발결과

3년에 걸친 본 연구의 연구수행 내용 및 결과는 다음과 같다. 우선 본 연구는 개인 신용 평가 시스템의 개발과 새로운 지능형 네트워크 침입 탐지기를 개발하는 것을 주요 목표로 연구가 수행되었다.

가. 지능형 개인 신용 평가 시스템의 개발

그동안의 많은 개인 신용 평가 시스템이 신용 스코어링 테이블, 통계적인 방법, 신경망 알고리즘등을 이용해 왔고, 신용 평가 예측력을 높이기 위해 일반화에 관한 연구가 진행되어 왔다. 본 연구에서는 이러한 기존 방법들과는 다른 신경망 퍼지 규칙의 진화를 이용하여 지능형 개인 신용 평가 시스템을 개발하였다.

○ 퍼지 전문가 시스템에서는

- 다양한 클러스터링 알고리즘을 이용한 퍼지 규칙의 정의하고, 그에 따른 신용 평가 예측도와 신용 평가 퍼지 규칙의 이해도를 평가
- 다양하고 새롭게 정의된 퍼지 멤버쉽 함수를 이용하여 퍼지 규칙을 정의하고, 그에 따른 신용 평가 예측도와 신용 평가 퍼지 규칙의 이해도를 평가

이상의 연구 내용들이 수행되었고,

○ 유전자 알고리즘을 이용한 퍼지 규칙 진화 시스템에서는

- 유전자 프로그래밍의 새로운 교차(Crossover)오퍼레이터의 개발,
- 개발된 신용 평가 시스템이 하나 이상의 다중 목표를 만족하도록 하는 적합함수(fitness function)의 구현
- 유전자 프로그래밍의 진화 과정이 불량 신용 고객의 패턴을 구성하는 여러개의 퍼지 규칙을 모두 찾을 수 있도록 유도하는 nested genetic search의 구현
- 많은 변수들의 서로 다른 조합이 만들어 내는 다양한 결과중 가장 이상적인 시스템의 판단결과를 선택하기 위한 위원회 결정(committe-decision)의 구현

이상의 연구 내용들이 수행되었다.

○ 개발된 시스템은 Lloys/TSB가 제공한 보험 고객 데이터와 국내 카드 회사에서 제공한 개인 신용 카드 고객 데이터에 적용하여 시스템의 성능이 평가되었으며, 노이즈가 많은 실제 데이타에도 불구하고, 높은 신용평가 예측력과 이해도를 보였다.

나. 새로운 네트워크 침입 탐지 시스템

본 연구에서 제안한 네트워크 침입 탐지 시스템은 불법 네트워크 침입자의 네트워크 침입시

정상적인 네트워크 트래픽에서는 관찰되지 않는 다른 패턴의 네트워크 트래픽을 탐지하여 네트워크 관리자에게 불법 침입자의 침입 가능성을 자동으로 알리는 시스템이다. 이러한 지능적 네트워크 침입 탐지 시스템의 연구 개발은 비교적 새로운 연구 분야로서, 대부분의 시도되고 있는 연구들은 중앙 집중적인 통계적 방법들을 이용하고 있다. 본 연구에서는 수행된 연구로는

○ 전혀 새로운 네트워크 침입 탐지 시스템인 인공 면역 시스템을 제안

- 현존하고 있는 네트워크 침입 탐지 시스템들의 중앙 집중적인 통계적 방법들의 한계를 극복
- 이러한 한계들을 극복하기 위한 전혀 새로운 접근법의 제안을 위한 인간 면역 시스템을 연구
- 인간 면역 시스템의 연구를 통해 인간 면역 시스템만이 갖고 있는 특징들을 네트워크 침입 탐지 시스템에 적용할 수 있는지의 여부와 그러한 적용이 이미 본 연구에서 지적된 종전의 중앙 집중적인 네트워크 침입 탐지 시스템들의 한계를 극복할 수 있는지를 연구
- 이상의 연구를 통해 전혀 새로운 네트워크 침입 탐지 시스템인 인공 면역 시스템을 제안

○ 인공 면역 시스템을 구성하는 주 네트워크 침입 탐지 시스템의 구현

- 정상적인 네트워크 트래픽과 불법 네트워크 침입자의 네트워크 침입시의 네트워크 트래픽의 차이를 구분할 수 있는 네트워크 트래픽 특징들을 선별
- 선별된 네트워크 트래픽 특징들을 모아진 네트워크 패킷에서 추출하여 트래픽의 profile을 만드는 자동네트워크 트래픽 profiling 모듈 구현
- 정상적인 네트워크 트래픽 패턴을 불법 네트워크 침입 패턴으로 잘못 인식하는 것을 막기위한 부정적 선택(Negative Selection) 알고리즘 구현
- 구현된 부정적 선택(Negative Selection) 알고리즘을 실 네트워크 트래픽 데이터에 적용해보고 그의 한계점 연구

○ 인공 면역 시스템을 구성하는 부 네트워크 침입 탐지 시스템의 구현

- 불법 네트워크 침입이 이미 알려진 네트워크 트래픽에서 그들의 패턴을 추출하여, 미래에 동일하거나 유사한 네트워크 침입 발생시 그의 탐지를 실시간내에 할 수 있도록 하는 복제적 선택(clonal selection) 알고리즘의 구현
- 복제적 선택(clonal selection) 알고리즘의 수행중 생성되는 탐지기(detector)의 일반화(generalization)를 높이기에 적합한 탐지기 표현에 관한 연구
- 부정적 선택(Negative Selection) 알고리즘을 복제적 선택(clonal selection) 알고리즘의 부분으로 통합시키는 것으로 부정적 선택(Negative Selection) 알고리즘의 본래의 목적을 달성하면서, 긴 computational time을 요구하는 부정적 선택(Negative Selection) 알고리즘의 한계를 극복

이상과 같이 본 연구는 기존의 네트워크 침입 탐지 시스템을 유사하게 구현하거나 확장하는 식의 현존기술의 국내화 보다는 연구과정 중 인식된 현존하는 네트워크 침입 탐지 시스템들의 근본적인 문제들을 해결하는데 그 중점을 두었다. 따라서, 국내-외 최초로 통합적인 모델로서의 인공 면역 시스템을 네트워크 침입 탐지 시스템의 새로운 대안으로 제시하였으며, 그에 대한 기본적인 부분의 구현과 평가가 실시되었다.

Ⅴ. 연구개발결과의 활용계획

(1)개발 기술의 국내 확산: 본 연구에서 개발된 신용평가 모델과 기법은 금융기관, 시스템 통합 업체(SI), 신용평가 기관 기업에서 반드시 필요한 기술로서 학회와 학술지 발표를 통해 국내 기관에 기술을 이전 전수하였다. 특히 신용평가는 현재 각 금융기관이 생존을 위해 필요로 하는 기술로서 수 차례의 세미나에서 상당한 평가를 받은 바 있다. IMF사태의 원인이 우리나라 금융기관들의 부실한 대출관리에 있고 현재 일본이 엄청난 무역 흑자에도 불구하고 대외적인 신용도가 낮고 주식 시장이 불황에 빠져 있는 것은 금융기관의 엉성한 신용평가 및 대출 관리 시스템 때문인 것으로 분석되고 있다. 따라서 향후 한국의 금융기관이 가장 필요한 것은 객관적인 예측력이 높은 신용평가 시스템이라고 하겠다. 이러한 상황에서 본 연구에서 진행된 신용평가 기술 개발은 상당한 의미를 갖는다 하겠다.

다음으로 Intruder Detection System은 2000년 들어와서 국내의 기술이 개발되기 시작하고 있으나 아직은 초보적인 수준이며 해외 기술을 따라잡기는 역부족인 것으로 평가되고 있다. 이는 아직 국내의 경우 대부분이 Firewall을 진화시킨 기술로 인식하고 있은 데다 현재까지 쌓여진 자료가 부족하여 지능형 시스템을 개발하기에는 부족한 편이다. 본 연구에서 개발된 면역학 기반의 외부침입 탐지 시스템은 현재의 모든 기술이 가지고 있는 모든 단점을 극복할 수 있는 기술로 인식되고 있다. 이미 여러차례 세미나와 학술대회를 통해 개발된 기술을 발표하였으므로 본 기술이 국내 산업계에 상당한 도움이 되었을 것으로 믿는다.

(2)상용화 기술이전: 공동으로 참여한 (주)Security Technologes International (www.stitec.com)에서는 보안 시스템의 개발에 본 연구에서 개발된 기술을 상당부분 수용하고 있어 상용화 측면에서 상당한 성과가 있었다. 현재 이 회사에서는 암호화 관련 서비스를 하고 있으나 금융 솔루션 제품과 기타 제품(J/LOCK, J/CAS, J/SSLOCK, J/SecureSession)에 개발된 기술을 채용하고 있다.

SUMMARY

With the explosive growth of the Internet users, the sales amount in electronic commerce market is rapidly increasing. However, the vulnerable structure of various e-commerce environments have feared many users. To provide safer e-commerce environment, this research is focused on two sub-goals: the development of financial fraud detection system and the development of network-based intrusion detection system. These two goals are interrelated to each other to guarantee safe infrastructures for e-commerce. However, the nature of each problem is different and thus various techniques are required to implement an individual system. For the first problem, the evolutionary fuzzy rule evolver was developed and evaluated by being applied on TSB/Lloys home insurance data and domestic credit card transaction data. On the other hand, as a new framework of network intrusion detection system, the artificial immune system was proposed and its basic components were developed and tested.

The evolutionary fuzzy rule evolver developed for financial fraud detection contains many novel elements, including a crossover operator designed to minimise disruption, binary genotype, and a new method for interpreting fuzzy rules designed to preserve all fuzzy set membership values. Consultation with Lloyds/TSB resulted in a set of evaluation criteria for the system: intelligibility, speed, handling noise and accuracy. With these aspects in mind, three sets of experiments were performed on the system, using two standard data sets to permit comparison with the literature. The first test investigated the effect of membership functions on the system. The second test investigated the effects of using different clusterers in the system. The final test investigated the ability of the system to cope with increasing levels of noise in the data. These experiments show that many factors affect accuracy of classification, intelligibility and processing speed only seem to be affected by the type and use of membership functions - noise and the choice of clusterer seems unimportant and noisy data causes at best a linear drop in accuracy, and at worst, a fall proportionate to the square of input noise.

As the second stage, this research has described the use of genetic programming to evolve fuzzy rules within a parallel committee decision system. Attention was paid to data preprocessing, describing some of the typical problems associated with real-world data in order to show just how hard this kind of classification becomes. Nevertheless,

despite having only 49 suspicious items in the first class to train the system, and an unknown number of suspicious items in the 10000-item second class, performance of the system was good. Given the quality and quantity of the data, accuracy rates of over 60% must surely be regarded as impressive. Indeed it seems very likely that better accuracy would only result in overfitting the meagre training data. In addition, intelligibility rates were excellent with many rule sets comprising a single, understandable rule. This work shows the benefit of committee decision making. Each of the four different committee members (different setups of the evolutionary fuzzy system) provided different rates of accuracy and intelligibility. The committee decision maker was able to analyse all results and pick the best rule set.

Finally, a committee-decision-making evolutionary fuzzy system developed in this work was applied for domestic credit card transaction data evaluation. The results for this real-world problem confirm previous results obtained for real home insurance data. They illustrate that the use of evolution with fuzzy logic can enable both accurate and intelligible classification of difficult data. The results also show the importance of committee-decision making to help ensure that good results will always be generated.

For the second problem which is network intrusion detection, this research has investigated the existing network-based IDSs extensively and provided a set of general requirements for them by a careful examination of the literature. Based on these requirements, three principal design goals were identified. After sketching the simplified human immune system, their salient features that can contribute to build a competent network-based intrusion detection system were analysed. This analysis show that the human immune system is equipped with a number of sophisticated mechanisms, which satisfy the three identified design goals. Consequently, the design of a novel network-based IDS based on the human immune system is promising for future network-based IDSs.

This research also investigated the existing network-based IDSs. They were categorised into three different approaches: monolithic, hierarchical and co-operative and problems were identified for each approach. In order to resolve these problems, a novel artificial immune model was presented. This model combines the three evolutionary stages: gene library evolution, negative selection and clonal selection into a single methodology. These three processes are co-ordinated across a network to satisfy the three goals for designing effective IDS's: being distributed, self-organizing and lightweight. Analysis of the characteristics of this unified evolutionary approach show that, unlike existing approaches, the proposed artificial immune model does satisfy the requirements of network-based IDSs. Consequently, algorithms based on this model show considerable promise for future IDSs.

A network-based IDS utilizing the artificial immune model is being implemented in order to prove the validity of this approach. Current work is focusing on building initial self profiles and detectors from normal and abnormal TCP/IP packets, which were collected from a real network environment. As the first attempt of this effort, the negative selection stage was implemented and experiments showed its infeasibility for its application to the essential profiling fields of real network data. This result directs this research to re-define the role of negative selection algorithm within the overall artificial immune system framework. Finally, the intrusion detection mechanism of clonal selection stage were investigated and the clear understanding of task of clonal selection stage helped us to comprehend the distinct job of negative selection stage.

The contributions of this work will provide an applicable methodology for designing an artificial immune system to be able to perform network intrusion in a truly distributed, self-organizing and lightweight way.

목 차

제1장 서론

인터넷의 폭발적인 증가는 최근 각국이 정부 최우선 과제로 경쟁적으로 추진하고 있는 초고속 정보통신망 사업에 의해 더욱 가속화되고 있다. 미국의 Information Superhighway, 일본의 신사회 간접자본 프로젝트, 캐나다의 Canarie프로젝트, 싱가포르의 Intelligent Island 2000 프로젝트 등을 통해서 대용량 고속의 광통신과 무선통신망이 구축되고 있으며 위성을 이용한 고속백본망이 연결되면 인터넷의 보급은 더욱 가속화될 것으로 예상된다. 이러한 초고속망의 보급과 인터넷의 확산은 전자상거래를 일반화에 크게 기여하고 있으며 향후 B2B, B2C, P2P 등의 많은 거래가 네트워크를 통해서 이루어질 것을 의미한다.

전자상거래(Electronic Commerce: EC)는 가상점포나 가상 백화점을 만들어 신용카드, 혹은 IC카드를 사용하여 상품이나 서비스를 일반 소비자에게 판매하는 협의의 점포개념(B2C)으로부터 CALS와 Internet-EDI시스템에 바탕을 두는 기업간(B2B)의 비즈니스까지를 포함하는 포괄적인 개념이다. 그런데 EDI의 보급수준에서 논의되던 전자상거래가 인터넷의 급속한 보급 확산으로 예상보다 빠른 속도로 일반인들에게 다가오고 있다. 인터넷 전자상거래 시장은 미국의 경우 매년 40%정도의 성장을 계속하고 있는데 2000년대 후반에는 전체 소비자시장을 석권하게 될 것으로 예측하고 있다. 예를 들어, 1994년 미국의 신용카드 사용액은 9조 3,000억, 인터넷 상거래 액수는 고작 10억 달러에 불과하였으나 2000년에는 신용카드가 16조 5,000억, 인터넷 거래액이 7조 달러로 증가하고 2005년이 되면 인터넷 상거래 시장이 17조 달러에 달하여 신용카드 시장을 앞지르게 될 것으로 미국의 킬른 어소시에이트사는 예측하고 있다.

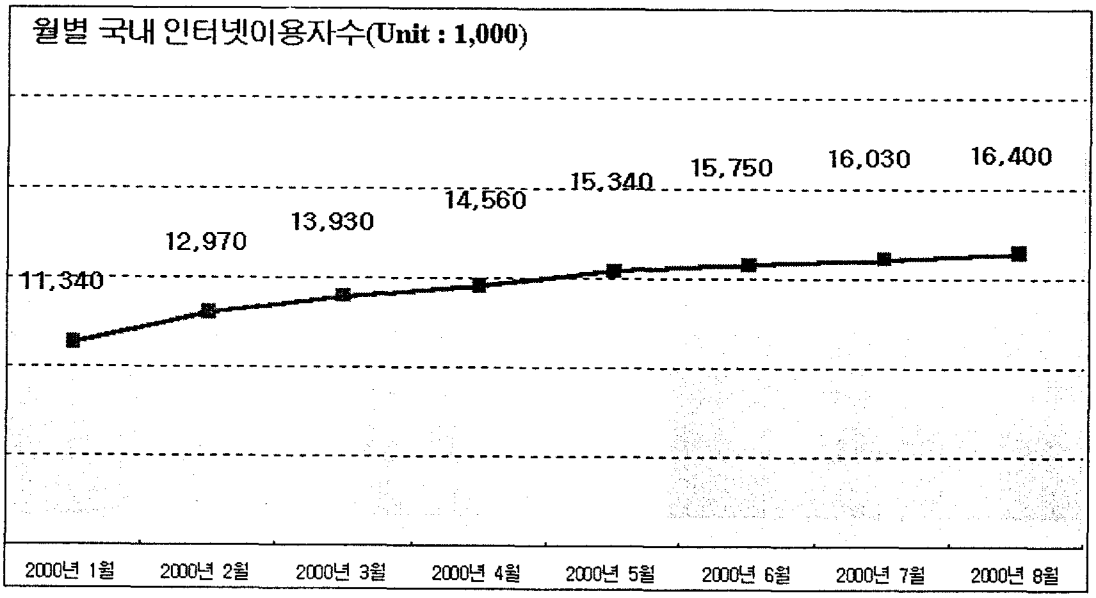
한국 인터넷 정보센타(KRNIC)의 조사에 따르면 2000년 8월말 현재 1,600만 명에 달하는 것으로 나타났다. 인터넷 사용자는 94년 12월 13만 8천명에서, 97년 12월 160만명으로, 그리고 99년 12월에는 10만명 수준으로 증가하였다. 호스트 증가면에서도 96년 12월에 73,191개이었던 것이 불과 3달 사이에 (97년 2월) 이미 81,118개로 늘어나고 2000년 6월에는 4만 3천개로 늘어났다. 이처럼 급팽창하고 있는 인터넷 시장에서 현재 데이콤 인터파크, 롯데 백화점, 현대 백화점, 미도파, 삼성 물산, 신세계, 한솔유통, 사이버 랜드, 파워넷, KTNET, 사이버 코리아, 인터피아, 인포넷, 영우소프트 등의 업체들이 서비스를 제공하고 있으나 2000년 8월 B2B의 경우 3000억원, B2C의 경우 2900억 원에 달하고 있다. 이는 1999년 700억 원과 1300억 원에 비해 각각 300%이상 성장한 수치이다. 가장 괄목한 성장을 하고 있는 분야는 인터넷 증권거래로서 99년 2월 3조2천 668억 원이던 거래액이 99년 12월 91조 9,199억

원으로 늘어났다.

이처럼 인터넷의 확산에 따르는 전자상거래가 향후 우리나라의 중요한 사회적 변화로 떠오를 전망이나 전자상거래가 이루어지기 위해 필요한 고객의 신용평가와 인증, 신용카드나 네트워크 접속 사용자 번호(ID Number)를 불법으로 사용하는 Frauds의 발견 기술을 확보하고 있지 않다. 이들 평가 및 인증 기술은 향후 전자상거래가 활성화하기 위한 필수조건이라고 할 수 있다. 우선 판매자의 입장에서는 타인의 신용카드나 네트워크 접속 번호를 불법으로 사용하는 고객 Fraud를 방지할 수 있는 통신망 보호대책이 필요하다. 한편 고객의 입장에서는 위장 거래업체를 차려놓고 고객을 유혹하는 사기업체로부터의 보호를 받을 필요가 있으며 자신의 신용카드나 인터넷 접속번호를 도용한 상거래를 방지하기 위한 Frauds Detection 기술이 필요하다. 물건이나 서비스를 구매하려는 소비자와 이를 제공하는 공급자의 신용도와 신분확인은 어느 쪽에서도 필요한 정보이며 기술이다.

Domestic Internet User(2000.1~2000. 8)

Date : 2000.8.31
Source : KRNIC



Korea Network Information Center

그러나 최근 이러한 인터넷 사기에 대해서는 소비자 보호 단체와 정부가 e-Trust Mark를 계획하고 있으며 한국에서는 Commerce-Net Korea가 주도적으로 이를 시행하기 위한 계획을 수립하고 있다. 따라서 본 연구에서는 인터넷 사용자들을 위한 신용평가와 네트워크 침입 탐지 기술을 개발하였다. 이는 공동 연구기관인 UCL(University College London)이 Frauds Detection 연구를 98년에 중단하고 Intruder Detection System기술 개발로 연구 주제를 옮겨갔으며 이 분야 역시 국제 협력이 필요한 분야이기 때문이다.

전자상거래 시스템의 활성화를 위해 필요한 이러한 기술들은 국내에서 개발되거나 축적된 경험이 없어 이러한 기술이 이미 확보되어 있거나 개발 중인 선진국의 연구기관과 손잡고 이를 공동연구로 수행할 필요가 있다. 전자상거래의 전제조건인 소비자/공급자의 신용평가 및 인증기술은 이미 선진국에서는 안정적으로 사용하고 있는 기술이나 이를 국내의 소비자나 단체의 거래특성에 맞도록 수정 보완하여 도입할 필요가 있다. 그리고 인터넷에서의 Mall운영이나 서버 운영자들에게는 외부침입을 차단하고 자원을 안전하게 지키는 일이 가장 중요한 일 중의 하나이다. 이러한 외부침입 탐지기술은 현재 미국과 영국의 일부기관에서 연구개발중인 기술이므로 공동 연구를 통해서 조기에 확보하여 향후 전자상거래를 보호할 필요가 있다.

본 연구개발에서는 오랫동안 전자상거래 분야의 기술을 축적해 온 University College London(UCL)과 손잡고 안전한 전자상거래에 필요한 기술인 financial fraud detection and network intrusion detection에 관한 기술을 공동 연구 개발하였다.

제1절 연구개발의 필요성: 기술적 측면

-미래의 첨단 금융기술의 확보: 1997년 1월말에 부도처리된 한보철강에 대해 한국신용평가(주)와 한국기업평가(주)는 금년 1월의 신용평가에서 지난해 한보철강이 받았던 회사채발행 등급 BBB-를 B+와 A3-로 각각 격상하였던 것으로 보도되고 있다. 이와 반대로 국제적인 신용평가사인 Moody 社의 모회사인 D&B가 발행한 보고서에서는 한보철강이 93년 이후 운전자본이 지속적으로 마이너스 증가하였고 그 폭이 해마다 커지는 점을 중시, - - (Blank)등급을 내렸던 것으로 알려졌다. -(Blank)라는 등급은 D&B가 부여하는 4단계의 평점보다도 낮은 것으로 현재 도산의 위험에 처한 기업에 내리는 평가이다. 국내에서 영업을 하고 있는 외국 신용평가 기업들과 같은 정보를 가지고 있거나 더욱 근접한 정보를 가지고 있는 국내 금융기관들

의 이 같은 실수는 결국 국내 기관들의 신용평가 기술이 초보 내지는 미숙한 수준이었다는 점을 단적으로 보여주고 있다. 이외에도 1995년과 1996년 동안 증권시장에서 사실과 반대되는 기업평가를 믿고 주식에 투자하였다가 기업의 도산으로 투자자들이 막대한 손실을 보게되어 신용평가 회사들이 정부의 경고와 제제를 받는 것도 평가 기술의 초보수준을 반영하고 있다.

IMF이후에도 한국시장에서는 대우 처리문제와 현대 문제 등 신용평가 및 대출관리에 상당한 기술적인 문제와 대출관리 기술 미흡으로 63조원의 초기 자금 외에도 추가 40조원을 금융산업의 회생을 위해 투입하고 있다. 이는 물론 우리나라 산업전체의 경쟁력이 약화되어 있기 때문에 생겨난 부실의 대가라 할 수 있으나 궁극적으로는 부실한 금융산업의 구조와 관행에 원인이 있었던 것으로 풀이된다. 금융기관의 부실이 변조되거나 위조된 기업들의 재무제표와 엉터리 감사 의견서 및 회계법인의 보고서에 그 첫 번째 원인이 있지만 만약 그러한 부실한 기업의 재무제표 자료를 정밀하게 검토하고 회계관행을 감시하였더라면 막을 수 있었을 것이다. 이중에서도 국내 금융기관의 신용평가는 자료의 미흡과 기법의 미숙함 등으로 상당히 낙후되어 있다.

국제적으로는 보편화된 신용평가 기법에 있어서 국내 금융산업체들이 이처럼 낙후되어 있을 뿐 아니라 현재 선진국의 금융기관들이 추진하고 있는 전자상거래를 위한 전자화폐 제도, 거래자 인증, 전자거래의 결제제도, Frauds Detection, 침입자 색출 등의 기술개발 아직도 미진하다. 이는 정보통신 인프라와 기술이 국가의 경쟁력을 결정할 향후 정보화 사회에서의 생존 수단에 관련된 절박한 문제라 하겠다. 즉, 선진국들의 200년 이상 지속되어온 신용사회라는 바탕 위에 전자상거래라는 새로운 질서와 체제를 구축하고 있으나 우리는 아직 신용상거래 축의 구축이 마련되지 않은 터전 위에 전자상거래라는 질서를 도입하여야 하는 어려움에 처해있다. 더구나 선진국들이 초고속 정보통신망이라는 인프라와 오랫동안 쌓여 온 신용평가와 신용결제 노하우 위에 전자상거래라는 새로운 틀을 구축하고 있는데 비해, 우리나라는 초고속 정보통신망이라는 인프라만을 가지고 전자상거래라는 새로운 틀을 운용하면서 신용평가 인증, 전자화폐, 전자화폐 결제 등의 노하우를 익혀야 하는 형편에 놓인 것이다.

-**Frauds Detection 기술** : 신용평가 뿐만 아니라 최근 선진국에서 심각한 문제가 되고 있는 신용카드 Frauds는 향후 전자상거래가 일반화될 경우 우리나라에서도 중요한 사회적 문제로 대두될 것으로 예상하고 있다. 영국에서는 1994년에만 신용카드 Fraud에 의해 5억 파운드의 손실이 발생하였고, 미국에서 의료보험에 사용된

6,500억달러의 경비중의 10%에 해당하는 650억 달러 정도가 Fraud에 의해 사라졌으며 미국 무선 전화의 Fraud에 의해 하루에 100만 달러 정도의 손실이 발생하고 있는 것으로 나타났다[1] 서비스분야에 있어서의 Fraud의 문제는 서비스산업 자체가 양면성을 띄고 있어 대단히 다루기 힘들다. 소비자들에게는 손쉽게 유연한 서비스를 제공하여야 하지만 동시에 Fraud에 의한 서비스 제공자들의 피해를 최소화시켜야 한다는 양면성을 가지고 있다. 또 하나의 문제점은 Fraud문제는 '동적인 성격'을 갖고 있다는 점이다. 즉, Fraud에 대한 어떠한 해결책도 급속하게 낡은 방법이 된다. 따라서 Fraud추적을 위한 새로운 방법들이 동원되고 있는 바, 국내에서도 향후 발생한 문제들을 해결하기 위해 이러한 기술 개발이 필요할 것으로 본다.

-**인공지능 기술의 확보**: 신용평가 기술과 Frauds Detection 기술, 패턴 인식, 신경망 시스템, 케이스기반 추론 시스템(Case Based Reasoning), Agent기술, Genetic Algorithm 등이 적극적으로 동원되고 있다.

예를 들어 미국과 영국, 유럽시장, 심지어 아시아지역의 신용평가 시장에 진출해있는 대부분의 기업들이 신경망과 Genetic Algorithm을 사용하여 시스템을 개발하고 있다[www.nestor.com, 1997; wuecono.wustl.edu/eprints/, 1997; www.cas.lanc.ac.uk, 1997; www. moneyworld.co.uk, 1997; www.wantiggelen.com, 1997; www.ramsearch.com, 1997; www.neuralt. com, 1997; www.financing.hosting.ibm.com, 1997]. Frauds Detection기술 개발은 대부분 인공지능분야에서 개발된 패턴 인식과 패턴 매칭에 의존하고 있다. 특히 신경망 기술과 퍼지 시스템, 전문가 시스템, Genetic Algorithm이 많이 사용되고 있다 [www.nestor. com, 1999; www.-rnks.infomatik.tu-cottbus.de, 1997; www.saic.com/it, 2000; www.c3.lanl.gov, 2000; www.csl.sri.com, 1999; www.senate.gov, 2000; Search Space, 200].

따라서 본연구 개발사업에서 목표로 하고 있는 지능형 신용평가 시스템이나 Intruder Detection 시스템 개발은 단순한 금융기관의 전자상거래 시스템 개발이 아니라 인공지능 기술을 복합적으로 적용하여야하는 기술이다. 예를 들어 전자상거래 시스템에서 거래에 참여하고 있는 구매자와 공급자사이에서 불량거래자를 찾아내기 위해서는(Genetic Algorithm기술적용) 우선 기존 불량 거래자의 패턴을 알고 있어야하고(지식기반 시스템 기술, 또는 전문가 시스템 기술), 불량거래자의 패턴을 학습하고 이를 의심스러운 거래자의 패턴과 비교하여야 하며(신경망 기술, Genetic Algorithm, 패턴인식 기술), 이를 수행하기 위해서는 어느 정도의 차이가 보이는 거

[1] Search Space, 1997.

래자만을 선택하여야한다(Thresholding, 퍼지기술). 이처럼 전자상거래 시스템의 성공적인 구축을 위해서 필요한 인공지능 기술을 공동연구 개발을 통해 도입확보 할 필요가 있다. 그리고 외부침입 탐지 시스템은 인공지능 기술뿐만 아니라 최근 연구되고 있는 면역학 기반의 Life Science 접근 방법이 요구된다는 점에서 첨단화된 연구가 필요하다.

일반적으로 우리나라에서는 단순한 비즈니스 기술, 특히 영업기술정도로 이해되고 있는 마케팅에서의 시장 점유율 계획, 보험에서의 요율 결정 및 고객 평가, 증권 등에서의 주가 예측이나 경기예측, 제조업에서의 생산량 결정과 유통 과정결정 등에 인공지능 기술이 폭 넓게 응용되고 있다. 우리나라에서는 현장에서의 感과 미래에 대한 透視력, 그리고 소문에 완전 의존하고 있는 증권투자를 선진국들은 Technical Trading이라는 극히 과학적이면서 인공지능 기술을 응용한 컴퓨터 거래에 의존하고 있다. 이러한 차이는 결국 우리나라 비즈니스의 주먹구구식 운영이 결국에 가서는 국가와 사회의 생산성을 떨어뜨리는 비효율적인 시스템이 될 수밖에 없다는 점을 의미한다. 따라서 향후 열리게 될 전자상거래에서의 기술 축적과 발전을 위해서는 이 분야에서 활용되고 있는 인공지능 기술을 적극 도입, 개발할 필요가 있다.

제2절 연구 개발의 필요성: 경제. 산업적 측면

-**부실 대출에 의한 금융비용의 증가**: 1996년 6대 시중은행이 발표한 부실여신액은 1조5천7백84억 원으로 총 대출금의 1%수준이지만 일본과 같이 6개월 이상 연체된 대출금을 포함하면 불량 대출금은 8조3천2백27억 원으로 전체 대출금의 5.5%에 달한다. 그런데 미국과 같이 3개월 이상 이자가 연체된 대출금을 포함시키는 경우, 불량 대출금은 23조3천 억원으로 전체 대출금의 14.3%에 달하게 된다. 이는 미국은행의 불량채권율 1.16%(95년도기준)의 14배이며 자기 자본의 76%에 이르는 심각한 수준에 와 있다.

이러한 부실채권의 증가는 결국 기업들의 금융 고비용으로 이어지고 있다. 96년도의 우리나라 기업들이 부담한 금융비용은 매출액 대비 5.6%로 일본(1.6%이하), 대만(1.7%), 미국(1.8%)등에 비해 3배 이상 높은 것으로 나타나고 있다. 90년대에 들어서면서 차입금리는 낮아지고 있지만 외부자금 차입확대와 매출액 둔화 등으로 우리나라 기업들의 금융비용이 높아지고 있으며 불량채권을 줄일 경우 기업들의 금융비용을 상당히 낮출 수 있을 것으로 보고 있다. 그러나 우리나라의 대출심사는 아직도 1960년대에 개발된 신용평가표(Scoring Table)와 신용평가 기관들의 금융거래

사고자 자료에만 의존하는 지극히 초보적인 수준에 머물러 있어 신용평가 기술의 선진화가 절실히 요구되고 있다.

지금까지 1차 금융기관들의 대출은 주로 정부의 경제개발 시책과 연계되어 이루어지는 금융기관의 국가기관화 내지 公기관화 때문에 대출평가 기술이 축적될 수 없었으며 여기에 항상 대출 수요가 대출 공급을 초과하는 수요초과 상태이었다는 점도 평가기술의 발전을 저해하는 중요한 원인이 되었다. 더구나 금융기관들의 대출이 부동산 내지 동산 담보와 연대 보증인 제도에 전적으로 의존하였으므로 대출세일(sale) 상태에서 소비자들을 지속적으로 평가하고 기술을 개발해온 선진국에 비해 낙후된 기술수준에 머물 수밖에 없었다.

우리나라 기업들의 금융비용을 낮추어 국제시장에서의 경쟁력을 회복하기 위해서는 부실 채권을 줄일 수 있는 자체적인 기법의 개발과 선진국으로부터의 기술이전이 필요하다고 하겠다. 선진국에서와 같이 향후 성장 가능성이 있는 기업과 도산 가능성이 있는 기업을 분석 구분하기 위해서는 우리나라 금융기관들이 전적으로 의존하고 있는 感과 소문에 의한 주먹구구식 평가를 버리고 체계적이고 과학적인 기법을 도입하고 개발할 필요가 있다. 대출신용 평가 기술은 컴퓨터 시스템으로 운영되기 때문에 금융기관의 感과 노하우가 아닌 정보기술로서 이전되어야 하며 네트워크의 발달로 인한 전산망 감시기술로 도입되어야할 것이다.

-불량 신용카드의 증가: 96년 6월의 통계에 따르면 우리나라 신용카드회사는 8개의 전업회사가 있고 29개 은행이 신용카드업을 인가 받아 겸업하고 있다. 이들이 발행한 카드는 모두 3,725만장으로 인구 1인당 0.8장 꼴이다. 신용카드를 이용한 거래금액은 연간 57조원을 넘어 민간 소비지출의 28%에 이르고 있으며 이용 잔액도 11조 7,000억원을 넘고 있다. 신용카드 가맹점은 313만개로 연평균 30%이상의 높은 증가율을 보이고 있다. 이중 연체 금액은 1조862억 원으로 이용잔액의 9.2%에 이르고 있다.

93년과 비교해볼 때, 93년에는 거래금액 27조원에 연체금액이 2,484억 원 이었으나 95년 말에는 거래금액 51조 5,817억 원에 불량카드 9,196억 원으로 카드 사용금액의 증가율(47%)보다 연체증가율(73%)이 훨씬 높은 현상을 보이고 있다. 이러한 부실 신용카드 거래에서 발생하는 피해액이 전체 신용판매액의 상당 부분을 차지하여 금융기관의 재무상태를 악화시키고 결국은 물가인상 요인으로 작용하는 부작용도 나타나고 있다.

불량신용카드에 의한 사회적인 비용을 줄이기 위해서는 현재 시장 점유율을 우선적

으로 높이기 위한 각 금융기관들의 카드 발행 관행도 바뀌어야 하겠지만 미국이나 유럽과 같이 개인 신용에 대한 엄격한 평가와 예측이 우선적으로 필요하다. 본 연구팀들의 신경망 기술을 적용한 사전 연구에 의하면 개인들의 신상 자료에 의한 신용도의 예측은 60%정도의 낮은 정확도를 보이지만 과거 거래실적에 의한 예측은 98%정도의 예측 정확도를 보이고 있다. 즉, 개인들이 신용카드 신청서에 기입한 직업, 직위, 소속기관, 결혼유무, 월수입, 주택 소유여부, 자동차 소유차종 및 연식 등에 의한 신용 예측은 정확도가 60%내외인데 비하여 지난 1년 6개월 동안의 연체횟수, 연체금액, 거래금액, 카드의 종류, 발행지 데이터를 사용한 경우에는 그 정확도가 98%에 달하였다. 이제 국내의 신용거래 자료도 어느 정도 축적이 되어 있으므로 선진국들이 개발한 신경망, Genetic Algorithm, Agent기술을 도입하여 적용할 시점에 도달하였다고 생각된다.

-**불량 소비자의 증가**: 이외에도 백화점, 의류업체, 고급 음식점, 호텔 등에서 발행하는 고객카드 역시 상당한 불량채권을 가지고 있어 물가상승과 나아가서는 사회적인 경쟁력을 약화시키는 주요한 원인이 되고 있다. 예를 들어 구두와 액세서리를 제조, 판매하는 에스콰이어의 경우 신용카드 발급은 18만 명에 이르고 있는데 그중 매달 3천 5백명 정도의 불량 신용판매 건수가 보고되고 있다[김 정원, 1994]. 1995년에 설립되기 시작한 할부금융회사들은 가전제품과 주택, 자동차, 피아노, 기업 설비 등의 신용판매를 하고 있는데 96년 대부분의 할부회사 불량채권율이 2%대에 이르고 있는데 이는 일본의 0.13%(3개월 연체, 1996년 기준)에 비해 15배 정도에 이르고 있다.

이처럼 금융회사는 물론 소비재기업, 백화점, 호텔, 골프장, 자동차 회사, 할부금융회사 등 사회전반에서 생기는 불량채권은 금융비용을 상승시켜 국제 경쟁력을 떨어뜨리는 주요한 요인이 되고 있으며 이는 향후 정보화 사회의 기반 구조로서 필요한 신용사회의 구축을 가로막고 있다. 더구나 IC카드 기술 발달에 의한 전자화폐의 보급과 인터넷을 통한 전자상거래의 보급이 빨라질 전망이어서 이에 대비한 신용평가 및 전자상거래 기술의 개발이 절실히 요청되고 있다.

-**금융 시장개방에 대한 대비**:UR협상 타결에 따라 향후 금융시장과 서비스 산업 시장이 완전히 개방되면 고객의 신용관리에 경험과 노하우가 전무한 국내업체들이 200년 이상의 경험과 노하우가 축적된 외국의 금융기관이나 서비스 회사와 경쟁을 하게될 것인 바 고객의 신용에 대한 과학적이고 체계적인 평가는 이제 국가 경쟁력과 개별 기업의 사활을 결정하는 중요한 기술이 될 것으로 예상된다. 선진국의 경우 고객신용에 대한 예측의 정확도를 높여서 부실 신용판매를 줄이기 위해 그 동안

과거의 경험과 통계적 방법에 근거한 고객신용 평가표(Evaluation Table; Scoring Method)방법에서 전문가 시스템을 이용한 평가에로, 그리고 최근에는 신경망 방법을 동원하여 고객신용을 평가하고 있다.

따라서 우리나라도 부실 신용 대출이나 판매에 따르는 개별 기업들의 손실과 사회적 비용을 최소화하기 위해서는 최신의 첨단 이론과 국내 기업들의 경험적 노하우를 바탕으로 하는 고객 신용평가를 위한 시스템을 개발하여야할 필요성이 있다고 하겠다. 더욱이 가까운 시일 내에 국내 시장의 완전 개방과 정보화 사회, 신용사회가 도래할 것으로 예상되고 있어 자동화된 신용평가 시스템의 개발은 중요한 국가적인 과제라 하겠다.

제3절 연구 개발의 필요성: 사회. 문화적 측면

-**사회적 신뢰의 구축:** 우리나라는 미국이나 일본에 비해 대체적으로 사회적인 불신감이 높은 것으로 조사되고 있다. 세계 각 국이 온 힘을 다해 추진하고 있는 정보화가 사실상 사회적 신뢰라는 기반 위에서 제대로 설 수 있다는 점을 고려한다면 우리나라는 정보화 이전에 신용사회의 구현을 앞당기기 위한 방안을 강구할 필요가 있다. 사회적인 신뢰도가 낮은 사회에서는 유통되는 정보의 진위를 가리는데 많은 시간과 노력이 필요하고, 이로 인하여 컴퓨터와 고속 대용량 통신망을 갖춘다 하여도 이중 삼중의 검증 과정을 거치다보면 정보의 속도는 수작업의 속도보다 늦어질 수도 있다. 예를 들어, 우리나라의 경우에는 사원채용의 경우 주민등록과 학력증명서, 졸업증명서와 성적증명서, 이력서와 추천서 등의 많은 데이터를 요구하지만 미국의 경우, 이력서 한 장으로 갈음하고 있다. 이는 사회적인 신용과 믿음이 있기 때문에 별도의 검증과정이나 복잡한 증빙 서류가 필요 없게 된 것이며 이러한 절차의 생략은 정보화와 함께 가장 짧은 시간에 업무처리를 가능케 하여 생산성향상에 있어 시너지효과로 나타나고 있다.

선진국들의 이러한 사회적 신뢰의 구축은 오랜 세월동안의 엄격한 신용관리로 가능해진 것이며 우리나라에서 필요한 신뢰사회 구축을 위한 신용관리는 엄격하고 공정하며 체계적인 개인이나 기업의 신용평가 기법의 개발이 있어야 가능해진다. 즉, 개인 신용자료의 공개와 기업신용의 과학적인 평가, 그리고 평가자료의 공정한 관리가 이루어진다면 개인이나 단체 스스로 신용관리를 하도록 유도할 수 있을 것이며 이는 결국 사회적 신뢰의 구축으로 이어지게 될 것이다. 신용평가 기법의 개발과 자료의 축적은 전자 상거래의 활성화뿐만 아니라 이처럼 건전한 소비생활과 기업운영을 유도하고 결국은 사회적 신뢰를 구축할 수 있을 것으로 예상한다. 따라서

우리사회의 정보화를 촉진하기 위한 전제조건으로서, 사회적 신뢰가 필요하고 사회적 신뢰를 쌓기 위한 필요조건으로서 신용평가 기법의 발달과 신용평가 데이터의 관리가 필요한 것이다.

-**안전한 전자상거래의 촉진**: 향후 각국의 사회적 생산성과 국가 경쟁력은 정보의 흐름, 즉 그 속도와 내용에 의해 결정될 것으로 보고 있다. 즉, 이중 삼중의 검증 절차가 없이 정보가 흐를 수 있는 사회적 구조와 신뢰가 형성되어 있어야 정보가 빠르게 흐를 수 있을 것이며 다음으로는 정확하고 풍부한 양질의 정보가 생성되고 유통될 수 있는 기반구조가 필요하다. 정보가 빠르게 흐르기 위한 초고속 정보망이나 사회적 신뢰 외에도 이들 정보가 안전하고 손상되지 않도록 보호해주는 장치와 기술이 필요하게 된다.

정보화 사회의 틀 속에서 생산성 향상과 경쟁력제고 효과를 얻기 위해서는 전자상거래를 안전하게 할 수 있도록 거래를 보호하고 시스템을 보호하는 기술과 장치가 필요하게 된다. 전자 상거래 시스템의 거래의 흐름을 안전하게 하기 위해서는 Fraud Detection기술이 필요하다. Fraud Detection기술은 사용자의 신용카드나 컴퓨터 접속 ID번호를 위조, 변조, 복사한 경우와 전자상거래 판매자를 위장한 가짜상인을 발견, 추적하는 기술로서 최근 미국을 중심으로 활발하게 연구되고 있는 기술이다.

Fraud Detection기술은 향후 일반소비자 시장과 기업거래, 그리고 생산자와 유통상과의 거래를 지배하게될 전자 상거래가 안전하게 이루어지도록 보호하고 도와주는 역할을 하게되는 기반기술의 핵심기술이라고 할 수 있다. 따라서 국가 경쟁력의 관점에서 이러한 기술의 확보와 개발을 지원하여야할 것이다.

-**정보화의 촉진**: 컴퓨터 터미널을 이용하는 전자금융거래의 경우, 선진국에서는 전체거래의 80%에 이르고 있는데 비해 우리나라는 아직도 20%미만에 머물러 있는 형편이다. 국내 은행의 홈뱅킹 시스템이 하루에 처리하는 거래는 약 9만건으로 이는 은행원 3백명 분에 해당하는 업무량이다. 따라서 고객이나 은행들의 업무효율성을 높이고 정보화를 촉진하기 위해서는 컴퓨터 네트워크를 통한 금융거래와 상거래를 활성화시킬 필요가 있다. 그러나 안전한 금융거래와 상거래를 위해서는 네트워크 시스템에 침입한 외부인을 감시하고 추적할 수 있는 기술이 개발되어야하며 불량한 거래자를 적발(Frauds Detection)할 수 있는 시스템 기술도 개발되어야 한다. 본 연구에서 개발된 Fraud Detection, Intruder Detection 기술은 정보화에 필요한 시스템의 물리적인 보호와 정보흐름의 보호라는 관점에서 가장 중요한 핵심기술이라고 할 수 있다. 따라서 우리나라의 정보화를 촉진하기 위해서는 신용평가의 기법

개발과 데이터 베이스의 구축, 그리고 Fraud Detection기술과 Intruder Detection 기술이 절대적으로 필요하다.

제2장 신용평가 시스템 개발

제1절 Intelligent Credit Evaluation System에관한 연구

Credit evaluation/financial fraud detection is one of the most important tasks usually assigned to experienced officers in credit card companies, mortgage companies, banks, consumer goods companies and other financial institutes. The first and most important benefit possibly obtained from the 'automated credit evaluation system' is that decisions made by the system can be objective and free from arbitrary and capricious behaviour possibly conducted by human credit evaluators. In addition, the evaluation system, if it can make statistically sound and reasonable decision, can reduce bad debt losses, grant more credit to consumers and achieve organizational consistency in decision making by evaluating credit more systematically.

1. Credit Scoring

'Credit Scoring' is one of the most commonly used financial risk assessment systems which is comprised of two separate subsystems: a scoring table and a repayment probability table(Noel, 1982). Traditionally, the credit scoring system lists up multiple attributes in the scoring table, consisting of attributes in row(column) for describing an application and possible attribute values in column(row) splitting into intervals. Then the system assigns weight to each of the attributes which the application belongs to, simply sums up the point values of each attribute and then compares total point with two threshold values. If the total score exceeds the upper threshold values, credit is automatically granted. If the summed score is less than the lower threshold value, the application will be rejected. IF the summed score falls between the upper and lower thresholds, further investigation will be made.

2. Expert Systems

The researchers in an accounting field who were interested in assessing bank loan loss(bad credit) developed prototype systems based on knowledge collected from human experts. AUDITOR developed by Dungan(1982) and Dungan and Chandler(1985) aids public auditors in estimating the dollar amount of client's uncollectable accounts receivable. The experts' knowledge is organised into a hierarchical structure and coded using AL/X, the expert development shell. At the bottom of the hierarchical structure, there are decision cues which are used in the experts' decision making process and are identified in audit manuals. In the system performance validation test, AUDITOR's judgements were confirmed by human experts. In the "Open-Book" test, the agreement rate showed about 90 percent and in the "Blind" test, it reached about 91 percent.

CFILE was developed by Peat, Marwick, Mitchell & Co., one of the Big Eight firms to assist auditors in assessing bank loan loss reserves(Messier and Hansen, 1987), (Connell, 1987), Before the bank loan is granted, the bank clerks in a loan department are asked details about the size of the loan, security held, period and conditions of the loan. Depending upon the answers to these questions, CFILE continues to ask further questions and when sufficient data is collected for making an appropriate decision as to how much should be reserved, the conclusion was displayed to users. In the performance test, two partners and one senior auditor investigated 16 loan cases and it was shown that CFILE's judgements agreed with the conclusion of human expert as follows: the partner's judgement in 11 out of 16 loans and inexperienced senior auditor's judgement in 10 out of 16 loan classes.

Leinweber(1988) introduced the American Express Authorizer's Assistant, a credit advisory system. The system is a high-speed, high-volume transaction processing knowledge server linked with American Express' existing Credit Authorization System(CAS). Routine charges were automatically approved by the system. Unusual purchases were passed to the human authorizers who evaluate charges according to previous purchase and payments, the velocity and type of recent transactions and any additional data related with the account.

3. Machine Learning System

Another popular approach to automated fraud detection is a machine learning. Among various machine learning algorithms, Carter and Catlett(1987) employed

ID3 approach to develop a credit card endorser. They found that ID3 algorithm suggested by Ross Quilan improved the accuracy of risk assessing capability fro 78.4% to 80.4%. When pruning was done using C4 which simplifies the decision tree by merging subtrees into leaves, the system was improved even more, achieving a prediction rate of 85.5%. Kim(1996) also used a decision tree algorithm C4.5 to evaluate motor insurance claims and it showed up to 85.79% classification accuracy on unseen data.

4. Neural Networks

Douglas et al(1990) emulated mortgage underwriting judgements by using a neural network developed at Nestor Company to test the performance of their neural network products. In the test, they found that it is possible to predict bad loans, that is the loan would go delinquent in payment if granted, with 95% accuracy. Even if the test results did not say that the mortgage underwriting system can be directly applied to real world problem solving, it is evident that the application of the neural network to solve the financial classification problems are highly possible.

Odom and Sharda(1990) compared the prediction capability of neural networks and discriminant analysis for bankruptcy prediction. They chose sample firms which went bankrupt between 1975 and 1982. The network topology has five input nodes, five hidden nodes and one output node. The first experiment was performed with data from 74 firms, 38 of which went bankrupt and 36 of which were non-bankrupt firms. The system was trained through backpropagation and correctly predicted all 38 bankrupt and 36 non-bankrupt firms respectively. The sampling data was adjusted for the real world ratio of non-bankrupt firms to bankrupt firms. The second data group consisted of 36 non-bankrupt firms and 9 bankrupt firms while the third data group consisted of 35 non-bankrupt firms and 4 bankrupt firms. In the test of bankruptcy prediction on the three groups of data, the neural network performed better than discriminant analysis. In the test of non-bankruptcy prediction, the neural network performed better than discriminant analysis on all except the second group data.

Surkan and Singleton(1990) used a neural network for bond rating. They employed seven financial factors and tested the network performance of two different configurations: a network with a single hidden layer and a network with two hidden layers. The single layer network showed a much lower level of accuracy than the two other networks with the two hidden layers. Based on the results, Surkan and Singleton concluded that the advantage of networks trained with two hidden layers over a single layered network is significant.

Kim(1992) performed a similar experiment with data collected for bond rating on the classification and prediction capability of three different methodologies: neural networks, knowledge-based systems and statistics techniques. His research was done with eight input variable and six output gradings of a business organisation: AAA, AA, A, BBB, BB, and B. AAA indicated the highest graded company. The analytic tools employed for comparison were backpropagation, regression analysis, logistic analysis, discriminant analysis and the ID3 inductive learning algorithm. Training was done on neural networks with various configuration of layers, number of hidden nodes, and activation functions. The classification capability was tested against three different data sets. In this study, he found that the neural network approach performed better than the other four prediction tools over the three data group.

제2절 계층화된 신경망 기반의 신용평가 시스템 개발

1. Research Background

In the trends of domestic market opening and globalisation of production systems, building an accurate credit evaluation system is increasingly important. The opening of Korean financial market is projected in 1988, just two years away, and thus introduction of automated credit evaluation system is inevitable. The korean financial institutes currently are not operating automated credit evaluation system, but also do not have knowledge of customers which is needed for developing such as system.

The reason why Korean financial institutes have not sophisticated credit evaluation system is explained by three elements: protection from government, co-sign system and security system. Traditionally, the Korean financial institutes have been controlled and protected by the government and thus need not develop ideas and products to attract customers. The low competitiveness of the Korean institutes can be explained by the over-protection of the government. Then, co-sign system and security system in which the financial institutes need not evaluate the debtor's credit have gradually deteriorated the competitiveness of the institutes.

Now that many foreign financial institutes are ready to open their shops in Korea, the Korean counter-partners should be prepared to protect their own interests and market share in Korea. For the reason, recently a few institutes tried to develop credit evaluation system using statistical models and expert system approaches. Scoring table which are constructed on statistical knowledge and intuitive observations have been in use for long time. As in the scoring systems employed in other countries the scoring table does not differentiate individual credit status, but discriminate categorized difference, such as job and position, company level, or even housing data. The categorized information cannot discriminate each individual's potential credit quality and projected capability, credit loss frequently occurs. The credit loss in Korea is as high as 2.5% of the sales amount, in contrast to 0.5% in Japan.

Recently, in foreign case and domestic case, used is the statistical method which are based on MDA(Multiple Discriminant Analysis), regression analysis, probit, logit, etc to company bankruptcy and credit evaluation, but since later of 1980s artificial intelligence method has been used which are based on inductional learning method, neural network, etc to company credit evaluation, bankruptcy prediction.[Lee, Han, 1995]

In the case of foreign countries, there are many success cases with neural network in the various fields: CROSBY[Hamscher, 1991], C. P. A. R. K[Bymes, Nealis, 1991], Margin Credit Evaluation System[Beshinke, Nigam, 1991] credit evaluation[Schumann, 1992], mortgage risk assessment[Douglas et al., 1990], bankruptcy prediction[Odom & Sharda, 1990], etc.

Compared with foreign case, Korean credit prediction rate is low. Kim [1994] and Choi [1995] tried to develop credit evaluation systems based on backpropagation algorithm of neural networks. The result was a little bit discouraging. The result shows that the factors included in the current system reflect many features of Korean customers and social practise. The factors included in the current system might be much different from those included in the system developed in other countries [Kim, Choi, Chung, Kang, 1994]. Hence, it is necessary to investigate whether any meaningful relationship between credit factors and final output exists or not.

The customer evaluations which are performed in Korean credit card company are processed through the interviews. We could consider the 3-level's evaluation model, the first is for assigning initial limitation, the second is for adjusting the limitation after issuing credit card, and the third is for overdue management and repayment method. In this research, the goal is to investigate the feasibility of developing an automated credit evaluation system that has high degree of generalization and is adequate to the actual circumstances of Korea. Specifically, the neural network mechanism, backpropagation with stratified method, was adopted as a credit evaluating processor in the research, because the neural network could predict the output values by non-linear mapping, even though it didn't know the direct relations between the input values and the output values.

2. Previous Approach

가. Credit Evaluation

Credit is granted on the basis of a subjective trust that the payment will be made in the future[Park, 1988]. Credit evaluation based on such a subjective trust is a process to evaluate the degree of ability/willingness to pay in the future, after investigating and analyzing the overall factors that influence the credit status of individuals and organizations.

The purpose of credit evaluation is to determine whether an applicant's dealing is beneficial or harmful to credit card companies and to minimize a loss that may be induced. However, it is one thing to maximize a profit, and another to minimize a loss. To maximize a profit by enlarging the business scope, a company has to put up with a little loss. So, when credit policies are established, a company could aim to minimize a loss under the given business scope, or to enlarge the business scope by admitting a little loss. This will affect credit investigation methods.

Credit evaluation is classified as a general purpose credit evaluation if it is done by banks to extend commercial loans. It becomes a special-purpose credit evaluation if done by credit evaluation agencies to grade stocks and bonds[Song, 1988].

Credit evaluation is divided into two types. The one is for the individual customers, and the other is for the companies. In case of individual assessment, based on the results of the credit evaluation determined are loan amount and amount limit per transaction. In case of corporate evaluation, the evaluation results are used as basic determinant for financial condition such as acceptance or rejection of the loan application, interest rate and loan period, and etc.

The outputs of credit evaluation in the credit card company are acceptance or rejection of the application, initial credit limit of new customer, modification and updates of credit limit for existing customers, and management of the delinquent customers and collection of the overdued loan. Information used to evaluate a company's credit consists of financial statement and non-financial statement. Information on financial status is represented in the form of financial ratios that summarize balance sheet, income statement, statement of changes in financial position(SCFP), statement of appropriation of retained earnings etc. However, numerical values in the financial statements, like the size of assets or sales, are often used. There is a variety of information which is related to management and business environment, forecasting, and market fluctuations. This is non-financial information [Choi, 1995].

The universal information used to evaluate individual credit consist of following factors, as shown in Table-II-1:

1	Age
2	Sex
3	Work place, Position
4	Ability of repayment covering family members
5	Material status
6	Type of housing
7	Place to send bill
8	Residential Area
9	Annual income
10	Application
11	Others needed

표 2 : Credit evaluation factors>

Credit evaluation is performed based on the information enlisted above. Otherwise, supplementary data could be collected. There has been a drastic change in credit management practices; from company-oriented to individual-oriented, and from mortgage-oriented to credit-oriented. With the changes and thanks to the growth of banking business, the financial institutes such as banks and credit companies must measure accurately and manage carefully credit information of customers [Lee, 1985].

최근의 신용평가기법은 회귀분석 및 다변량모형과 다변량판별분석모형, 로짓분석모형, 반복분할분석모형을 이용한 통계적인 방법을 이용하여 신용평가를 하고 있다. [백춘봉,1999/이정도,설병문1998]

나. Neural Network Approach

The studies applying neural network models show that the neural network approach is very powerful tool for business applications. Douglas et al.[L. R. Douglas E. Collins, C. Scofied, and S. Ghosh] emulated mortgage underwriting judgement by using a neural network developed at Nestor Company to test the performance of their neural network products. In the test, they found that it is possible to predict bad loans, that is, loan would go delinquent in payment if granted, with 95% accuracy. Even if the test result did not say that the mortgage underwriting system can be directly applied to real world problem solving, it is evident that the applications of the neural network to solve the financial classification problems are highly possible.

Odom and Sharda[Odom & Sharda, 1990] compared the prediction capability of neural networks and discriminant analysis for bankruptcy prediction. They chose sample companies which went bankruptcy between 1975 and 1982. The network topology has five input nodes, five hidden nodes and one output nodes. The first experiment was performed with data from 74 firms, 38 of which went bankrupt and 36 of which were non-bankrupt firms. The system was trained through backpropagation and correctly predicted all 38 bankrupt and 36 non-bankrupt firms, respectively. The sampling data was adjusted for the real world ratio of non-bankrupt firms to bankrupt firms. The second data group consisted of 36 non-bankrupt firms and 9 bankrupt firms while the data group consisted of 36 non-bankrupt firms and 4 bankrupt firms. In the tests of bankruptcy prediction on the groups of data, the neural network performed better than discriminant analysis. In the tests of non-bankruptcy prediction, the neural network performed better than discriminant analysis on all except the second data group.

Surkan and Singleton[Surkan & Singleton, 1990] used a neural network for bond rating. They employed seven financial factors and tested network performances of two different configurations : a network with a single hidden layer and a network with two hidden layers. The single layer network showed a much lower level of accuracy than the two other networks with the two hidden layers. Based on the results, Surkan and Singleton concluded that advantage of networks trained with two hidden layers over a single layered network is significant [In-goo Han, Young-sig Kwon, Hong-kyu Jo, 1995].

Lee, Han and Kim [1995] employed a hybrid approach to evaluating corporate credit which integrate statistical method with neural network technology, based on heuristic method. Credit data of 1043 companies are collected to compare the performance results of MDA and neural network systems. The experimental data were collected in 1991 and 1992 from the credit evaluation institutes, Korea Credit Evaluation, Korea Credit Information, and Korea Enterprise Credit Evaluation.

The credit data are classified into 5 classes. The evaluation factors are categorized into financial factors and non-financial factors. The financial factors include sales growth, unit profitability, stability, activity, productivity, and cash flow, while the non-financial factors include type of the company, type of the industry, and environmental factor. In contrast to the traditional classification of 'good' and 'bad' categories for enterprise credit, the companies are classified into 5 groups. For the MDA, OPP(Ordinal Pairwise Partitioning) was employed which is a new discriminant method. In the performance test, they found that the new mythology enhanced the classification accuracy, from 62.3% of MDA to 67.2% of neural networks.

On the other hand, when forward OPP method was employed, the classification accuracy was raised to 78.2% for MDA, and to 82.2% for neural network. As was in previous comparative research of traditional classification method and neural network training, the experiment proved that performance of the neural network systems is better than traditional methods. One distinctive feature of the experiment is that the forward OPP method shown much better results, when integrated with neural networks; 67.2% for the traditional neural network, 82.17% for the neural network in forward OPP method, and 85.61% for the integrated method.

Dutta and Shekhar(1988) showed that predictive accuracy of the neural network outperformed that of regression in bond rating, although their neural network implementation might not be adequate. Liang, Chandler, Han, Roan[1992] compares the performance of neural network, ID3, and probit models under various data conditions. The classification models include eight numeric variables(financial ratios and accounting numbers) and one nominal variable(industry classification). They shows that the neural network model mostly outperform the probit and ID3 methods in predicting accounting inventory method choice.

Tam and Kiang[1992], using bank failure data, compares a neural network approach with linear discriminant function, logit model, *k* nearest neighbor, and ID3. They proposed a backpropagation learning algorithm modified to include prior probabilities and misclassification cost. The empirical results shows that the neural net is a promising method of evaluating bank conditions in terms of predictive accuracy, adaptability, and robustness, especially under the conditions of multimodal distribution, adaptive model adjustment.

Chung and Silver[1992] compared linear models derived by logit model with rule-based systems produced by two induction algorithm, ID3 and the genetic algorithm(GA). The techniques performed comparably in modeling the experts at one task, graduate admission, but differed significantly at a second task, bidder selection, implying that categorical conclusions concerning the relative performance of the linear model and the induction algorithms are not appropriate. The other findings are that, for both induction algorithms, predictive performance depends on characteristics of the problems-solving task under consideration, and the linear should not necessarily be abandoned as a useful paramorphic model of human expertise, which attempts to simulate expert decisions without regard to the cognitive processes through which those decisions were reached.

Kim[1992] performed a similar experiment with data collected for bond rating on the classification and prediction capability of three different methodologies: neural networks, knowledge-based systems and statistical techniques. His research was done with eight input variables and six output gradings of a business organization : AAA, AA, A, BBB, BB, and B. AAA indicates the highest graded company. The analytic tools employed for comparison were backpropagation, regression analysis, logistic analysis, discriminant analysis, and the ID3 inductive learning algorithm. Training was done on neural networks with various configuration layers, numbers of hidden nodes, and activation functions. The classification capability was tested against three different data sets. In this study, he found that the neural network approach performed better than the other four prediction tolls over the three data groups.

Jo[1994] applied the MDA, neural network, and analogical reasoning which is included in AI techniques in bankruptcy predicition problem. The analogical reasoning method is the fundamental ability of humans to easily understand new situations by relating them to old ones and to solve problems based on previous experience of analogous problems. The method which had the highest classification ability among the three methods was neural network. He suggested the data refinement method to reduce the irregularity of source data. Two theorems were applied in building the architectures of neural network, those could determine the outline of architecture. The outline are related to the following two questions, how many hidden layer are needed and how many PEs are needed in hidden layer.

The performance the results of selected previous studies are summarized in <Table-III>.

표 3 :Performance of previous studies
([In-goo Han, Young-sig Kwon, Hong-kyu Jo, 1995])

Source	Statistical method	AI method	Domain
Braun and Chandler (1987)	53.8(DA)	63.8 (ID3)	Stock market
Green (1987)	79.9(Logit)	71 (ID3), 80.7 (GA)	Simulated data
Dutta et al. (1988)	64.7(Reg.)	88.3 (2-layer NN) 82.4 (3-layer NN)	Bond rating : result using ten variables
Surkan and Singleton (1990)	39.0(DA)	65.0 (3-layer NN)	Bond rating
Odom and Sharda (1990)	59.26(DA)	81.48 (3-layer NN)	Bankruptcy prediction : result of 50/50 training sample portion
Chung and Silver (1992)	79.0 (Logit) 83.8 (Logit)	77.8 (ID3), 80.0 (GA) 50.5 (ID3), 88.6 (GA)	Graduation admission task Bidder selection task
Kim (1992)	77.84 (Reg.) 75.0 (Logit) 76.67 (DA)	76.67 (ID3) 84.5 (3-layer NN)	Bond rating
Cronan et al. (1992)	85.9 (RP) 62.5 (RP) 89.0 (RP)	78.4 (ID3) 50.0 (ID3) 80.0 (ID3)	Mortgage loan Commercial loan Consumer credit
Tam and Kiang (1992)	84 (DA) 81.8 (Logit) 77.2 (NN) 77.2 (NN)	79.5 (ID3) 81.8 (2-layer NN) 85.2 (3-layer NN)	Bankruptcy prediction : One year period hold-out sample case
Chung and Tam (1992)		38.5 (ID3(threshold)) 59.6 (ID3(chi-square)) 48 (AQ) 73 (2-layer NN) 79.5 (ID3 (threshold)) 79.5 (ID3(chi-square)) 77.5 (AQ) 85.3 (2-layer NN)	Construction project performance assessment : test set Bankruptcy prediction : One-year period, test set
Jo (1994)	88.75 (DA)	90.91 (3-layer NN) 87.68 (AR)	Bankruptcy prediction

Note : Reg.=Regression, DA=Discriminant Analysis, GA = Genetic Algorithm, RP=Recursive Partitioning, 1 NN=1 Nearest Neighbor, n-layer NN=n-layer Neural Network, AR = Analogical Reasoning.

The review of previous studies shows that the neural network model is the most powerful classification tool although there remain many unresolved issues on the design and implementation of neural network.

Design of a Neural Networks Architecture Using Genetic Algorithms : Application to Credit Evaluation [J.W.Kim, 1996]

다. Expert System Approach

Leinweber[Leinweber, 88] introduced the American Express Authorizer's Assistant, a credit advisory system. The system is a high-speed, high-volume transaction processing knowledge server in operation, linked with American Express' existing Credit Authorization System(CAS). Routine charges were automatically approved by the system. Unusual purchases and the human authorizers who evaluate charges according to previous purchase and payments, the velocity and type of recent transactions and additional data related with the account.

Instead of employing the rule-based system approach, Carter and Catlett[Carter & Catlett. 87] employed the ID3 approach to develop a credit endorser. They found that the ID3 algorithm suggested by Quilan improved accuracy of risk assessing capability from 78.4% to 80.4%. When pruning was using C4 which simplifies the decision tree by merging subtrees into leaves, the system was improved even more, achieving a success rate 85.5%. In another test, when a class probability tree was used, additional performance improvement was gained.

AUDITOR developed by Dungan(1982) and Dungan and Chandler(1985) demonstrated human expert level performance. AUDITOR aids public auditors in estimating the dollar amount of a client's uncollectable accounts receivable. As in the MYCIN system[Shortliffe and Buchanan, 1975], the expert's knowledge is organized into a hierarchical structure and coded using AL/X, an expert system development shell. At the bottom of hierarchical structure there are decision cues which are used in the expert's decision making process and are identified in audit manuals. In the system performance validation test, AUDITOR's conclusions were confirmed by human experts[Dunggan, 1982].

CFILE was developed by Peat, Mawick, Mitchell &co., one of the Big Eight firms to assist auditors in assessing bank loan loss reserves[Messier and Hansen, 1987; connell, 1987]. Before the bank loan department, is asked detailed about the size of the loan. security held, period and conditions of the loan. Depending on the answers to these questions CFILE goes on to ask further questions, and when sufficient data is collected for making appropriate decision as to how much should be reserved the conclusion was displayed to users. The system requires two years of audited information or three years of unaudited information. In the performance test, two partners and one senior auditor investigated 16 loan cases and it was revealed that CFILE's judgments were consistent with the conclusion of human expert : partner's judgments in nine out of ten loan cases, second partner's judgment in 11 out of 16 loans and inexperienced senior auditor's judgment 10 out of 16 loan cases.

수정된 ACLS 를 이용한 신용평가 전문가 시스템의 지식획득에 관한 연구에서 전문가의 지식획득을 이용한 신용평가시스템을 개발하였다.[임성식, 1995]

라. Stratified Method

As the learning mechanism of neural networks basically rely on non-decreasing interpolation, homogeneity is important to keep closer similar data into the same group. To maintain homogeneity of the training data, the way is to categorize data into homogeneous groups. The stratified grouping can be conducted based on the following principles. First, the population should be categorized into small groups of mutually independent and collectively exhaustive set. Second, the number of sample is determined by the predetermined ratio of sample obtained from the small group. There are two stratified grouping methods. First, proportional stratification determines the size of sample in proportion with the small population group. Second, in the non-proportional stratification the sample data are collected based on size of the population group and degree of the deviation.

“경기변동을 고려한 기업신용평가 - 인공신경망의 응용”은 경기변동을 고려한 기업의 신용등급평가 모형에 관한 연구로써, 기업의 신용 등급과 경기변동이 유사한 분

포형태를 갖고 있음에 착안하여 기업 신용등급 평가모형에 경기변동을 고려했으며, 또한 분류기법으로 인공신경망을 활용하였다. [안은주, 2000]

3. Implementation

가. Problem domain

One of the most difficult problems in credit evaluation is that customer data collected before endorsement of the credit cards contain too little information of the customer's personal properties. Because many financial companies are involved in the highly competitive market in Korea, credit evaluation systems of high accuracy should be developed.

나. Data gathered

The customer data used in this research was collected by a Korean credit card company.

(1) Independent Variable

In developing credit evaluation systems, selection of variable is difficult and complex, because the multicollinearity between multi variables should be considered. When two different variables are correlated each other, then the interpretation of the system can be biased. For example, the three variables of job position, years in the organization, and annual income are interrelated, because annual salary is determined by job position and years in the organization.

In the previous research on the same data set, the analysis on the significance of each variable was conducted and the results are shown in the table. In the analysis, 12 important variables were selected and then the significance of each variable was measured by the regression coefficient.

표 4 : Independent Variable

	Variable	Contents	Coefficient
1	Overdue number over than two times		0.8460
2	Demand place of bill	Home=1, Work place=2	0.8340
3	Cash service jump	Standard cash service / average cash service during 6month	0.6514
4	Average of overdue number	12 Months	0.6161
5	Number of sales canceled		0.5989
6	Resident condition	Own=0, Rental=1	0.5476
7	Amount for payment jump	Standard Demand / Average Demand for 6 Month	0.5099
8	Duration		0.4249
9	Family Card		0.3900
10	A lump-sum payment jump	Standard lump-sum / Average lump-sum for 6 Month	0.3746
11	Sex	male=1, female=0	0.3136
12	Installment amount		0.3095
13	Age		0.1548
14	Max. overdue status		0.1302
15	Cash service for six month		0.1270
16	Overdue number	12-Month	0.1185

In selecting factors which the credit evaluation system should consider, environmental factors such as social customs, cultural difference, and customer's behaviors should be included. In Korea, as the behaviour of the customers are so different from customers in other countries, the credit evaluation system developed in other countries cannot directly applied to Korean customers. As was revealed in this research, the behaviour of customer such as the number of overdued payment in a period, the number of initial overdued payment in installment sales, billing address, and cash service jump were more important. In contrast, other personal data such as occupation, position, organization, age, sex, and residential area are less important than the other variables.

On the other hand, occupation, years in the organization, marital status, and annual income are very important factors considered int the evaluation in the States [Capon]. Organization may be important factor in determining credit risk. Also, residential area has been thought to be an important consideration, however, it was found that it does not affect credit status [Beranek, W. and W. Taylor].

(2) Dependent Variable

Dependent Variable is 'Credit Status'. According to the number of overdue payment, credit status was divided into two statuses such as 'good' and 'bad'. 'Good credit' of the customer is determined when the number of overdue payment in the following 6 months is more than 3. But, when the number of overdued payment is less than 3 times, then the customer is classified as 'good credit' customer.

표 5 : Dependent Variable

	Variable	Condition
1	GOOD	Max. Overdue Number ≤ 2
2	BAD	Max. Overdue Number ≥ 3

(3) Data Sampling

The number of customer data collected in this research was 3,999. The customer data is divided into two data sets; the data set-I (January 1995 - December 1995) and data set-II (July 1995 - December 1995).

The customer data is divided into four classes, depending on the credit status. The gradel means the group of customers with "VIP card", grade2 means the group of customers with "GOLD card", grade3 means the group of customers with "general card", and grade4 means the group of customers with "card". The data is illustrated in Table-6.

	Total Size	Training	Testing
GRADE1	836	200	636
GRADE2	847	200	647
GRADE3	1920	200	1720
GRADE4	396	200	196
TOTAL	3999	800	3199

표 6 : Sample Size & Status

다. Implementation of the credit evaluation system

(1) Backpropagation

The system, as usual of the backpropagation systems, is comprised of three layers: input layer, hidden layer, output layer. Each layer has 16 nodes in input layer, 8 nodes in hidden layer, 1 nodes in output layer, respectively. To investigate the degree of generalization, the neural network was designed as Figure-2.

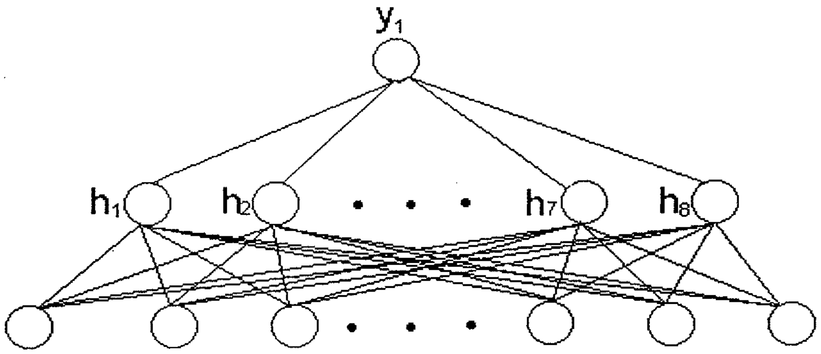


그림 2 : Structure of Backpropagation Network>

x_1 : Overdue number over than 2times

x_2 : Demand place of bill

x_3 : Cash service jump

x_4 : Average of overdue numbers

x_5 : Number of sales canceled

x_6 : Resident condition

x_7 : Amount for payment jump

x_8 : Duration

x_9 : Family card

x_{10} : A lump-sim payment jump

x_{11} : Sex

x_{12} : Installment amount

x_{13} : Age

x_{14} : Max. overdue status

x_{15} : Cash service for six month

x_{16} : Overdue number

h_1 · · · · · h_8 : hidden nodes

y_1 : output node(overduced number over than three times)

(2) Backpropagation with Stratified method

Basically, each layer has 16 nodes in input layer, 8 nodes in hidden layer, 2 nodes in output layer. To enhance the prediction accuracy, the customer data is categorized into four classes, based on the credit card grade. The credit card grade is determined by the credit grantors based on the personal data, such as organization, type of job, years in the organization, position, and annual income. We believe that information on the credit card grade is very important in discriminating customers. Following table shows the rate of good customers in the class.

	Data	Rate of good customers
GRADE1	836	.190
GRADE2	847	.205
GRADE3	1920	.507
GRADE4	396	.099

표 7 : Stratified Input Data

4. Experiments and the Result

The credit evaluation system was implemented in C-language, running on IBM-PC compatible with Microsoft operating systems. The backpropagation, which is one of Artificial Neural Network algorithm, was applied in this system. The system was comprised of 16's of input nodes, 1's output nodes, and 8's hidden nodes in one hidden layer. The initial values of learning rate and momentum values are set 0.9 and 0.7, respectively. The error limit was 0.01. The experiments were performed with two way. The first trial was applied stratified method which stratifies the data for card level : gradel, grade2, grade3, grade4. The second trial was not applied stratified method.

가. Backpropagation with stratified method (grade 1)

Gradel is the group of the customers with "VIP Card" and consists of 836 data. The 200 training data and the 636 test data are randomly chosen from gradel. The initial error size, which is defined as the difference between output value of unadjusted network and desired output value, was measured 0.0698912. With further training iterations, at the epoch 1,944 the error size was reduced to

0.0000997, which is less than the predetermined convergence limit level. The prediction accuracy was encouragingly high : 635 correct predictions, 99.84% accuracy levels. The results are shown to the Table-8.

	Contents	Result
Training	Number of hidden layer	1
	Number of iterations	1944
	Number of hidden nodes in hlayer	8
	Total error(%)	0.00009970
	Recognition error rate(%)	0.0
Test	Number of test pattern	636
	Number of error pattern	1
	Recognition error rate(%)	0.163132
	Total error(%)	0.000717
	Accuracy rate(%)	99.84

표 8 : Result of Backpropagation with Stratified (grade 1)

나. Backpropagation with stratified method (grade 2)

Grade2 is the group of the customers with "Excellence Card" and consists of 847 data. The 200 training data and the 647 test data were randomly chosen from grade2. The initial error size, which is defined as the difference between output value of unadjusted network and desired output value, was measured 0.067267. With further training iteration, at the epoch 1,797 the error size was reduced to 0.0009966, which is less than the predetermined convergence limit level. As a result, The prediction accuracy rate was 98.6% : 638 correct predictions. The results are shown to the Table-9.

	Contents	Result
Training	Number of hidden layer	1
	Number of iterations	1797
	Number of hidden nodes in hlayer	8
	Total error(%)	0.0009996
	Recognition error rate(%)	0.5
Test	Number of test pattern	647
	Number of error pattern	9
	Recognition error rate(%)	1.442308
	Total error(%)	0.002442
	Accuracy rate(%)	98.6

표 9 : Result of Backpropagation with Stratified (grade 2)

다. Backpropagation with stratified method (grade 3)

Grade3 is the group of the customers with "General-1 Card" and consists of 1920 data. The 200 training data and the 1720 test data are randomly chosen from grade3. The initial error size, which is defined as the difference between output value of unadjusted network and desired output value, was measured 0.051918. With further training iterations, at the epoch 1,797 the error size was reduced to 0.0009970, which is less than the predetermined convergence limit level. The prediction accuracy was inferior to two class: gradel, grade2. It was found that the result of test was 95.11%: 1636 correct predictions. The results are shown to the Table-10.

	Contents	Result
Training	Number of hidden layer	1
	Number of iterations	1034
	Number of hidden nodes in hlayer	8
	Total error(%)	0.00009963
	Recognition error rate(%)	4.0
Test	Number of test pattern	1720
	Number of error pattern	84
	Recognition error rate(%)	5.072464
	Total error(%)	0.013229
	Accuracy rate(%)	95.11

표 10 : Result of Backpropagation with Stratified (grade 3)

라. Backpropagation with stratified method (grade 4)

Grade4 is the group of the customers with "General-2 Card" and consists of 396 data. The 200 training data and the 196 test data are randomly chosen from grade4. The initial error size, which is defined as the difference between output value of unadjusted network and desired output value, was measured 0.0576434. With further training iterations, at the epoch 1,371 the error size was reduced to 0.00099836, which is less than the predetermined convergence limit level. The prediction accuracy was encouragingly complete : 196 correct predictions, 100% accuracy levels. The results are shown the Table-11.

	Contents	Result
Training	Number of hidden layer	1
	Number of iterations	1371
	Number of hidden nodes in hlayer	8
	Total error(%)	0.00099836
	Recognition error rate(%)	0.5
Test	Number of test pattern	196
	Number of error pattern	0
	Recognition error rate(%)	0
	Total error(%)	0.000286
	Accuracy rate(%)	100

표 11 : Result of Backpropagation with Stratified (grade 4)

마. Backpropagation with stratified method(total)

The 800 training data and the 3199 test data are randomly chosen from the total data. The initial error size, which is defined as the difference between output value of unadjusted network and desired output value, was measured 0.0697456. With further training iterations, at the epoch 1,944 the error size was reduced to 0.0009971, which is less than the predetermined convergence limit level. The prediction accuracy was encouragingly high : 1357 correct predictions, 98.6% accuracy levels. The results are shown the Table-12.

	Contents	Result
Training	Number of hidden layer	1
	Number of iterations	1944
	Number of hidden nodes in hlayer	8
	Total error(%)	0.00009971
	Recognition error rate(%)	0.0
Test	Number of test pattern	3199
	Number of error pattern	113
	Recognition error rate(%)	0.5250
	Total error(%)	0.0012376
	Accuracy rate(%)	96.47

표 12 :Result of Backpropagation with Stratified (total)

Compared with above result, it was found that stratified method shows a little higher accuracy than non-stratified method. In the stratified method, according to the each level, accuracy level was different. However, the average of each accuracy level was higher than non-stratified method.

	Stratified Method	Non-stratified Method
Contents	99.84(grade1)	96.47(total)
	98.60(grade2)	
	95.11(grade3)	
	100(grade4)	
Accuracy rate(%)	98.64(average)	96.47

표 13 : Stratified method v.s. Non-stratified method

5. Conclusion and Future Research

가. Conclusion

Previous research in Korea showed better performance results than traditional statistical method and others. However, the classification accuracy is still lower than expected, and much lower than the results of experimental research conducted in the States. The lower classification accuracy of the system developed in Korea may be attributable to the noise contained in the data which frequently shows inconsistency. For example, a customer with higher income should be in a better credit position than customers with less annual income. In the States, the customer data show the quasi-linear, if not exactly straightly linear, relationship between annual income and credit status. However, the customers in Korea are widely scattered around average value of the dependent variable. In other words, the spectrum is too broad that no relationship between two variables can be identified.

For the reason, the advanced technique of credit evaluation cannot be usefully applied to the financial institutes for the practical use. Another problem with the credit data collected in Korea is that information of customer's potential credit is not contained in the data set. Because of the highly competitive market situation and aggressive marketing policy of the credit companies to increase market share, the customer data included in the data sheet include only customer name, address, and telephone. Occupation, position, work place, birthday, work phone number, and personal hobby are very important and useful information, but are not included in the data sheet.

For the reason, in this research the prediction of customer's behaviour was conducted, along with the historical data of the customer. Many financial institutes are interested in continuous monitoring of the customer behaviour and then early warning system of possible delinquency. As the result, the final conclusion is not to determine whether the credit application is accepted or rejected. Rather than determining the acceptance or rejection of the application, the system continuously monitor the customer's behaviors and then send a warning signal when the customer's expected financial risk exceeds the threshold value.

In addition to the inconsistency included in the noisy data set, there are many missing data in it. It is pointed out that the behaviour of the Korean customers is quite different from that of the Americans, because of the different culture and custom. To overcome the problems, in this research stratified neural networks were experimented on the relationship between previous credit records and potential delinquency. In this research, it was found and proved that the neural network system can outperforms other classification systems in evaluating and monitoring credit holder's behaviour. Surprisingly, the system showed very higher accuracy, say 98.25% correct prediction for future behaviour of the credit holders. The low classification accuracy of past research in this field may be explained by the inappropriate selection of independent variables and then combination of multiple variables in the same way as the systems developed in the States.

However, the result of this research does not claim that the research result can be directly applied to practical systems and can cure problems of current financial systems; the rate of delinquent cases in Korea is very high, compared with that of other countries and thus much noise is included in the customer data to make prediction complex and difficult. In fact, previous researches in Korea showed that the customer data is too noisy to justify automated credit evaluation [Kim. J. W., Choi. H. Y., J-U Choi, 1992]. However, we believe that the approach experimented in this research can greatly enhance the prediction accuracy of the system and thus grantors can drastically reduce the risk of losing financial resource.

Important findings of this research is that the difference between customer classes is so significant that stratification of the customer data based on the customer class was very effective. Different from previous research, this research did not consider type of job, position, organization, annual income, age, residential area, and other personal data. In general, those personal data have been considered to be important in evaluating credit. However, in a analytical study on customer behaviour it was revealed that the significance of those variables is very low. In contrast with the belief that customer's behaviour is effected by the personal data, such as type of job and position in the organization. One of the possible explanation is that information on the customer type itself include many other personal data, such as organization, type for job, position, and income level, because the customer type is determined by the credit grantors based on the personal data.

Another significant contribution is that the neural network system, when well organized, can greatly enhance the prediction accuracy of the evaluation system. The statistical method which employed multi-variable regression but used the same variables showed the accuracy level of 83.3%, while the neural network system with stratified structure showed accuracy level of 98.25%.

The belief that customer type is very important factor in determining credit risk is supported by the test result that the accuracy level of some classes was higher than 99%, while the accuracy level of other classes was equal to the average rate. This evidence confirms that the research should put heavier weight on the customer type in the analysis. Accordingly, the credit evaluation needs multi-stratified structure to enhance the prediction accuracy.

나. Future Research

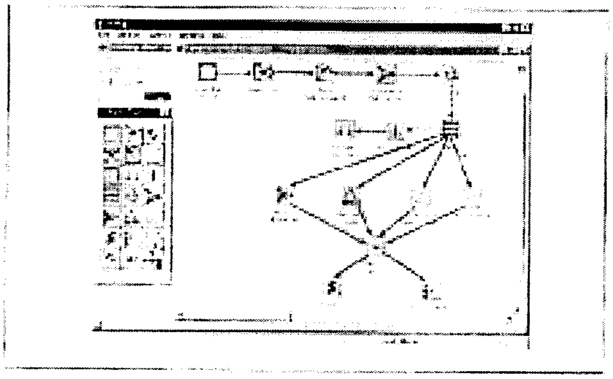
Even though a successful result was obtained in this research, future research is needed in stratification methods and relationship between degree of stratification and performance improvement. Also, a hybrid approach should be experimented. In the hybrid approach, the data can be statistically analyzed for determining factors and degree of stratification and then trained using stratified neural networks. Another approach to the hybrid method is that the customer data can be categorized using neural clustering technique, and then the data can be trained for each cluster.

제3절 국내의 산업 응용

국내에서는 대부분의 신경망 연구가 문자, 음성, 영상 등의 패턴인식에 집중되어 있어서 신용도 평가에 관한 신경망의 응용연구는 김정원, 최종욱, 정윤[1994a, b, c]팀의 신용카드 발급시 개인 신용평가에 관한 연구와 과학원 한인구박사팀의 Bond Rating 에 관한 연구가 있으며 중앙대에서도 일부 연구가 진행되었다. KAIST 테크노경영대학원(한인구 교수팀)에서 "Credit Evaluation & Fraud Detection System" 개발을 주도적으로 이끌어 가고 있으나, 통계적인 기법에 인공지능기법을 결합한 형태이다. 성균관대학의 이건창 교수팀에 의한 신용평가 연구가 꾸준히 이루어져왔으며 국내기업으로는 삼성과 LG, 한국 신용평가 등에서 진행되었다.

국내 Fraud Detection System 은 90년초까지 대부분 통계적인 방법을 이용하여 연구 및 개발 되었었으나, 최근들어 "데이터마이닝을 이용한" Fraud Detection System을 이용한 상품들이 출시되고 있다. 특히, 통계툴로써, 유명한 SAS 및 SPSS에서 제공하는 데이터마이닝을 이용한 Fraud Detection System 은 중소기업

에서 많이 사용되고 있다.



대기업 카드사의 경우, 자체기술력으로 시스템을 개발 및 운영하고 있다. 삼성카드사의 경우, "트라이어드 시스템"이라 불리우는 신용평가시스템을 이용하여, 회원의 한도와 채권 및 개인 신용대출 자격심사등에 적극 활용하고 있다. 이 시스템은 통계적인 방법을 적용하여 지난 1년간 카드 사용내역을 분석후, 향후 불량 발생가능성이 있는지 여부를 예측한다.

엘지카드사의 경우, 인공신경망기법을 적용한 사고성거래 조기검색 시스템을 개발, 운용함으로써, 약 9억원 에 달하는 카드 부정사용으로 인한 손실 절감하고 있다. LG카드에서 사용하고 있는 FDS(Fraud Detection System)을 사용하기전에는 일정기간 동안 카드 사용건수와 금액 빈도수 등을 체크해 결제승인을 하는 방식의 "즉시 대응 시스템"을 운용하고 있었다. 하지만, 즉시 대응 시스템은 부정사용 사전 색출률이 부정사용 발생 대비 금액으로는 21.3%, 건수로는 16.7%에 머물러 부정사용 방지 효과가 크지 못했으나, FDS 사용후 금액면에서 17.52%, 건수면에서 13.11% 개선되는 효과를 거두었다.

국민카드사의 경우, 국민카드사가 숭실대 기술연구소와 공동으로 사고카드 적발시스템을 개발해 운영하고 있다. 현재 국내에서 연구되고 있는 신용평가 기술의 적용분야는 다음과 같다:

- Credit Card Fraud Detection
- Money Laundering
- Securities Fraud
- Phone Fraud
- Banking
- Insurance
- e-Commerce(Creddit Card Holder & Merchnat's Fraud)

Internet Gambling

Lottery

On-Line Banking

Internet Stock Dealing

제3장 Building an Intelligent Frauds Detection System

제1절 An Intelligent Financial Fraud Detection System

This research aims to discover fraudulant patterns in a large insurance and credit card transaction databases. This aim is achieved by developing a hybrid Fuzzy-Genetic programming system. The developed system is able to classify home insurance claims and credit card transactions into "suspicious" and "non-suspicious" classes. This section describes the details of developed system and its evaluation results based on various evaluation criteria.

1. Algorithm Background

가. Fuzzy Rules

Fuzzy rules provide attractive features for individuating classes of phenomena described in data sets(Eberhart et al. 1996). They are easy to understand, verify and extend by allowing a system to represent of 'vagueness' and uncertainty of its rules, which a conventional rule-based system cannot handle. Since fuzzy logic was introduced by Lofti Zadeh in 1965, this feature have made them greatly attractive for use in domains where experts exist who can seed the systems with a number of effective rules from the outset. Even though the successful story about applying fuzzy logic to developing controllers is well-known, its equally successful cases can be found in solving classification problems (Chiu 1997), (Bezdek and Pal, 1992) which is also our research problem.

Another appeal for fuzzy logic is its intelligibility, which is also understood as the comprehensibility of represented rules. Fuzzy rules use directly linguistic terms such as 'high', 'short', 'good' and 'bad', etc. Since the human users of developed system usually think any given concept in such vague terms, the intelligibility of rules, which are represented via fuzzy logic, becomes higher. When we consider that most of users of developed automated financial fraud detection system are not computer system experts, this approach which shows its intelligibility is regarded as one of ideal methods for our research domain. In summary, the combination of representation of uncertainty, precision and intelligibility has motivated the use of fuzzy logic in pattern classification problem (Bezdek and Pal, 1992) and surely drives us to use the same approach in this research.

Despite of its strengths such as uncertainty representation, precision and intelligibility, fuzzy logic does not provide automated learning mechanism. The weakness of conventional fuzzy-rule expert system is that a system developer should define inference mechanisms by using fuzzy logic operators and human expert knowledge and this task is often not easy (Giarratano and Riley, 1994). For instance, most of traditional credit scoring system defines its creidt scoring inference rules depending on human experts' expertise. However, the task to extract subtle knowledge from human experts is known as one of difficult work to be tackled due to its communication or perception problem (Giarratano and Riley, 1994). This leads many researchers to use machine learning algorithms to learn automatically inference rules to solve a given problem (Mitchell, 1997). One of obvious approach to gain distinct strengths both from fuzzy logic and machine learning algorithms is a hybrid system. Typically, clustering is carried out in the N-dimemsional space defined by all records, using Kohonen unsupervised learning algorithms and back propagation methods (Chung& Lee, 1985). Initial rules represented by fuzzy logic are then associated with each cluster center (Ross, 1995). Other approaches which combine fuzzy rules with evolutionary algorithm or neural network are also found in (Marmelstein and Lamont, 1998), (Pedrycz, 1997), (Ross, 1995).

From these previous approaches, we focused on the algorithms presented by Abe and Lan(1993). Their algorithm employs one of popular matching learning approach "separate and conquer", which extracts rules from nested fuzzy cells and the process terminates when the current fuzzy grid contains only homogeneous cells. This nested process is considered as an efficient and fast method to search for a given class concept by shrinking a original large search space gradually. In other words, as search continues, the search space becomes smaller and discovers the concept of a given class quickly.

나. The Evolutionary Algorithms

As described above, our approach to classify home insurance claims and credit card transactions into "suspicious" and "non-suspicious" classes is employing fuzzy rules to represent given examples and search for the concept description commonly existing in these examples. While fuzzy logic was used for representing given examples/rules in this research, a genetic search is adopted as a search mechanism which looks for the hidden concept of each class. It is a biologically inspired algorithm which mimics the natural selection mechanism of nature. Within an evolutionary algorithm, a population of solutions to the problems is maintained with the 'fittest' solutions (those that solve the problem best) being favoured for 'reproduction' every generation. 'Offspring' are then generated from these fit parents using random crossover and mutation operators, resulting in a new population of fitter solutions.

A genetic search is originally well known as its powerful search for optimization problems (Goldberg, 1989). In addition, it has shown the significant performance in rule classification/concept learning domain (Langley, 1996), (Mitchell, 1997). The rule search process by a genetic algorithm are mainly divided into two different approaches. One can either evolve populations of rule set, called the *Pittsburgh approach* or just individual rules, known as the *Michigan approach.*

(1) The Pittsburgh Approach

This approach employs the evolution of rule sets (Smith, 1983). Each individual in a population is a single rule set and thus a genetic algorithm searches for the optimal single rule set which is a collection of rules showing the best performance collectively. Since S. F. Smith at University of Pittsburgh introduced this method, many researchers adopted this technique for concept classification problems (De Jong et al., 1993), (Hekanaho, 1997), (Janilkow, 1993), (Flockhar, 1995) and (Ishibuchi et al, 1998). For more details, each chromosome of a population is a binary string with a fixed size of length equal to the size of rule set. The value, 1, on the nth bit means that the rule is included in that rule set and the value, 0, shows the absence of that rule.

The strength of this approach is that the rule evolution by a genetic algorithm is not driven into a single dominating rule. It naturally provides an optimal rule set which contains multiple rules. In a classification problem, it is often difficult to describe a class concept by a single super rule. The is because there are a number of different peaks in a concept search space and an individual peak should be represented by a single rule of rule set. However, it has serious drawback especially when it is used for handling real data set. Because this approach requires encode all candidate rules into a single chromosome, it leads the length of single chromosome very long and results in the generation of far too complicated search space. For instance, if any rule has ten attributes with five membership functions, $6 * 10^7$ rules would be necessary and this size would be unmanageable to be encoded into a single chromosome and be searched (Mallinson and Bentley, 1999).

(2) The Michigan Approach

Even though the Pittsburgh approach avoids the generation of single dominating rule, it clearly has a severe limit to cope with a real world data set. On the contrary, the Michigan approach takes a chrosome containing only single rule (Booker et al, 1989). This approach suits for larger problems which suffer from the 'combinational explosion' (as the number of fields/dimension increases the search space of rules grows exponentially). In this method, the entire population models a single rule set, where every individual in the population is a single rule. However, the nature of genetic algorithm, which searches for the fittest one of given objective function, leads this approach to coverage into a single fittest rule, rather than a rule set. This feature is also surely inappropriate to describe the multi-peak search space formed by real world data.

To avoid this problem (which is known as the maintenance of rule diversity), several techniques have been suggested. Among them, the most popular methods are bucket-brigade(Goldberg 1989), crowding(Goldberg 1989), fitness sharing(Deb and Goldberg, 1989), sufferiage operators(Giodana and Neri, 1996) and co-evolution(Potter, 1997) etc. Although these approaches have their own slightly different sub-components, the common feature among them, which makes the genetic search converge into multiple peaks, is that indivisuals are somehow grouped and each individual competes with only others belonging to the same group. By doing so, these approaches can maintain a number of individuals which survive by being selected as the fittest one from a given group.

This research uses the Michigan approach in order to handling real world data set, which is collected from a home insurance company and a credit card company. In addition, in order to maintain the diversity of population, this research follows the matching learning approach "separate and conquer", which is described in the previous section. This alternative approach allows the evolutionary classifier converge into a single rule, but performing multiple runs, a rule set built up, one rule at a time. The more details about the genetic search employed in this research will be described in Section 2, The Evolutinary-Fuzzy Evolver for financial fraud detection.

(3) The Genetic Programming

John Koza (1992) developed a genetic programming(GP) for the purpose of automatic programming, which allows computer programs to evolve by themselves. GP differs from other evolutionary algorithsmes in three main aspects: individuals are represented by tree-structure, crossover normally generates offspring by concatenating random subtrees from the parents, and inividuals are evaluated by executing them and assessing their function. Like all evolutionary algorithms, GP maintains poulations of solutions. These are evaluated, the best are selected and 'offspring' that inherif features from their 'parents' are created using crossover and mutation operators. The new solutions are then evaluated, the best are selected, and so on, until a good solution has evolved, or a specific number of generation have passed.

There are some recent works which employe especially GP for the evolution of classification rules (Koza et al, 1998), (Raymer et al., 1996), (Ryan et el., 1998), and (Ryu and Eick, 1996). It takes advantage of the flexibility of GP that can adopt various functions sets, such as negation, larger-than, etc for a rule operator set. This flexibility allows its phenotypes, classification rules, to express more complex conditions and results in producing a smaller number of rules. However, this expressive power often generates rules which are very difficult to understand and thus the intelligibility of evolved rules is very low. This research focused on tackling this problem. Our system chooses the GP as its main rule learning engine mainly because of its expressive power. We provides the intelligibility of evolved rules by controling crossover points and mates for applying genetic operators. The more details about these methods will be discussed in Section 2, The Evolutionary-Fuzzy Evolver for financial fraud detection.

제2절 The Evolutionary-Fuzzy Evolver for Financial Fraud Detection

The system developed during this research comprises two main components: a Genetic Programming(GP) search engine and a fuzzy expert system. The overall system overview is illustrated in figure 1. This chapter describes the system overview of developed fuzzy rule evolver and the details about this system.

1. The System Overview

Before the details of evolutionary-fuzzy evolver system are described, the overview of this system is briefly described. The system starts by taking training data, which has both "fraudulant" and "non-fraudulant" class examples. The training data is immediately passed to the clusterer and the clusterer classifies each attribute(=column) values into three groups. The system describes each attribute value with one of these fuzzy sets: 'LOW', 'MEDIUM' and 'HIGH' and the clustering stage is necessary to define the possible attribute value ranges of these three fuzzy sets. The cluster passed three generated clusters of each attribute to the fuzzy expert system. The fuzzy expert system defines fuzzy membership functions and their possible ranges based on the generated clusters.. The fuzzy expert system fuzzifies the attribute values of each training item according to the fuzzy membership functions. Then, a GP engine starts evolution by being seeded with random genotypes. The generated genotypes are mapped into phenotypes, which are fuzzy rules. These fuzzy rules are evaluated by being applied to the fuzzified training data in the fuzzy expert system. The results of this procedure return to the GP engine as the defuziffied scores and these scores are evaluated by fitness functions. The estimates fitness scores allow the GP to select fitter phenotypes and reproduce their genotype offsprings. These new genotypes are passed the fuzzy expert system and repeates their evaluation, selection and reproduction until the evolved rules show the certain performance or maximum number of repeats(=generations).

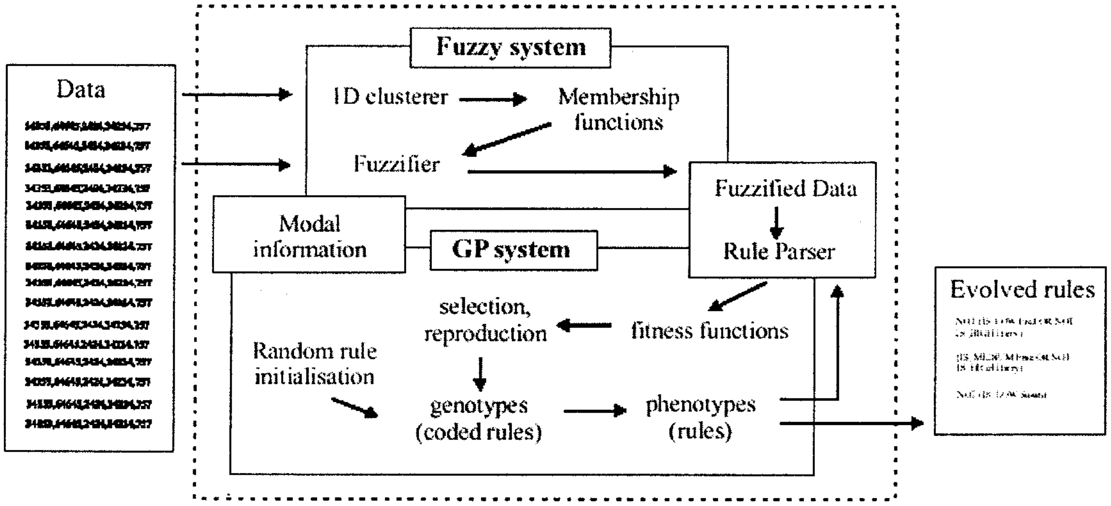


그림 4 :Block diagram of the Evolutionary-fuzzy system.

Clustering

When started, the system first clusters each column of the training data into three groups using a one-dimensional clustering algorithm. A number of clusterers are implemented in the system, including C-Link, S-Link, K-means (Hartigan, 1975) and a simple numerical method (in which the data is sorted, then simply divided into three groups with the same number of items in each group). This paper investigates the last two of these methods in the system. Once selected by the user, the same clusterer is used for all learning and testing of the data.

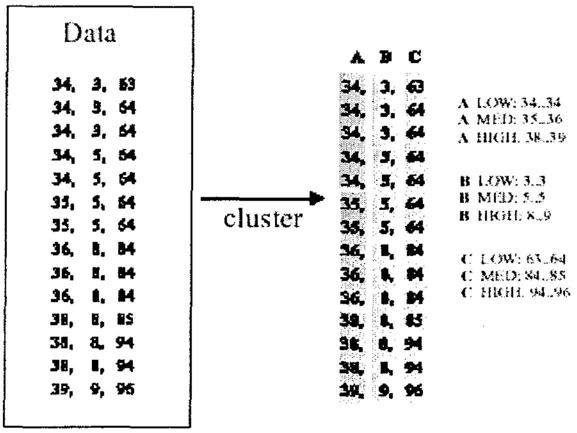


그림 5 : Data is clustered column by column to find the fuzzy membership function ranges.

After every column of the data has been successfully clustered into three, the minimum and maximum values in each cluster are found, see Figure-5. These values are then used to define the domains of the membership functions of the fuzzy expert system.

2. Define Fuzzy Membership Functions

The fuzzy rules are used in this resarch for increasing the intelligibility and we select the simplest and the most comprehensive three fuzzy sets.: 'LOW', 'MEDIUM' and 'HIGH'. In order to use these liguistic terms directly to collected numeric data, the clustering stage described in the previous section was necessary. The data attributes have wide ranges of values and they are clustered into three groups using a k-mean clustering algorithm. Since each attribute has different range of values and features, the ranges of generated clusters for an individual attribute are very diffrent The three cluster ranges of single attribute are used for defining the 'degree of membership' of three fuzzy sets for the attribute.. This results in providing various ranges of fuzzy set definition (more precisely fuzzy membership function values) according to a given attribute.

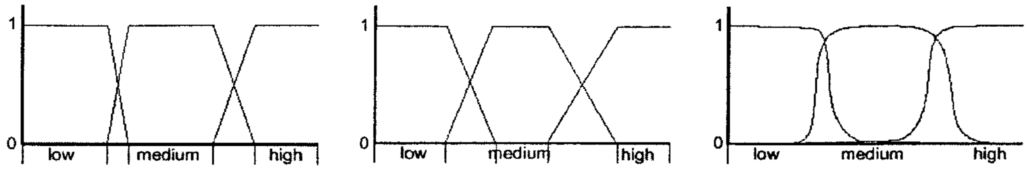


그림 6 :The three types of membership functions used by the system: non-overlapping (left), overlapping (middle), smooth (right).

As we introduced before, the degree of membership for each fuzzy set is defined by a given fuzzy membership function and there are several different shapes of fuzzy membership functions. The system developed in this research provide three various fuzzy membership functions: 'non-overlapping', 'overlapping' and 'smooth', shown in figure 6. The first two functions are standard trapezoidal functions and the third one returns a smoother, more gradual set of 'degree of membership' by taking the arctanget of the input.

The different function shapes of these three different functions determines the various definition of each fuzzy set. For the 'non-overlapping' functions, when a specific attribute value belongs to a cluster, they return the membership value 1.0 to the corresponding fuzzy set and 0.0 to the other two fuzzy sets. For instance, if an attribute value is included in the cluster having the lowest range of values, this value would be fuzzified into (1.0, 0.0, .0.0) for 'LOW', 'MEDIUM' and 'HIGH' fuzzy sets, respectively. Because the shape of 'non-overlapping' fuzzy membership function restricts the output areas between two adjacent fuzzy sets to be nearly non-overlapped, it is very usual for given attribute value falling into strictly only one fuzzy set. In other words, it generates the identical fuzzy membership values (1.0, 0.0, .0.0), (0.0, 1.0, 0.0) or (0.0, 0.0, 1.0) to most of attribute values.

In contrast, the second fuzzy membership 'overlapping' function widen the overlapping areas between neighboring fuzzy sets and more versatile fuzzified membership values can be expected. This means that a value towards the outer degree of the 'LOW' fuzzy set might be fuzzified into (0.8, 0.2, 0.0) instead of (1.0, 0.0, 0.0). Finally, the last 'smooth' functions expands the overalpping areas among three fuzzy sets even more. For instance, even when a value resides on the center of 'LOW' fuzzy set area, this fuzzy membership function returns the fuzzified value (0.98, 0.02, 0.0) signaling somehow the value locating at a much nearer point from the 'MEDIUM' set than the 'HIGH' set.

In summary, when a training example is provided to the system, it fuzzifies all attributes of this example according to a selected fuzzy membership function. and the fuzzy set ranges are determined by clusterers generated using the k-mean clustering algorithm.

3. Evolving Rules

When fuzzified data is passed to a GP engine and the fuzzy rule evolver starts the rule evolution, The rule evolution by the GP is required to pass a number of important stages to achieve its task, generating fuzzy rules classifying fraudulant data and non-fraudulant data.

가. Genotypes and Phenotypes

The operation of natural evolution can be understood by the evolution of a set of coded instructions for how organisms should be grown (Bentley, 97). That is to say, the genetic operators to drive the natural evolution are not applied directly on organisms. Rather, they are applied on a set of coded instructions: DNA. According to these instructions, the organisms evolve and appear to have various types of patterns. In order to understand the natural evolution clearly, it is necessary to distinct DNA from the evolved organisms based on their DNA. The former is known as a genotype and the latter is called as a phenotype.

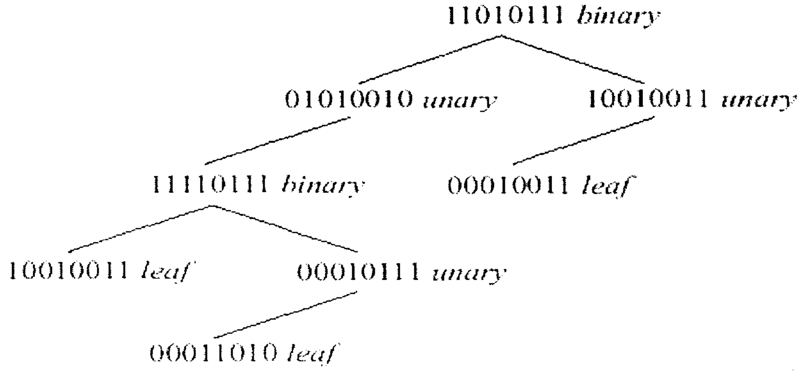


그림 7: An example genotype used by the system.

The genotypes consist of variable size trees, where each node is comprised of a binray number and a flag indicating whether a node is binary, unary or a leaf. A binary node has two brabches (left and right), a unary node has one branch (left or right) and a leaf node is a terminating node of given branch. When evolution starts, genotype genes with tree structures are ranomly generated and the created genotype genes are usually have no more than three binary and four unary nodes. We restrict of tree size at the start to avoid the generation of unneccerily long depth of trees. These genotype genes are immediately mapped into phenotype genes in order to be evaluated. Figure-7 shows the mappinig of genotype genes in Fig. 6 into the phenotype genes, which are in a form of fuzzy rules:
IS_MEDIUM (Height OR IS_LOW Age) AND IS_MEDIUM Age).

The system employes two binary functions: 'OR' and 'AND', four unary functions: 'NOT', 'IS_LOW', 'IS_MEDIUM', 'IS_HIGH'. Each leave of tree indicates a single attribute and the system restricts the number of leaves up to 256. An individual genotype tree is easily interpred into a fuzzy rule by reading a given node type and translate binary value into tenary value.

나. Rule Evaluation

Every chrosome containing phenotype genes, which is a fuzzy rule, is evaluated by applying it to the fuzzified training data. This work is performed by a fuzzy expert system. The results of this work is returned in a defuzzified score between 0 and 1 for every fuzzified data item. Then, these defuzzified scores are assessed by four different fitness functions:

- Low misclassification rate: minimize the number of misclassified examples. The misclassified examples are those which have the defuzzified scores larger than 0.5 when they are 'non-fraudulent' items. In fact, this is the case when 'non-fraudulent' case is misclassified as 'fraudulent' case. The system regards the item with the defuzzied score close to 1 as 'fraudulent' case.

- Maximize the distinction between 'non-fraudulent' and 'fraudulent' classes: to distinct two classes claserly, this fitness function measures the average defuzzified scores for correctly classified 'fraudulent' cases and the average defuzzified scores for correctly classified 'non-fraudulent' cases. As the difference of these two average scores increase, this fitness score becomes higher.

- Assign a high priority to detect 'fraudulent' items more: the system considers the detection of 'fraudulent' cases more importantly. Thus, this fitness function assess the sum of scores for 'fraudulent' cases and assigns higher fitness value when this score gets higher.

- Increase the intelligibility: in order to increase the intelligibility, the system penalise the length of any fuzzy rules that has more than four identifier, which are binary, unary or leaf nodes. By doing so, the system ensures that the evolved rules always have the readable length of condition parts and also can prevent the bloat caused by the GP.

One distinct feature of our evolutionary fuzzy evolver is that it has multiple fitness functions to collectively satisfy the desired function of the system. Most of real world problems require more than one subtask to be fulfilled due to its complicated nature. However, this feature causes a major problem for the standard GA. It usually cannot cope with more than one fitness value per phenotype(Goldberg, 1989). The standard GA equips only one fitness value for every individual in the current population of solutions. This value represents how well its corresponding solution satisfies the goal of a given problem. According to the estimated fitness values, the GA can select fitter ones and gives higher chances to reproduce their offsprings which inherit some features of original solution. The problem is raised when the GA tries to select the fitter ones based on calculated fitness values. In order to do so, the GA should apprehend the single relative fitness value of each candidate solution for comparison even when each individual has multiple fitness values. The question is how can we make the GA to define a single fitness value that represents the aggregation of multiple fitness values accurately ?

Bentley and Wakefield(1997) developed the Sum of Weighted Global Ratios(SWGR) method to tackle this problem and showed its successful results in their evolutionary design system. The system developed in this research also employs this approach. To define the overall fitness values from multiple fitness values, the SWGR first scales each fitness value using the *effective ranges* of each function. Since the input of each fitness function shows different range of possible values, this scaling is perforemed first. For example, for a given individual I, the scaling is simply done by

$$fitnessRatio = \frac{fitnessValue_i - \min(fitnessValue)}{\max(fitnessValue) - \min(fitnessValue)}$$

Then, the normalized fitness values are multiplied by *importance* values defined by system users. The basic notion of importance is that even though the final goal is achieved by satisfying the multiple subtasks, the relative importance of each subtask will be different and this difference can be defined by the users according to their perceived task priority. For instance, if the user specified that the second criteria should be twice as important, all fitness ratios corresponding to the second criteria are simply multiplied by two. Finally, SWGR sums these importance weighted global fitness ratios and generates the single global fitness values.

다. Rule Generation

○ Genetic Operators

Child rules are generated using one of two forms of crossover. The first type of crossover emulates the single-point crossover of genetic algorithms by finding two random points in the parent genotypes that resemble each other, and splicing the genotypes at that point. By ensuring that the same type of nodes, in approximately the same places, are crossed over, and that the binary numbers within the nodes are also crossed, an effective exploration of the search space is provided without excessive disruption (Bentley & Wakefield, 1996). The second type of crossover generates child rules by combining two parent rules together using a binary operator (an AND or OR). This more unusual method of generating offspring (applied approximately one time out of every ten instead of the other crossover operator) permits two parents that detect different types of suspicious data to be combined into a single, fitter individual. Mutation is also occasionally applied, to modify randomly the binary numbers in each node by a single bit.

○ Selection

The GP system employs population overlapping, where the worst *Pn%* of the population are replaced by the new offspring generated from the best *Pm%*. Typically values of *Pn* = 80 and *Pm* = 40 seem to provide good results. The population size was normally 100 individuals.

○ Modal Evolution

Each evolutionary run of the GP system (usually only 15 generations) results in a short, readable rule which detects some, but not all, of the suspicious data items in the training data set. Such a rule can be considered to define one mode of a multimodal problem. All items that are correctly classified by this rule (recorded in the modal database, see figure 4) are removed and the system automatically restarts, evolving a new rule to classify the remaining items. The parameter *nichsize* specifies the number of "fraudulent" data items sought to be classified in each run. This enables monitoring of over-fitting: the fitness is correlated with the number of class members classified by a rule as well as the number miss-classified.

○ Modal Re-Evolution

In addition to the process of modal evolution, the system re-examines each mode already classified by a rule; it attempts to improve the rule by ignoring all data except that characterized (and mis-classified) by the rule already. This provides a shrinking environment, with the associated gains: a 'purer' gene pool of solutions for each archetype is facilitated, and accelerated search.

○ Nested Evolutionary Search

After shrinking the environment(reducing the number of claims again which a rule is tested) the system can recluster and carry out a finer search. This process of modal evolution continues until every suspicious data item has been described by a rule. However, any rules that misclassify more items than they correctly classify are removed from the final rule set by the system.

라. Assessment of Final Rule Set

Once modal evolution has finished generating a rule set, the complete set of rules (joined into one by disjunction, i.e., ORed together) is automatically applied to the training data and test data, in turn. Information about the system settings, number of claims correctly and incorrectly classified for each data set, total processing time in seconds, and the rule set are stored to disk.

4. Applying Rules to Fuzzy Data

The path of evolution through the multimodal and multicriteria search space is guided by fitness functions. These functions use the results obtained by the Rule Parser - a fuzzy expert system that takes one or more rules and interprets their meaning when they are applied to each of the previously fuzzified data items in turn.

This system is capable of two different types of fuzzy logic rule interpretation: traditional fuzzy logic, and *membership-preserving* fuzzy logic, an approach designed during this research. Depending on which method of interpretation has been selected by the user, the meaning of the operators within rules and the method of defuzzification is different.

가. Traditional Fuzzy Logic Rule Parser

Traditional fuzzy logic involves finding degrees of membership in the fuzzy sets for each value in the current data item, then using operators to select which membership value should be selected and used in combination. So, given a data item comprising two fuzzified values:

A(0.0, 0.2, 0.8)

B(0.1, 0.9, 0.0)

and a fuzzy rule:

(IS_LOW A AND IS_MEDIUM B)

the traditional fuzzy rule parser takes the degree of membership of A for fuzzy set LOW and the degree of membership of B for the fuzzy set MEDIUM, and calculates which of the two is smaller. So in this case, the result of applying the rule is 0.0. Table 1 describes the behaviour and syntax of each of the fuzzy operators.

Operator	**Result**
IS LOW <a, b, c>	a
IS MEDIUM <a, b, c>	b
IS HIGH <a, b, c>	c
NOT a	1-a
(a AND b)	min(a,b)
(a OR b)	maz(a,b)

표 14 : Traditional fuzzy operators.

This fuzzy grammar imposes certain constraints upon allowable solutions. For example, the argument to IS_LOW, IS_MEDIUM or IS_HIGH must always consist of a fuzzy vector: $<Low_{membership}, Medium_{membership}, High_{membership}>$. The arguments to AND, OR and NOT functions must always be single-valued results obtained from the application of one or more of the functions.

As is clear from the example phenotype, evolved rules do not always satisfy the constraints imposed by fuzzy grammars. However, rather than impose these damaging constraints on evolution, such grammatically incorrect rules are corrected by the rule parser. (Work performed during this research showed that using mapping to satisfy constraints in a GP system is one of the more effective approaches (Yu & Bentley, 1998).)

Operator	**Result**
<a, b, c>	IS HIGH <a, b, c>
IS LOW a	a
IS MEDIUM a	a
IS HIGH a	a

표 15 :Mapping performed by the Rule Parser.

Functions requiring a fuzzy vector, but receiving a single value do nothing.

Functions requiring a single value, but receiving a fuzzy vector, apply IS_HIGH by default in order to generate the single value. Table 15 describes this behaviour in full. Consequently, when interpreted by the fuzzy rule parser, the rule in section 3.가 equates to:

((IS_HIGH Height OR IS_LOW Age) AND IS_MEDIUM Age).

Defuzzification of the final output value is unnecessary (although it is possible to impose a scaling, or non-linear function to transform the output in some way). It was decided simply to use a one-to-one function for defuzzification (i.e., return the output of the fuzzy rule as the defuzzified value).

나. Membership-Preserving Fuzzy Logic Rule Parser

The alternative behaviour of the rule parser preserves the three membership values within data items, even after the application of operators such as IS_LOW. This is done in an attempt to permit rules to use all the information found by the clusterer, and thus hopefully to reduce the number of rules needed to classify data. In addition, the operators are designed to be more conducive to evolution by allowing multiple operators to have combined effects without constraints on syntax. For example: the alternative behaviour of the rule parser preserves the three membership values within data items, even after the application of operators such as IS_LOW. This is done in an attempt to permit rules to use all the information found by the clusterer, and thus hopefully to reduce the number of rules needed to classify data. In addition, the operators are designed to be more conducive to evolution by allowing multiple operators to have combined effects without constraints on syntax. For example:

IS_HIGH IS_HIGH $\underline{v}$ is now equivalent to   IS_VERY_HIGH $\underline{v}$

It should be noted, however, that the English descriptors for these operators does not always fully encompass their behaviour in a rule. Table 16 shows the new behaviors of the operators.

Operator	**Result**
<a, b, c>	<a, b, c>
IS LOW <a, b, c>	Conc. <c, b, a>
IS MEDIUM <a, b, c>	Conc. <0, max(a,c), b>
IS HIGH <a, b, c>	Conc. <a, b, c>
NOT <a, b, c>	<c, b, a>
(<a, b, c> AND <d, e, f>)	min(<a, b, c>,<d, e, f>)
(<a, b, c> OR <d, e, f>)	max(<a, b, c>,<d, e, f>)
Where Conc. *concentrates* the vector (making the largest value larger and the other two values smaller).	

표 16 : Membership-preserving fuzzy operators.

Because this novel approach preserves all three membership values during the application of all operators (although the values may be intensified or reduced), the final result is also a vector comprising three values. To obtain a single, defuzzified value, three defuzzification functions are applied, using the vector to define three trapezoidal shapes, see figure 8. The shapes are then piled up on top of each other and the centre of mass calculated (using overlapping shapes results in a loss of information). A centre of mass falling in the centre results in an output of 0.5, falling to the right gives a score between 0.5 and 1.0, and if the centre of mass falls to the left, the final defuzzified value is between 0.5 and 0, see figure 9.

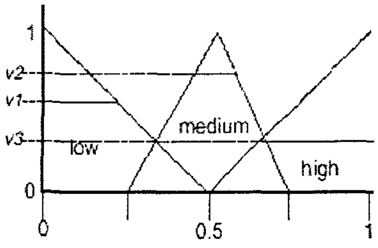


Figure-8: Defuzzifying the three membership values *<v1, v2, v3>*

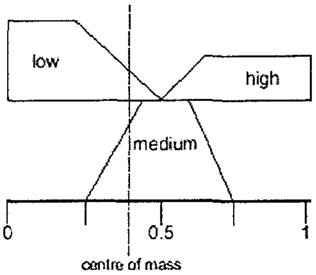


Figure-9: Finding the centre of mass during defuzzification.

The membership-preserving (M-P) fuzzy logic is designed to make use of overlapping membership functions. Indeed, for non-overlapping functions, the behaviour of the M-P fuzzy operators becomes largely identical to the traditional operators.

5. Committee Decisions

As should now be apparent, the evolutionary-fuzzy system has a number of very different elements that can be used at any one time. The choice of clusterer, membership functions, fuzzy interpreter, fitness functions and GA settings can cause varying degrees of success for different input data. What may be a good setup for one data set is not so good for another. In addition, previous work has identified the need for results to be both intelligible and accurate, so multiple results generated by multiple different system setups need to be assessed against multiple criteria.

To achieve this, the system has been extended into a multi-model decision aggregation system. The user can now set up as many as four different versions of the system and have them run in parallel on the same data set, for a user-defined number of times. On completion, the committee decision maker analyses all results written to disk by the different systems, writing the analysis and recommendation of the best evolved rules to disk, see figure 6. The separate evolutionary fuzzy systems (or committee members) have been modified to allow efficient parallel processing (e.g., reading of the data files is performed one at a time on a first-come-first-served basis to avoid excessive disk thrashing caused by simultaneous accessing).

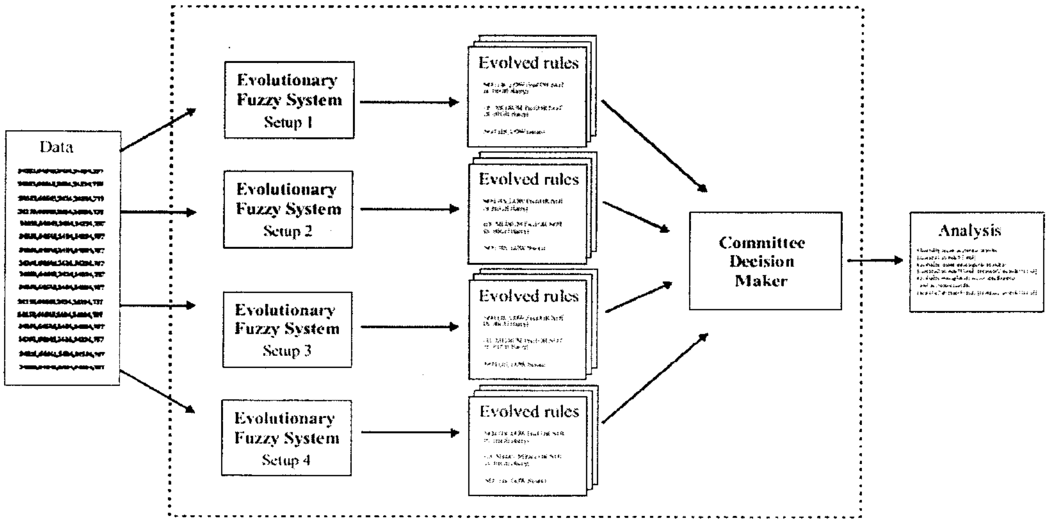

Data
Evolutionary Fuzzy System Setup 1
Evolutionary Fuzzy System Setup 2
Evolutionary Fuzzy System Setup 3
Evolutionary Fuzzy System Setup 4
Evolved rules
Evolved rules
Evolved rules
Evolved rules
Committee Decision Maker
Analysis

그림 8 : Block diagram of the committee decision system

Three simple forms of analysis are automatically performed by the committee decision maker. First, the rule sets generated by each committee member are examined separately. The most accurate rule set(s) (measured using the number of items in the first class correctly classified) and the most intelligible rule set(s) (where fewer rules = more intelligible) are found. The most accurate *and* intelligible rule set(s) are then chosen for each committee member using decision aggregation.

In a similar way to (Bunn, 1989), the committee decision maker employs aggregation of weighted normalised values. In other words, each rule set is given a score *s*, where:

$$s = w1\left(\frac{a - \min(a)}{\max(a) - \min(a)}\right) + w2\left[1 - \left(\frac{p - \min(p)}{\max(p) - \min(p)}\right)\right]$$

w1 is the importance weighting for accuracy,

w2 is the importance weighting for intelligibility,

a is the accuracy of the current rule set (higher values are better),

p is the intelligibility of the current rule set (lower values are better).

To force the different effective ranges of the multiple criteria to be commensurable (Bentley & Wakefield, 1997), the accuracy and intelligibility values are normalized (and the intelligibility value is inverted) before being weighted. Using information provided by Lloyds TSB, the default weighting values were 0.3 and 1.0 for accuracy and importance, respectively.

Once every rule set has been assigned a score, the set(s) with the highest score for each committee member are reported to the user. The committee decision maker then performs the same analysis globally, finding the globally most accurate and intelligible rule set(s), then assigning every rule set a score based on globally aggregated, weighted, normalized values. The best overall rule set(s) are then reported to the user. Finally, a histogram of field occurrences in all evolved rule sets is automatically constructed. As will be shown later, this provides a clear picture of which fields are most important for classification of the data.

제3절 The Criteria For Fraud Detection

The research described here is being carried out with the eventual aim of the detection of suspicious home insurance claims. This difficult real-world classification task does not simply involve finding the most accurate method for distinguishing between ordinary and dubious data items. There are, in fact, more important criteria for evaluating the performance of a technique. Table 18 shows the four capabilities considered to be most important by our collaborating company Lloyds/TSB, with importance rankings.

Feature	Importance
Intelligibility of classification rules	1
Speed of classification	2
Handling noisy data	2
Accuracy of classification	3

표 18 : Important features of a good fraud-detection system.

It may be surprising to note that accuracy is considered less important than intelligibility. However, for this type of application, an expert must review all suggestions made by the classifier (wrongly accusing anyone of fraud is a serious and potentially libellous activity, so the computer should be used only to identify the possibility of suspicion to experts). If the person cannot find an easily understandable explanation of why a particular data item has been labelled as suspicious, then the result is of little use, regardless of the reported accuracy of classification.

Speed of classification is also essential, for most real-world financial problems of this type involve an enormous quantity of data. Increasingly it is becoming necessary for learning techniques to be performed in real time (as new data arrives), but at the very least, the detection method must be fast enough to keep up to date, and also fast enough to justify its use at all. The ability to handle noisy data was ranked equal in importance with speed. Input errors, omitted 'data, or conversion problems may cause noise in the data. Although such noise is unlikely to affect more than a small percentage of values in the data, it is clearly important that the classifier is not misled by any occurrence of noise. Other important considerations include minimising the misclassifications by the system - it is considered better to miss a few dubious data items than to misclassify normal data. It is clearly not good for customer relations if too many people are wrongly investigated for potential wrong-doing. This is the reason for the inclusion of the first fitness function, described in section

제4절 Experiments and Results

1. Data

As with any real-world problem, classification of real data is often far removed from the clean, perfect world of mathematical theories. Data is usually noisy, inconsistent and sometimes inadequate. Even though intelligent techniques such as GP and fuzzy logic can handle such characteristics better than many approaches, significant data preprocessing will always be required.

가. Preprocessing the Data

Lloyds/TSB INSURANCE DATA

The insurance data used for this work was no exception. The data came from numerous sources within the bank, resulting in two somewhat incompatible files. One file contained 98 cases of suspicious insurance claim, each with 73 fields (this was assembled from numerous different files provided). The other contained 20,000 cases of unknown insurance claims (that might or might not be suspicious), each with 36 fields. The fields comprised items such as policy number, claim number, date of birth, policy type, etc. However, the two files had very few fields in common. Even after constructing some new fields by processing others in different formats, only 14 common fields in both files could be found.

Once all non-corresponding fields were removed, we were left with two files, one containing 98 claims, each with 14 fields, the other containing 20000 claims, each with the same 14 fields. The data for every pair of fields was then converted into the same format (for example, dates were initially stored in different formats, different codes were used, etc). Missing values in the files were replaced by random values within the range of normal values for each field. (Attempting to classify data with missing values is difficult, so it is simpler to fill the gaps with random values. This has the effect of adding a small level of noise to the data in this case 1.07% overall. However, the distribution of missing values, and hence noise per field was not even it varied from 0% to 17%. By keeping a record of the percentage noise per field, the reliability of evolved rules that use the noisiest fields can then be reduced. Note that additional noise in the form of errors within the data was also evident.)

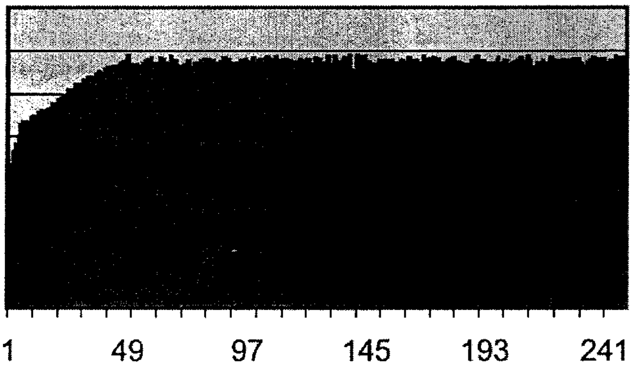


그림 10 : Chart showing the first 250 values, for a field related to the date. Note how the suspicious values in the first 49 are much lower on average than the unknown values in claims 50 upwards.

In an attempt to extract more information from the data, and give the classifier a better chance of success, six new fields were created by processing existing fields. For example a new field called days before claiming was constructed by subtracting values in the field accident date from the values in the field notified date.

A training and test data file was constructed, each containing 49 suspicious claims and 10000 unknown (alternate claims taken from the original files). A series of experiments were then performed using the evolutionary fuzzy system. The results were suspiciously good indeed, accuracy was 100%. From these experiments it became clear that inconsistencies in the data were proving considerably more useful as indicators of fraud than anything else. The disparity was mainly caused by the fact that the 98 cases of suspicious claim were gathered over a period of some years, whilst the unknown data was gathered over a recent period of three months. Any field that varied according to the date was therefore lower, on average, for the suspicious fields compared to the unknown. By plotting charts of each field, it was simple to discover that this adversely effected six of fields, e.g. see Figure-10.

While it is possible that variations on the frequency of fraud may depend on absolute values of dates (e.g. perhaps fraud becomes more likely during a particular month of a year, or following a television programme on how to do fraud), this was seen as unlikely. It was therefore more desirable to attempt to find more generic indicators of fraud, not those dependent on absolute times or specific policy numbers. Consequently, all six fields were deleted (and a seventh which had the same value for all claims was also removed), leaving thirteen fields in each data item. Information was not lost, however. The new fields mentioned earlier contained *relative* date information, so the data contained within five of the deleted fields was still available (with the benefit that the time biases were removed, as differences between fields were used, rather than absolute values).

○ Domestic Credit Card Company DATA

The data used in this work was gathered from a domestic credit card company. Even though the company provided real credit card transaction data for this research, it required that the company name was kept confidential. The data was gathered from January to December of 1995 and a total of 4000 transaction records were provided, each with 96 fields. 62 fields were selected for the experiments. The excluded 34 fields were regarded as clearly irrelevant for distinguishing the credit status. (Examples include the client code number and the transaction index number.) The details of selected field names were not allowed to be reported. In order to allow the fuzzy rule evolution of the system, the collected data was labeled as suspicious or non-suspicious. These labels were made by following the heuristics used in the credit card company. Specifically, when the customers payment is not overdue or the number of overdue payment is less than three months, the transaction is considered as non-suspicious, otherwise it is considered suspicious.

To prepare a training set and a test set, we employed a simple cross-validation method. We held one-third of the data for testing and used the remaining two-thirds for training. The system executed its rule-evolution three times on three different training data sets. For each run, the system replaced the training set with the other third of the data set. This cross-validation was performed in order to ensure the evolved rule sets were not biased by a certain group of training set. By comparing the three different evolved rules based on three different groups of training data set, the final rule set is expected to represent the features of the entire data set. Unfortunately, the distribution of collected credit card transaction data was not even for each class. It had a larger number of examples for the "non-suspicious" class than for the "suspicious" class. The total number of items belonging to the smaller size of "suspicious" class was 985. This number is large enough to be divided into three subsets. Thus, the four committee members with identical experiment setups were run three times on each data subset respectively. The examples included in each set are shown in Table-19.

Exp	"SUSPICIOUS"		"NON-SUSPICIOUS"	
	Training	**Test**	**Training**	**Test**
1	1-656	657-985	1-2000	2000-3015
2	329-985	1-328	1001-3015	1-1000
3	657-985 & 1-328	329-656	2001-3015 & 1-1000	1001-2000

표 19 : Credit card data distribution for three experiments. The number in this table shows the IDs of examples belonging to each set. Exp stands for the experiment.

2. Experiments

With the requirements for a good fraud-detection system in mind, this section describes a series of experiments designed to evaluate these key capabilities of the system.

The experiments investigate three aspects of the system: the effect of using different membership functions and fuzzy operators, the effect of using different clusterers, and the ability of the system to cope with noisy data. For all three sets of experiments, the intelligibility of results, processing time, and accuracy of detection are assessed.

가. Experiment Setup

To allow comparison of this system with other techniques reported in the literature, the fuzzy rule evolver was applied to two standard data sets for all experiments: the Iris and Wisconsin Breast Cancer data sets.

The Iris data is perhaps the best known database to be found in the pattern recognition literature according to the information provided by UCI with the data and it comprises a simple domain of 150 instances in three classes, each of 50 items. Data items have four attributes; there are no missing values. Because the 'Setosa class' is linearly separable from the other two classes, for all experiments the system was set the harder task of detecting the 'Virginica' class from the 'Versicolour' and 'Setosa' classes combined. Training and test data files were prepared by splitting the data set into two (taking alternate data items for each file). Misclassification rates for this data set are normally reported as 0% for the Setosa class and very low for the other classes in the literature e.g. (Dasarathy, 1980).

The Wisconsin Breast Cancer data is a more complex data set, comprising 699 instances in two classes: Malignant (241 data items) and Benign (458 items). There are 16 missing values in the data, which were filled by random numbers. The training and test data sets were constructed by splitting the file into two, taking alternate values. (For the sake of symmetry, one Malignant item was discarded and two Benign items duplicated, resulting in two sets of 350 data items, each with 120 Malignant.) Results reported in the literature include accuracies of 93.5%, 95.9% (Wolberg, and Mangasarian, 1990), and 92.2% (Zhang, 1992).

50 trials were run for each experiment, with the average and best accuracies reported here. Percentage accuracy of detection was found by calculating:

$$100-\frac{100(MisclassifiedItems+Unclassifieditems)}{TotalItems}$$

Intelligibility was measured in terms of the average number of rules evolved - the fewer the rules, the more intelligible the result. Average processing speed was measured in seconds (and includes the negligible time taken to apply the completed rule set to both data sets).

The fitness functions reported were used without change for all experiments. Importance rankings (Bentley & Wakefield, 1997) were set as 0.5, 2.0, 1.0 and 0.5 for fitness functions one to four, respectively. Mutation of a single bit occurred with a probability of 0.001 in each genotype. Population sizes of 100 were used, and each modal evolutionary run was for exactly 15 generations. The K-Means clusterer was used in the system (unless otherwise stated). Experiments were run on a PC with a 233Mhz AMD K6 processor.

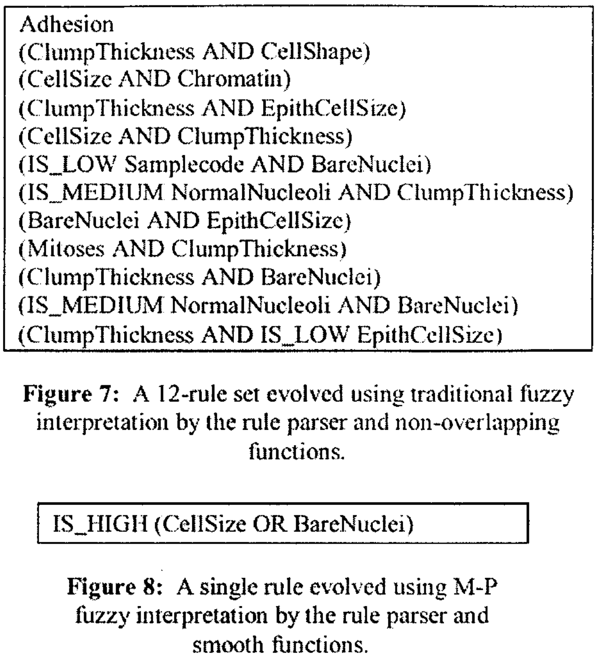
나. EXPERIMENT 1: investigating The Effects Of Membership functions

The objective of the first set of experiments was to examine the effects of different membership functions (and different ways of using the information contained in the membership functions) on the ability of the system to detect data items with good intelligibility, speed, and accuracy. Four different system set-ups were investigated: traditional fuzzy logic with non-overlapping and overlapping membership functions, and M-P fuzzy logic with overlapping and smooth membership functions. (Traditional fuzzy logic does not work well with the level of overlap provided by the smooth functions, and the M-P fuzzy logic with non-overlapping functions behaves in the same way as traditional fuzzy logic, so these set-ups are not investigated here.) Table-20 to Table-25 shows the results obtained from 50 runs of each system set-up for each data set.

	Iris data				Cancer data			
System:	Av. accuracy	Best accuracy	Av. time	Av. # of rules	Av. accuracy	Best accuracy	Av. time	Av. # of rules
FL, non-overlapping MFs	96.67% **95.79%**	**97.3%**	25.6 secs	2.94	98.6% **94.07%**	**96.0%**	317 secs	9.88
FL, overlapping MFs	91.3% **94.69%**	**96%**	13.6 secs	1.50	93.01% **90.44%**	**96.29%**	335 secs	7.16
M-P FL, overlapping MFs	96.05% **90.67%**	**90.67%**	15.1 secs	1.04	91.19% **86.93%**	**92.57%**	175 secs	4.58
M-P FL, overlapping smooth MFs	82.69% **82.59%**	**88%**	16.2 secs	1	95.14% **95.71%**	**95.71%**	162 secs	1

표 25 : Mean and best accuracy rates, processing times and intelligibility of solutions when using different membership functions and fuzzy operators. (Accuracy values in normal intensity indicate results for the training set, bold values show results for the test data set.)

As shown in Table-20 to Table-25, for both data sets the average accuracy appears to fall as the level of overlap of membership functions is enlarged. However, there is clearly a quite dramatic increase in intelligibility (a reduced number of rules) as the overlap increases. This is illustrated by two example solutions evolved by the system for the Wisconsin Breast Cancer data set. Figure-7 (in Figre-12) shows a typical 12-rule set evolved when using traditional fuzzy logic and non-overlapping membership functions. Figure-8 (in Figure-12) shows a typical single rule evolved when using M-P fuzzy logic with smooth membership functions. It should be apparent that the latter is substantially more intelligible than the former. Not only that, but by reducing the number of rules, far more effective feature-selection takes place (i.e., instead of using all ten fields in the data, the single rule shows that only two are required).



Adhesion
(ClumpThickness AND CellShape)
(CellSize AND Chromatin)
(ClumpThickness AND EpithCellSize)
(CellSize AND ClumpThickness)
(IS_LOW Samplecode AND BareNuclei)
(IS_MEDIUM NormalNucleoli AND ClumpThickness)
(BareNuclei AND EpithCellSize)
(Mitoses AND ClumpThickness)
(ClumpThickness AND BareNuclei)
(IS_MEDIUM NormalNucleoli AND BareNuclei)
(ClumpThickness AND IS_LOW EpithCellSize)

Figure 7: A 12-rule set evolved using traditional fuzzy interpretation by the rule parser and non-overlapping functions.

IS_HIGH (CellSize OR BareNuclei)

Figure 8: A single rule evolved using M-P fuzzy interpretation by the rule parser and smooth functions.

그림 12: Figure-7 and Figure-8

It is clear that accuracy is reduced as the number of rules that classify the data is reduced. The exception to this is the MP-FL system with smooth MFs applied to the cancer data, which generated both accurate results with a very intelligible single (and simple) rule, e.g. fig 8. However, this result seems likely to be more the exception than the rule the accurate result may well be due to a fortunate combination of placement of membership functions, and the combination of the three fuzzy membership values for this particular problem. Nevertheless, the result certainly indicates that it is possible to classify real-world problems with both accuracy and intelligibility.

As Table-20 to Table-25 shows, processing times fell as the level of overlap of membership functions was increased. This speedup is readily explainable: as the overlap of MFs was increased, the number of rules evolved by the system fell, and since each rule is the result of one modal evolutionary run of 15 generations, system speed is proportionate to the number of rules evolved during classification. The longest learning time for the 7000-value Cancer set took around five and half minutes in these experiments. However, once learned, the time taken to apply the rules to the data is less than one second.

다. Experiment 2: Investigating The Effects of Clusterers

The objective of this second set of experiments was to determine the impact of using different clusterers in terms of the three performance measures of intelligibility, speed and accuracy. Two extremes of clusterer were employed: the basic method and the substantially more advanced K-Means approach. For these tests, the system used traditional fuzzy logic and non-overlapping membership functions.

Tables-26 to Table-29 shows the results obtained from 50 runs of each system set-up for each data set. As can be seen from the average and best accuracy percentages, the basic clustering does result in slightly reduced performance of classification. The performance loss is perhaps surprisingly low, though, when it is recalled how simple the basic clustering method is, compared to the K-Means approach. Different rules and different numbers of rules were evolved when using each type of clusterer, as shown by the other results in Table-26 to Table-29. There does not seem to be any clear correlation between intelligibility or processing speed and the type of clusterer used.

	Iris data				**Cancer data**			
System:	**Av. accuracy**	**Best accuracy**	**Av. time**	**Av. # of rules**	**Av. accuracy**	**Best accuracy**	**Av. time**	**Av. # of rules**
FL, non-ol MFs ***basic cluster***	97.84% **92.4%**	**93.3%**	14.9 secs	1.38	97.3% **93.62%**	**95.43%**	419 secs	11.3
FL, non-ol MFs ***k-means***	96.67% **95.79%**	**97.3%**	25.6 secs	2.94	98.6% **94.07%**	**96.0%**	317 secs	9.88

표 29 : Mean and best accuracy rates, processing times and intelligibility of solutions when using different clusterers. (Accuracy values in normal intensity indicate results for the training set, bold values show results for the test data set.)

라. Experiment 3: Investigating THE EFFECTS OF NOISE ON THE SYSTEM

The objective of this final set of experiments was to evaluate the change of performance of the system as levels of noise in the data sets was increased. Noise was cumulatively added to both data sets (and both the training and test files of each) in steps of 2%. This was achieved by scaling one randomly chosen value in every fifty by a random value. The experiments investigate levels of noise up to 10% (i.e. one in ten values is wrong). Observed levels of noise in the data sets for which this system is designed are around 1%. These experiments were performed with the system using traditional fuzzy logic and non-overlapping membership functions.

	Iris data				**Cancer data**			
Noise level:	**Av. accuracy**	**Best accuracy**	**Av. time**	**Av. # of rules**	**Av. accuracy**	**Best accuracy**	**Av. time**	**Av. # of rules**
0%	96.67% **95.79%**	**97.3%**	25.6 secs	2.94	98.6% **94.07%**	**96.0%**	317 secs	9.88
2%	94.67% **97.33%**	**97.33%**	23.0 secs	1.0	98.51% **93.33%**	**95.71%**	347 secs	9.66
4%	92.29% **87.47%**	**94.67%**	18.5 secs	2.10	97.56% **93.87%**	**95.43%**	380 secs	8.46
6%	74.19% **84.11%**	**94.67%**	19.3 secs	1.90	96.98% **89.54%**	**92.29%**	454 secs	9.82
8%	73.95% **84.83%**	**94.67%**	19.3 secs	1.82	95.67% **86.16%**	**90.57%**	423 secs	8.86
10%	74.93% **87.55%**	**96.0%**	19.5 secs	1.96	95.41% **78.02%**	**84.86%**	523 secs	13.6

표 37 : Mean and best accuracy rates , processing times and intelligibility of solutions for different noise levels. (Accuracy values in normal intensity indicate results for the training set, bold values show results for the test data set.)

Table-30 to Table-37 shows the results obtained from 50 runs of the system for both data sets, at six different levels of noise. Generally, the results show a gradual decrease in accuracy for both data sets. At ten percent noise the accuracy for the Iris data does increase, but this is likely to be chance and the fact that the number of values in the set is insufficient for a 2% noise differential to affect a significant number of data items. However, the accuracy fall off for the larger, Wisconsin Breast Cancer data set is particularly revealing.

Figure-13 shows the rate at which accuracy falls for classification of items in the training and test data sets as noise levels increase. Note the way accuracy falls linearly for the training data, but appears to fall proportionate to the square of the percentage of noise in the test data. This large decrease in performance is likely to be caused by the noise reducing the homogeneity of the training and test sets, so rules evolved for the training set work less and less well for the test set.

Upon consideration, such reduced effectiveness of rules may be manifested in two ways. Firstly rules become misled by the noise (perhaps because of overfitting by too many excessively specific rules) and thus do not generalise well to the test data set. Secondly, the noise disrupts the clusterers, so that the clustering for training and test sets becomes increasingly different. The resulting LOW, MEDIUM and HIGH fuzzy sets for the test and training data become increasing disparate, reducing the effectiveness of the rules further.

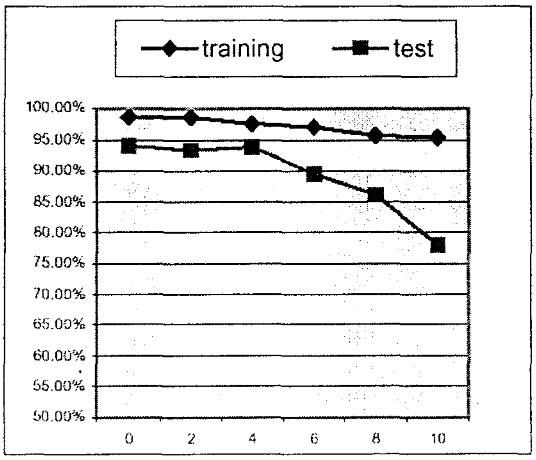


그림 13: Average accuracy for increasing levels of noise in training and test cancer data.

These effects are not so obvious for the Iris data where only a linear decrease in accuracy is evident, perhaps because of the small number of data points in this set or because few rules were used, helping rules to generalise without being misled by noise. Table 8 also shows the number of rules and processing times for different levels of noise. From these results, there does not appear to be any clear correlation between levels of noise in the data and intelligibility or speed.

마. Experiment 4: Investigating The Effects of Committee-Decisions for Insurance Data

As should be apparent, the task of detecting genuine patterns of fraud using the data provided was not trivial. Indeed, although the data was now in a fit state to be used by a classifier, there still remained the problem of the unknown data set. Lloyds TSB suggested that up to 5% of the items in this set might be suspicious, but which claims and exactly how many was unknown. To tackle this problem, three sets of experiments were performed with the committee decision system. The first experiment assessed the ability of the system to find rules indicative of suspicious items, without those patterns describing any unknown items. The second experiment assessed how well the system could find suspicious rules that also detected up to 5% of the unknown items. The third experiment assessed the ability of the system to find rules that detected suspicious items and up to 10% of the unknown items. (Note that although the system does report which claims in the unknown data set were found to be suspicious, these results cannot be provided here.)

Each experiment used four setups of the system:

1. standard fuzzy logic with non-overlapping membership functions
2. standard fuzzy logic with overlapping membership functions
3. membership-preserving fuzzy logic with overlapping membership functions
4. membership-preserving fuzzy logic with smooth membership functions

All four committee members were trained on one file and tested on the other, then trained on the second and tested on the first. This resulted in 24 different rule sets being generated for this problem, each with different levels of intelligibility and accuracy.

Estimate of fraud in 'unknown'		[A] Fuzzy Logic with non-overlappingMFs			[B] Fuzzy Logic with overlapping MFs			[C] MP-Fuzzy Logic & overlapping MFs			[D] MP-Fuzzy Logic with smooth MFs		
	files:	rules	train	test	rules	train	test	rules	train	test	rules	train	test
No more than 0%	1, 2	3	6, 0	5, 0	failed	0, 0	0, 0	failed	0, 0	0, 0	failed	0, 0	0, 0
	2, 1	2	6, 0	4, 0	failed	0, 0	0, 0	failed	0, 0	0, 0	failed	0, 0	0, 0
No more than 5%	1, 2	5	28, 177	23, 219	4	14, 464	44, 9347	2	3, 418	4, 358	1	30, 236	20, 168
	2, 1	3	29, 318	27, 312	4	16, 304	16, 387	3	3, 165	1, 174	1	24, 340	31, 278
No more than 10%	1, 2	4	35, 940	26, 399	5	12, 853	9, 725	1	4, 740	6, 759	1	30, 344	26, 420
	2, 1	4	32, 889	28, 931	5	21, 628	19, 622	2	11, 558	6, 583	1	24, 335	29, 258

그림 14 : Intelligibility (number of rules) and accuracy (number of correct classifications of suspicious items) of rule sets for test and training data. Accuracy rates are listed as n, m where n = number out of 49 correctly classified in class 1, m = number classified out of 10,000 in class 2. Results are given for training on file1, testing on file2 and training on file2, testing on file1.

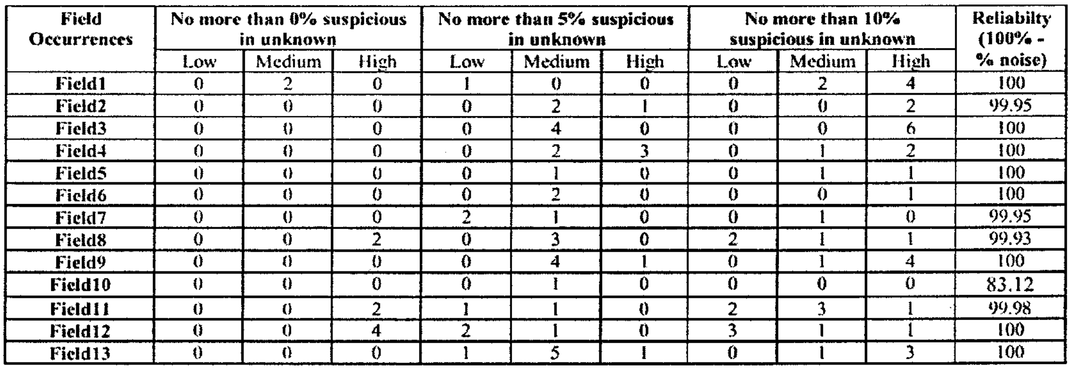
Estimate of fraud in 'unknown'	Committee decision for accuracy	Committee decision for intelligibility	Committee decision for weighted intelligibility (1) and accuracy (0.3)
No more than 0%	[A] 2nd rule set	[A] 2nd rule set	[A] 2nd rule set
No more than 5%	[A] 2nd rule set	[D] 2nd rule set	[D] 2nd rule set
No more than 10%	[A] 1st rule set	[D] 1st rule set	[D] 1st rule set

그림 15 : Best results as reported by committee decision maker.

Figure-14 and Figure-15 present the results of the experiments. It should be apparent in Table 1 that no committee member managed to find useful rules that detect 0% suspicious claims in the unknown set indeed most failed to generate any valid rules at all. When up to 5% or 10% suspicious claims are assumed to exist in the unknown data set, accuracy rates increase dramatically. As the tables explain, committee members [A] and [D] provide the most accurate and intelligible classifications for all experiments with this data. The best accuracy overall is achieved by [A], finding 61 out of 98, or 62% of the suspicious claims, whilst suggesting that 1339 out of 20000, or 6.7% of the unknown claims are also suspicious. But the most accurate and intelligible rule sets are generated by [D], with most rule sets containing just a single rule. Overall, the best rule set as reported by the committee decision maker is:

(IS_LOW Field8 OR Field3)

which can be translated as: If either the value for field8 is low or the value for field3 is high, then in 57% of observed cases the claim will be suspicious. This rule suggests that 3.8% of the unknown claims are suspicious.



Field Occurrences	No more than 0% suspicious in unknown			No more than 5% suspicious in unknown			No more than 10% suspicious in unknown			Reliabilty (100% - % noise)
	Low	Medium	High	Low	Medium	High	Low	Medium	High	
Field1	0	2	0	1	0	0	0	2	4	100
Field2	0	0	0	0	2	1	0	0	2	99.95
Field3	0	0	0	0	4	0	0	0	6	100
Field4	0	0	0	0	2	3	0	1	2	100
Field5	0	0	0	0	1	0	0	1	1	100
Field6	0	0	0	0	2	0	0	0	1	100
Field7	0	0	0	2	1	0	0	1	0	99.95
Field8	0	0	2	0	3	0	2	1	1	99.93
Field9	0	0	0	0	4	1	0	1	4	100
Field10	0	0	0	0	1	0	0	0	0	83.12
Field11	0	0	2	1	1	0	2	3	1	99.98
Field12	0	0	4	2	1	0	3	1	1	100
Field13	0	0	0	1	5	1	0	1	3	100

그림 16 :Frequencies of fields in all rule sets and reliability of fields (based on noise caused by filling missing values). Note that NOT IS_X field is expanded to IS_Y Field or IS_Z Field and IS_LOW IS_LOW is translated to IS_HIGH for mp-fuzzy logic.

Further analysis can be performed by examining the occurrences of fields in the evolved rules, see Figure-16. In general, the tally of field occurrences in the rules as shown above indicates that suspicious claims seem to be more likely when:

Fields 1, 5, 7, 9 and 13 are medium or high

Fields 2,3,4 and 6 are high

Field 8 is low or high

Fields 11 and 12 are low or medium

Interestingly, the only field with significant levels of noise Field10 is hardly used for classification in the rules. The table also shows that Field 3 seems to provide the single best indication of suspiciousness. Indeed, even used on its own, the rule:

IS_LOW IS_LOW Field3

which in mp-fuzy logic should be translated as:

ISVERYHIGH Field3

is capable of detecting 54 out of 98 suspicious claims.

바. Experiment 5: Investigating The Effects of Committee-Decisions for Credit Card Data

Three sets of experiments were performed with the committee decision system and the four different setups of fuzzy rule evolver were run for each experiment:

1. standard fuzzy logic with non-overlapping membership functions
2. standard fuzzy logic with overlapping membership functions
3. membership-preserving fuzzy logic with overlapping membership functions
4. membership-preserving fuzzy logic with smooth membership functions

All four committee members were trained on one selected training set and test set. This resulted in different rule sets being generated for this problem, each with different levels of intelligibility and accuracy.

Table-38 presents the results of the experiments. The accuracy of the system is described by a True Positive (TP) prediction rate and a False Negative (FN) error rate. The TP is the rate that the predicted output is "suspicious" class when the desired output is "suspicious" class. The FN is the probability of which the predicted output is "suspicious" when the desired output is "non-suspicious" class. The desired system will have a high TP and a low FN.

As Table-38 explains, committee member [B] provides the most accurate and intelligible classifications for all experiments with this data. The best accuracy overall is achieved by [B], detecting 100% of the suspicious claims for both on the training and the test set, whiles showing that 5.79% of false negative error, which is relatively low. In addition, the most accurate and intelligible rule sets that are generated by [B] contain just three rules. Overall, the best rule set as reported by the committee decision maker is for experiment 2:

(IS_LOW field57 OR field50)
IS_MEDIUM field56
(field56 OR field56)

and for the experiment 3:

(Filed49 OR Field56)
(IS_LOW Field26 OR field15)
IS_MEDIUM field56

These best rule sets are clearly dominated by the field 56. This implies that this field seems to be the single best indicator of suspicious case. In summary, the prediction results of these best rule sets are satisfying in terms of the accuracy and intelligibility.

Exp	[A] Fuzzy Logic with non-overlapping MFs					[B] Fuzzy Logic with overlapping MFs					[C] MP-Fuzzy Logic with overlapping MFs					[D] MP-Fuzzy Logic with smooth MFs				
	R	Training		Test		R	Training		Test		R	Training		Test		R	Training		Test	
		TP %	FN %	TP %	FN %		TP %	FN %	TP %	FN %		TP %	FN %	TP %	FN %		TP %	FN %	TP %	FN %
1	3	6.09	3.81	10.4	3.35	2	100	0	100	83.1	16	10.9	5.79	100	100	5	48.6	5.79	42.5	10.3
2	2	44.1	5.79	47.8	9.45	3	100	1.67	99.7	6.38	3	1.37	5.64	99.7	100	10	41.6	5.79	47.6	12.5
3	3	46.8	5.18	46.9	6.09	3	100	5.78	100	5.79	4	1.67	5.64	86.9	100	16	42.7	5.94	42.9	6.40

표 38: Intelligibility (number of rules) and accuracy (number of correct classifications of suspicious items) of rule sets for test and training data. **R** shows the number of rules in the generated rule set and **TP** and **FN** is represented in %.

Another interesting observation is that the results of experiments rapidly change depending on the specific experiment setup. While [B] setup always generated the good rule sets, [C] setup provided almost meaningless rule sets, which showed nearly random prediction results. The setup [D] showed the consistent results, which the differences of TP and FN for both the training and the test sets are within 6%, but the best result is not satisfying. These results show again the large variance of committee member performance and illustrate the validity of the committee-decision maker approach for this problem.

In addition, from [A] and [B]s results, it could be implied that the data set used in the experiment 1 seems to have somewhat different characters from other two data sets. The quite large difference, about 40% for TP in [A] and 80% for FN in [B] represent that the importance of data sampling during the fuzzy rule evolution stage.

제5절 Conclusion

This research has investigated the use of a genetic programming system to evolve fuzzy rules for the purpose of detecting suspicious data amongst normal data. The system contains many novel elements, including a crossover operator designed to minimize disruption, binary genotypes, and a new method for interpreting fuzzy rules designed to preserve all fuzzy set membership values. Consultation with our collaborating company, Lloyds/TSB resulted in a set of evaluation criteria for the system: intelligibility, speed, handling noise and accuracy.

With these aspects in mind, three sets of experiments were performed on the system, using two standard data sets to permit comparison with the literature. The first test investigated the effect of membership functions on the system. By increasing the overlap between fuzzy membership functions and by preserving the information held in the membership values, the results showed that the number of rules needed to classify data could be reduced. This reduction often led to a decrease in accuracy of classification, but this was offset by the dramatically increased intelligibility of output, faster processing time, and better feature selection. The second test investigated the effects of using different clusterers in the system. It was found that a basic clusterer slightly reduced the accuracy of the system, compared to the more complex K-Means approach. The choice of clusterer did not seem to have any consistent effect on intelligibility of output or processing speed. The final test investigated the ability of the system to cope with increasing levels of noise in the data. As one would expect, accuracy of classification was detrimentally affected as noise increased. Interestingly, the intelligibility and processing speed showed no clear trend for increasing levels of noise.

Together, these experiments show:

- many factors affect accuracy of classification

- intelligibility and processing speed only seem to be affected by the type and use of membership functions - noise and the choice of clusterer seems unimportant.

- noisy data causes at best a linear drop in accuracy, and at worst, a fall proportionate to the square of input noise.

As the second stage, this research has described the use of genetic programming to evolve fuzzy rules within a parallel committee decision system. Attention was paid to data preprocessing, describing some of the typical problems associated with real-world data in order to show just how hard this kind of classification becomes. Nevertheless, despite having only 49 suspicious items in the first class to train the system, and an unknown number of suspicious items in the 10000-item second class, performance of the system was good. Given the quality and quantity of the data, accuracy rates of over 60% must surely be regarded as impressive. Indeed it seems very likely that better accuracy would only result in overfitting the meagre training data. In addition, intelligibility rates were excellent with many rule sets comprising a single, understandable rule.

This work shows the benefit of committee decision making. Each of the four different committee members (different setups of the evolutionary fuzzy system) provided different rates of accuracy and intelligibility. The committee decision maker was able to analyse all results and pick the best rule set.

The evolved rules and the table of field frequencies in rules have provided important and interesting information about the nature of fraud in home insurance claims. Sadly the names of the fields and the true meanings of the rules cannot be reported in this article, but Lloyds TSB have stated that the results were sensible as confirmed by previous analysis, and support the potential for even more useful results with improved data.

Finally, a committee-decision-making evolutionary fuzzy system developed in this work was applied for domestic credit card transaction data evaluation. The results for this real-world problem confirm previous results obtained for real home insurance data. They illustrate that the use of evolution with fuzzy logic can enable both accurate and intelligible classification of difficult data. The results also show the importance of committee-decision making to help ensure that good results will always be generated.

제4장 Intrusion Detection System 개발

제1절 Early Work on Intrusion Detection Systems

1. Anderson's Computer Security Monitoring and Surveillance

An intrusion detection system (IDS) is an automated system for the detection of computer system intrusions using audit trails provided by operating systems or network monitoring tools. The main goal of an IDS is to detect unauthorised use, misuse and abuse of computer systems by both system insiders and external intruders. The research on IDSs started from Anderson's work (Anderson, 1980) which aimed to improve the auditing facilities and the surveillance abilities of computing systems. In his work, he defined and classified threats as follows (Lunt, 1988), (Lunt, 1993):

- *External penetrators* (who are not authorized to use the computer)
- *Internal penetrators* (who are authorized to use the computer but not the data, program, or resource accessed), including
  - *Masqueraders* (who operate under another users id and password)
  - *Clandestine users* (who evade auditing and access controls)
- *Misfeasors* (who authorized to use the computer and resources accessed but misuse their privileges)

Anderson proposed a new idea to detect each of these intruder types using audit trails. Firstly, he claimed that abnormal frequencies of failed login attempts might imply the presence of external penetrators. Secondly, abnormal frequencies of failed access attempts to files, programs, and other resources would indicate the presence of internal penetrators. Thirdly, if currently observed users behaviour is significantly deviated from the users' profiled normal behaviour, it could be interpreted as the penetration of masqueraders. In addition, clandestine users can breach the auditing process and the access control of an operating system. Clandestine users typically use a system privilege or operate at a level below which auditing is being performed. When clandestine users use a system privilege, it would be detected by investigating all use of functions which cripple auditing, change the user identity numbers of audited users, or change other auditing parameters. Alternatively, clandestine users operating at a lower level than an auditing level could be detected by auditing service or kernel calls. Finally, the detection of misfeasors requires monitoring particularly the abnormal accesses and usage patterns of legitimate users who have access to vulnerable resources.

2. Denning's Generic Intrusion Detection Model

Dorothy Denning is the pioneer of intrusion detection research and she propounded the first generic intrusion detection model in 1987(Denning, 1987), which is independent of any particular system, application environment, and system vulnerability or intrusion type. Since her proposal of generic intrusion detection model, various types of intrusion detection systems(IDS) has been developed. Most of these systems follow the basic notion of her generic IDS, which is a real-time expert system whose knowledge is derived from statistical inference based on the audit trails of users or system resources.. This model was employed for building a number of intrusion detection systems (Ilgun et al., 95), (Jackson et al., 94), (Lunt, 88), (Lipins and Vaccaro, 89), (Lunt et al., 92), (Mykerjee et al., 94). Denning inspired many researchers to build automated intrusion detection systems by providing the first general framework based on Anderson's rudimentary notion (Anderson, 1980). These systems proved the validity of the generic model.

The fundamental structure of this model stems from a rule-based expert system. The main units of a rule-based expert system are a knowledge base, which is a collection of knowledge represented in the form of rules, a working memory, which is a global database of facts used by the rules and an inference engine, which makes inferences by selecting rules that match the available facts. In Denning intrusion detection model, a working memory stores profile facts describing the normal behaviour of subjects with respect to objects and provide the signatures of abnormal behaviours. A statistical metric and model are used to present profiles. A subject can be an individual system user, a group of system users or a system itself, while objects can be files, programs, messages, records, terminals etc. When a subject acts upon a specific object, it generates an event, which alters the statistical metric state of both subject and object. A knowledge base contains activity rules to be fired for updating profiles, detecting abnormal behaviour, and producing reports. An inference engine makes its inference by triggering rules matching profile facts.

제2절 Taxonomy of Intrusion Detection Systems

Early intrusion detection systems operated at the *host level*, whereas contemporary systems tend to be *network-based* (Mykerjee et al., 94).

Host-based IDSs monitor a single host machine using the audit trails of a host operating system. The host-based IDS can exist in two different forms: *an intrusion detection daemon* or a *separate intrusion detection system*. The separate dedicated IDS is known to be advantageous in terms of both performance and security (Lunt, 93). A separate IDS avoids the performance degradation of a monitored system caused by the adoption of an intrusion detection daemon. Furthermore, it rules out the subversion of an intrusion detection daemon by the security compromise of a monitored system. Host-based IDSs can be referred to as stand-alone intrusion detection systems because their monitoring scope is restricted to only a single host in the form of a single process or a single system.

On the other hand, *network-based IDSs* (Mykerjee et al., 94), (Garvey and Lunt, 91), (Habra et al., 92), (Javitz and Valdez, 91), (Ko, 96), (Kumar, 95), (Paller, 98) monitor any number of hosts on a network by scrutinizing the audit trails of multiple hosts. Even though host-based IDSs have shown encouraging results (Anderson, 93), it raises the problem of how to detect intrusions attempted across the network rather than an attempt to access only a single host. In order to detect this kind of intrusions, it is necessary for an IDS to monitor multiple events generated on several hosts to integrate sufficient evidence. Furthermore, network-based IDSs monitor network traffic. Clearly a large proportion of computer system intrusions are achieved via intra- or inter-network. The use of network traffic information for security auditing renders IDSs more effective. This problem motivates the evolution from host-based IDSs to network-based IDSs. Network-based IDSs are also referred to as distributed intrusion detection systems, because it copes with multiple hosts in a distributed environment.

Network-based IDSs can be implemented by two approaches: *monolithic* and *co-operative*. The monolithic approach is to deploy a central server to monitor multiple hosts. The rapid growth of the Internet requires security officers to monitor not only local hosts but potentially also remote hosts to which they connect. It makes it impossible to restrict the security domain to only a few hosts on a local network. Even though network-based IDSs with a central server have shown promising results in small-scale networks (Mykerjee et al., 94), (Mounji et al., 95)(Mykerjee et al., 94), it is difficult or impossible to support a large-scale network. Due to the number of hosts, which may be several thousand or more, a huge amount of auditing data needs to be transferred from these hosts to the central server. This causes a severe degradation of the network performance as the number of hosts grows. The co-operative approach has been suggested to resolve this problem (Staniford-Chen et al., 96), (Porras and Neumann, 98), (Balasubramaniyan *et al.*, 98), (White et al., 96), (Vigna and Kemmerer, 98). This approach attempts to distribute a number of responsibilities of a single central server to a number of co-operative host-based IDSs. However, their ideas are at the initial stage and their validity is being currently tested.

Host-based IDSs and Network-Based IDSs mainly employ two techniques: *anomaly detection* and *misuse detection*. The anomaly detection approach establishes the profiles of normal activities of users, systems or system resources, network traffic and services using the audit trails generated by a host operating system or a network-scanning program (Mykerjee et al., 94), (Liepins and Vaccaro, 89), (Teng et al., 90), (Lane and Brodley, 97c), (Deber et al., 92), (Obaidat and Macchiarolo, 93), (Heady et al., 83). (Crosbie and Spafford, 95b), (Forrest *et al.*, 96). This approach detects intrusions by identifying significant deviations from the normal behaviour patterns of profiles. The assumption made in the anomaly detection approach is that if an intruder exploits the vulnerability of a system, it will show anomalous patterns in the activities of users, systems, system resources or network traffic and services.

The misuse detection approach defines suspicious misuse signatures based on known system vulnerabilities and a security policy (Ilgun et al., 95), (Porras, 92), (Ko, 96), (Habra et al., 92), (Garvey and Lunt, 91), (Kumar, 95), (Me, b). This approach probes whether these misuse signatures present or not in the auditing trails. Many host-based IDSs and network-based IDSs adopt both anomaly detection and misuse detection concurrently. This is because each technique has different strengths and drawbacks. These two components, anomaly detectors and misuse detectors, should be reciprocal in a complete intrusion detection system.

제3절 Implementation Issues of Intrusion Detection Systems

There are a number of factors influencing the effectiveness of intrusion detection systems. These factors can be determined by considering various examples of security breaches of a real system. In this section, the various effects of these factors on the capability of IDSs are discussed.

1. Audit Level

The two main streams of IDSs, anomaly detectors and misuse detectors use various levels of audit data such as user activities, system activities, system resource activities and network traffic. An individual IDS employs one or more audit levels of data and each audit level provides a different degree of granularity and efficiency. The following are diverse audit levels, which have been employed in the existing IDSs.

○ Normal User Activities

Some intrusions can be detected from abnormal user activities. For example, anomaly detectors can detect masqueraders by investigating an unusual login time, an unusual login location or an unusual login frequence. Appropriate measures should be determined, which can characterise user behaviour. User activities can be represented by three different measures.

§ *Login and Session Activity*: Users normal activities can be characterised by profiling their login sessions. IDSs build profiles of users normal activities from their use of application programs, commands and system resources. Candidate measures used to profile user activities for a session include last login time, session elapsed time, quantity of output to terminal, CPU usage, I/O characters per session, frequencies of failed login and command issue rate (Denning, 87), (Lunt, 93), (Lunt, 88), (Halme and Bauer, 95), (Jackson et al., 94), (Liepins and Vaccaro, 89), (Mykerjee et al., 94). These measures can be examined according to a single user or a group of users. IDSs can analyse not only the change of each measure but also the variation of overall session activities by examining correlation among multiple measures.

§ *Causal Relationship among Command Sequences*: A users normal behaviour can be identified by command sequences which users type. The assumption of employing this measure is that a user behaves in a similar way to achieve a specific goal, leading to discernible patterns of action. Thus, the causal relationships among command sequences differ on a per-user basis because an individual user has a specific goal under a specific circumstance. A specific causal relationship among command sequences of a particular user can be defined by the number of matching adjacent tokens without separation by interleaving tokens (Lane and Brodley, 97a), (Lane and Brodley, 97b), (Lane and Brodley, 97c).

§ *Temporal Patterns of Command Sequences*: Another type of measure, which describes normal user activities, is the temporal patterns of command sequences which a user types. The assumption of adopting this measure is that the sequences of commands of a single user, which are snapshots of temporal processes, follow a discernible pattern. IDSs profile normal temporal patterns from the observed behaviour of individual users and user groups (Deber et al., 92), (Teng et al., 90).

§ *Key Strokes*: A single user can be distinguished by the unique typing characteristic of each authorised user. IDSs usually profiles the time intervals, an analogue of the pressure of successive keystrokes when a user types (Obaidat and Macchiarolo, 93), (Lunt, 88). However, no operating system audits these measures while other measures are easily audited by operating systems. Thus, an extra effort is required to implement a particular auditing program.

○ Normal Command or Application Program Activities

Another audit level consists of the activities of commands or executed application programs. The commands and executed application programs are audited at the system level rather than on a per-user basis. This is an effective method if detecting viruses, Trojan horses, worms, trapdoors, logic bombs and other software intrusions. The normal activities of commands and application programs can be profiled by the following measures.

Login and Session Activity: Commands and application programs can show normal activities for a session (Denning, 87), (Anderson, 93). For example, the normal activities of commands and application programs can be represented by program CPU usage, I/O characters, number of attempts to execute an authorised program, the number of abnormal program termination for a session and the execution frequencies during a short time period. However, commands and application programs can be aliased, making it difficult to examine real activities of commands and applications.

Temporal Patterns of System Call Sequences: The other measures to describe the normal activities of commands and application programs are system call sequences (Lunt, 93). When a command and an application program are executed, we can observe that they generate sequences of system calls showing unique temporal patterns according to their activities. The argument for adopting system call profiling is that many intruders operate at a lower level that bypasses the auditing and access controls, and so the lowest level auditing is desirable. However, the number of generated system calls is enormous and this requires expensive computation.

○ Normal Activities of Privileged Programs

The main problem of adopting above two audit levels, which are normal user activities and normal activities of commands and application programs, is the enormous size of audit data. Thus, many audit data reduction tools have been reported and these tools are discussed in more details in section 2.3.4 Audit Data Reduction. Apart from these tools, a different method for reduction of a huge amount of audit data is to audit only privileged programs (Forrest *et al.*, 96), (Hofmeyr et al.), (Ko et al., 94), (Lee et al., 97). These programs running with high privileges allow them to bypass the kernel protection mechanism. These vulnerabilities in privileged programs such as rdist, sendmail, and finger have been major security holes, which intruders can easily abuse. The activities of privileged programs can be profiled in the same way as profiling commands and application programs. In other words, they can be profiled in terms of login and session activities or temporal patterns of system call sequences. However, this can degrade the detection accuracy of IDSs. Even though the privileged programs are considered as the most dangerous vulnerability, there are still many other unrelated intrusions.

○ Normal System Resource Activities

System resources are audited for profiling the normal activities according to individual users or user groups. System resource activities can also be audited on a system-wide basis (Halme and Bauer, 95), (Denning, 87), (Lunt et al., 92), (Mykerjee et al., 94). For example, storage media, file accesses, CPU usage, I/O usage, etc can be monitored in terms of usage frequencies, resource exhaustion or normal usage time on a sessions-by-session basis. This is mainly useful for detection of collaborating intrusions, and understanding minor changes in overall system activity.

○ Normal Activities of Network Traffic and Services

Many sophisticated intrusions such as sweeps, co-ordinated attacks and Internet worms are achieved via intra- or inter-network and these intrusions are detected by monitoring audit data at the network level. Generally, the network level of auditing is performed to detect these intrusions by identifying anomalies in network traffic and services, and some other security policies. The following are possible auditing measures at the network level.

Normal Activities of Network Traffic and Services: IDSs profile the normal activities of network traffic by observing the number of packets for a specific short period, expected data paths, typical network services or network service security levels. These measures can be profiled according to single users, single hosts or the overall network system.

As previously, there are various audit levels to be considered when building IDSs. Apart from these measures, we need to audit external facts to support audited measures. For example, facts about charges in user status, new users, meaning of user groups, terminated users, users on holiday, changed job assignments, user locations, etc. and profiles of expected or socially acceptable behaviour (Lunt, 93).

There is a trade-off between intrusion detection accuracy and system performance when determining the appropriate audit level. The more audit levels an IDS employs, the more valuable information can be collected. However, it is more computationally expensive. The appropriate selection of audit level is achieved by considering the following:

- *Target Intrusions:* The selection of audit level depends on the intrusions which a specific IDS intends to detect. An effective IDS certainly needs to detect a range of intrusions. For example, an IDS, which is supposed to detect Internet worms, needs network level audit data.
- *Efficiency of Auditing:* The generated audit data size varies according to each audit level. If the most significant issue is the system performance, an audit level, which generates a relatively small size of audit data, should be selected.
- *System Environment:* Different audit procedures are required depending on a particular system. The use of an existing audit procedure is easier from a practical point of view and can save time. Therefore, the audit level, which a currently available audit procedure provides, can be a more attractive option.

These considerations should be collectively considered depending on the requirements of a particular situation.

2. Batch System VS Real-Time System

Porras (Porras, 92) briefly compares two different modes of IDS. A batch mode IDS analyses audit trails some time after they have been gathered for a certain period. Generally, the amount of audit data is large and analysis is computationally expensive. A batch mode IDS is one option to avoid performance degradation due to expensive computation. This can be achieved by examining audit data during low CPU usage periods on a separate machine. If a target system does not require very strict security, a batch-mode IDS can be a very useful option. This type of IDS performs its analysis periodically, such as every week or every month.

However, some systems demand a very strict security status. For those systems, the occurrence of intrusions should be informed immediately. In contrast to a batch mode IDS, a real-time IDS examines each single audit record immediately after it is generated by the operating system. While this system is able to guarantee a more secure system by prompt detection, it raises a system performance issue. A real-time IDS can exist in two different forms: an IDS daemon and a separate IDS. In the case of an IDS daemon, a target system must be equipped with a high-speed processor, a large memory and a large storage capacity. Similarly, a separate IDS requires a high speed and large bandwidth network to transfer an enormous quantity of audit data quickly from a target system to a separate IDS.

In addition, Vaccaro(Liepins and Vaccaro, 89) pinpointed the data buffering problem of auditing programs. Vaccaro noted up to 9-minute delay between an audited transaction and generation of an audit record on disk when a target machine is a VAX running VMS 4.5. Equally, a SPARC station running SunOS 4.1 showed a similar problem. A real-time IDS can be successfully implemented after addressing this delay.

Finally, a true implementation of a real-time IDS should consider an automated reaction to the detected intrusions. Conventional real-time IDSs provide a prompt warning to a security officer. However, a human security officer may not be available 24 hours a day and 365 days a year. A intruder who subverts a target system out of the security officers working hours should be subject to an immediate reaction rather than waiting for a security officers decision. An automatic reaction might be determined according to the extent of damage that an intrusion causes. For example, if a target system is seriously sensitive in terms of security, a real-time IDS can immediately turn off the target system when it detects any intrusion. However, reactions must be carefully designed not to interfere unnecessarily with users to perform their important work.

3. Anomaly Detection Systems vs Misuse Detection Systems

Host-based IDSs are able to trace both unknown and known system breaches by the analysis of auditing trails. They mainly employ two techniques, anomaly detection and misuse detection.

○ Anomaly Detection

The assumption with the anomaly detection approach is that if an intruder exploits the vulnerability of a system, it will result in anomalous patterns in the activities of users, systems, or system resources. The anomaly detection approach builds the profiles of normal activities of users, systems, or system resources using the audit trails generated by a host operating system (Denning, 87). This approach detects intrusions by identifying significant deviations from the normal behaviour patterns of profiles.

The strength of the anomaly detection approach is that a prior knowledge of the security flaws of the target systems is not required. Thus, it is able to detect not only known intrusions but also unknown intrusions. The nature of intruders is such that innovative new approaches will be developed, as existing techniques become known to system designers. Therefore, the reliability of detection of unknown intrusions is regarded as being very significant. Furthermore, it requires only very few system dependent rules, so it is very portable. These strengths of the anomaly detection approach can make anomaly detection IDSs efficient in the detection of the following intrusions. (Denning, 87), (Porras, 92).

- *Masquerading*: a masqurader often logs into a system by using an authorized users account and password. She or he might log into a system in an unusual time, from an unusual place, and unusually frequently in a short time period. Furthermore, her or his behaviour is clearly different from the authorized user whose account and password is being abused. She or he often spends more time browsing through directories, executing system status commands or causing access denials and systems errors than authorized users.

- *Attempted Break-ins*: if an intruder attempts to break into a target system, she or he would cause password failures with the abnormally high rate.

- *Misfeasance*: when legitimate users abuse their privileges, they might break a security policy, attempting to access unauthorised resources or execute programs that are not normally executed before.

- *Illegal Dissemination of Information*: sensitive documentation can be leaked outside by legitimate users. The illegal data dissemination can be detected by observing data transferring at unusual time or through unusual system ports.

- *Inference by Legitimate Users*: legitimate users attempt to aggregate and make an inference of valuable information from a database, which is not authorized. The higher number of data retrievals and queries would distinguish them.

- *Trojan Horses and Viruses*: the abnormal CPU usage time or the abnormal number of I/O characters, which are read and written, detects intrusions of Trojan horses or viruses. This can be a valuable supplement to more traditional detection techniques.

- *Denial of Service*: intruders monopolise resources show abnormally high activities with respect to the denied resource while the other users show low activities.

There is no doubt as to whether the anomaly detection approach is efficient in detecting the above intrusions. The characteristic of these intrusions is that the signatures of system compromise are not specific. This is because these intrusions are achieved by the abuse of legitimate users or masqueraders without breaking a security policy. Therefore, these intrusions can be approximately detected by anomalous activities. However, it has several limitations, which originate from its assumption, as follows (Porras, 92).

- *False Negative Error*: the anomaly detection misses intrusions, which do not show anomalous behaviour. Not all intrusions are guaranteed to provide abnormal activity and this makes an intrusion detection system falsely report absence of intrusions.

- *False Positive Error*: the anomaly detection might report legitimate users as intruders if they show anomalous behaviours, but which is not intrusive. Depending on a users personality or a special occasion, the activity of a legitimate user can be shown as being anomalous. In this case, the anomaly detection approach falsely reports intrusion.

- *Gradual Misbehaviour*: If misfeasors are aware that their activities are monitored by the anomaly detection system, they can deceive the anomaly detector by change their behaviours gradually. The anomaly detection system learns these changed behaviours and will judge intrusive behaviours as normal behaviours.

- *Sporadic User Environments:* the feasibility of anomaly detection approach depends on the system environment. The users in some environments such as a banking system generally show regular behaviour and access a target system enough times to build a sensible behaviour profile. However, the users in university networks do not show regular behaviours or access to a target system enough times. In this case, it is difficult to establish a meaningful normal behaviour profile.

- *Expensive Computation*: anomaly detection is performed based on profiles of normal behaviour. In order to maintain more sensible profiles, the dynamic update of profiles according to the change of system environments and user behaviour is necessary. This requirement brings about expensive computation.

In order to overcome these problems, the intrusion detection system should employ a misuse detection system in parallel, which will be introduced in the next subsection. These two components should be reciprocal in a complete intrusion detection system. Hybrid systems that adopt both anomaly detector and misuse detector have been already developed (Lunt et al., 92), (Mykerjee et al., 94).

○ Misuse Detection

Misuse detection approach defines suspicious misuse signatures based on known system vulnerabilities and a security policy (Denning, 87). This approach scrutrizes whether these misuse signatures are present or not in the audit trail files. The idea of the misuse detection approach is that the known intrusions can be detected if the signatures implying their presence are defined and those signatures are present in audit data. Generally, security officers who are acquainted with system susceptibilities and a security policy designate misuse signatures. Misuse signatures can be defined in a form of rules that describe a single attributable event or a sequence of events representing an intrusion scenario.

The advantage of this approach is that it shows few failures to detect previously notified intrusions. However, this approach has a serious limitation. Misuse detection cannot detect unknown intrusions because its detection is based on known intrusion scenarios or known system vulnerabilities. Intruders often develop new paths to break into a system once when their methods have been revealed. This motivates the concurrent use of an intrusion detection system alongside an anomaly detection system.

Conventional misuse detection systems define misuse signatures in a form of rules. There are some problems in developing an efficient rule-based misuse detection system (Porras, 92), (Ilgun et al., 95), (Ilgun, 92):

- *Inflexibility of Misuse Signature Rules*: conventional misuse signature rules are directly described in terms of events recorded on audit trails. Misuse signature rules are triggered when rules and events match one-by-one. However, this one-to-one representation makes misuse signature rules inflexible. If slightly modified intrusions generating a different sequence of events occur, rules and events no longer match. Independent misuse signature rules on audit trails are suggested to solve this problem.

- *Difficulty of Creating and Updating Rules*: misuse detect system programmers have to create and update rules by interviewing system administrators and security officers and identifying corresponding events. The process of extracting experts knowledge and transforming it into rules is a difficult and error-prone task.

In order to overcome these problems, new approaches to the development of misuse detection systems have been reported. They include a model-based approach, state-transition analysis, a pattern matching approach and so on (Ilgun et al., 95), (Porras, 92), (Ko, 96), (Habra et al., 92), (Garvey and Lunt, 91), (Kumar, 95), (Me, b). These are discussed in detail in section 3.2 Misuse Detection Systems.

4. Audit Data Reduction

The success of any anomaly and misuse detection system is determined to a great extent by the inputted audit data. A competent IDS must base its output on data that provides sufficient evidence of intrusions. For this reason, most auditing components of operating systems aim to record a large number of events at a fine level of detail. As a result, the sheer volume of auditing data generated by the operating system is enormous. This massive amount of data causes a shortage of storage space and protracted processing. Furthermore, operating systems all-inclusive approach to data recording results in a quantity of data, which may be considered excessive. Therefore, a tool to reduce audit data is desirable in order to increase efficiency. The audit data reduction aims to reduce audit data without losing indicative information. There are various approaches to reduction of audit data as follows:

- *Compression Techniques*

As an obvious first step, compression techniques have been employed to reduce storage space. However, this technique brings about extra post-processing, i.e. decompression.

- *Judicious Selection of Audit Level*

As described before, there are several different audit levels. Among them, the monitoring of privileged programs can reduce audit data massively. However, this approach should be adopted after investigating whether this audit data is enough to detect intrusions.

- *Judicious Selection of Audit Features*

Similarly, audit data can be reduced by the meticulous choice of audit features, focussing on those, which are indicative of intrusions. One suggested solution to achieve this goal is standard audit formats. The suggested standard formats intend to prevent operating systems from determining audit features in an ad-hoc manner (Bishop *et al.*, 96), (Wee, 95), (Hoagland et al., 95), (Bishop, 95). These standard formats employ the most general features, which are necessary to detect the prevalent known intrusions. As another solution, AI feature selection algorithms are applied (Frank, 94), (Doak, 94). These algorithms aim to identify and delete poor and redundant features, which can lead IDSs to show improvement in their performance. They search through the space of all possible combinations of features and a specific state in this space, which is a particular subset of full features, is selected. The selected state should be the most indicative of intrusions by containing few irrelevant or noisy features. There is a myriad of algorithms to do this task and details are discussed in (Frank, 94) and (Doak, 94). Do (Frank, 94) particularly tested several techniques on simulated attacks to determine whether feature selection algorithms can improve IDSs performance.

- *Reduction of Audit Data Dimension*

While feature selection techniques attempt to identify and remove poor features, data dimension reduction techniques aim to project a high-dimension space of audit data onto a lower dimensional space. The data dimension techniques view all possible combinations of features as a high-dimension space. There are various techniques employing different mechanisms The most prevailing technique is principal component analysis (Ross and Hallem., 95), (Lam et al., 96). This technique finds a linear transformation of the co-ordinate system, which ignores dimensions having only small variances in the data. As another solution, clustering algorithms have been deployed (Frank, 94). Grouping all possible features into several clusters can reduce audit data. IDSs only need to analyse a small number of clusters rather than analysing all audit data. There are several techniques, which are statistical algorithms or artificial intelligent algorithms. More details about clustering algorithms can be referred to in (Decker and Focardi, 95).

Even though various audit data reduction techniques have been suggested, more sophisticated techniques such as feature selection and data dimension reduction are not popularly used. This is because many IDSs require real-time processing and data preprocessing work, which requires extra processing time. Furthermore, more indicative features or clusters should be dynamically ascertained according to the change of normal activities of audit targets. This also requires extra processing time. However, when these techniques are employed, the required extra processing time to apply them and the reduced time remaining to analyse the resulting audit data must be considered.

5. Audit Data Format

The efficiency of audit data depends on content and format (Bishop, 95). The first requirement of efficient audit data, content, is whether they provide meritorious information to detect intrusions. The second element of efficient audit data, format, is whether audit data format is extensive enough to handle a heterogeneous set of target systems or not. Conventional audit data formats are disparate according to their target systems. When an intrusion detection system monitors multiple hosts, it should convert all different formats of audit trail into a standard format. This is vital when IDSs attempt to detect intrusions that subvert systems across the network rather than an attempt to access only on a single host. IDSs demand a standard format of audit trails, which satisfies the needs of heterogeneity, transpobility across various network protocols, and flexibility to meet variegated needs in various environments. There are a number of endeavors to designate a standard audit format to satisfy this requirement (Bishop *et al.*, 96), (Wee, 95), (Hoagland et al., 95), (Bishop, 95).

6. Test Methodology

Even though a lot of effort has been made to develop IDSs, it is difficult to rely entirely on them. Trust in IDSs might be achieved through stringent testing. However, no standard test methodology, which can guarantee the performance of IDSs, exists. Security officers need to know that IDSs are invaluable to detect intrusions and the range of intrusions that IDSs are able to manage. Rigorous tests of IDSs enable security officers to set up synthetic anti-intrusion systems comprised of not only intrusion detection systems but also intrusion prevention systems.

However, it is very difficult to test IDSs thoroughly (Pukeza et al., 96). First of all, it is impossible to collect a comprehensive set of all intrusions. This is because the number of known intrusion techniques is fairly large and new intrusion techniques are ceaselessly introduced. Secondly, the performances of IDSs are likely affected by various conditions of target systems and users. In other words, even though IDSs detect an intrusion in computer systems, the IDSs possibly fail to detect the same intrusion when the computing activity is high. For these reasons, it is difficult to find an IDS which was tested in a stringent way in development. The IDSs which have been developed were tested in a surprisingly primitive way, such as whether the anomaly detector can distinguish different users or not, or whether the misuse detector can detect a few simulated intrusions.

Puketza, Zhang, Chung and Mukherjee (Pukeza et al., 96) suggested a new standard methodology for testing IDSs. The basic notion of this methodology is to simulate a broad range of prevalent intrusions. In their report, they identified the performance goals that must be achieved in testing as follows:

- Broad Detection Range: all the intrusions that are simulated must embody a broad range of intrusions.
- Economy in Resource Usage: IDSs should use system resources such as CPU and disk space in an economic way.
- Resilience to Stress: IDSs should perform their work correctly under stressful conditions of systems, such as a very high level of computing activity.

The first test goal is obviously necessary so as to establish an absolutely secure system. However, as described above, it is impossible to simulate a large number of candidate intrusions. The main issue in meeting this objective is which intrusions are selected for tests. The details about this topic can be referred from (Pukeza et al., 96).

The second goal must be achieved for practical reasons. Even if IDSs show good performance at detecting intrusions, if they require a large amount of memory, CPU and disk space causing overall system performance degradation, the adoption of these IDSs will be impractical. However, this problem is not so serious due to the falling cost of computer hardware resources. Therefore, according to system environments, the acceptable consumption of system hardware resources by IDSs must be investigated.

The third goal is also necessary because stressful conditions can occur unexpectedly often. In addition, some intruders intentionally take up so much of system resources in order to put computer systems in stressful conditions. A stressful condition can lead to false positive detection by misuse detection components, but competent anomaly detection components might detect the presence of intruders. Therefore, in the tests, IDSs must satisfy this goal.

However, according to computer system environment and security status, the significance of these three goals varies. The detection of a broad range of intrusions is ideally recommended, but this is not necessary for all cases. For example, if security officers already comprehend which types of intrusions can be surely prevented by existing intrusion prevention components, the IDS will only be required to detect other intrusions which threaten the system. Similarly, the satisfaction of the second goal is also flexible. If security issues are paramount, the deployment of IDSs will not be restricted due to considerable expenditure. Likewise, some systems can prevent intruders from creating stressful conditions by the restriction of users system usage based on their privileges. Therefore, the tests of IDSs should be designed by considering the current environment of target systems.

제4절 Literature Review of Host-Based Intrusion Detection Systems

A number of IDSs have been developed since Denning suggested her generic intrusion detection model. In this chapter, the host-based IDSs which have been developed are introduced and evaluated in terms of current research issues.

1. Anomaly Detection Systems

Various approaches have been attempted to implement anomaly detection IDSs. The basic idea of anomaly detection comprehends normal behaviours of users, systems and system resources. There have been a number of approaches are reported. They are statistical approaches and artificial intelligent approaches such as machine learning, neural networks, evolutionary computing and computer immunology. In this section, these approaches are described and compared with respect to current research issues.

가. Current Research Issues of Anomaly Detection Model

The development of anomaly detection systems can be divided into main two tasks: how to build more sensible profiles and how to detect excessive deviations from normal activity profiles. There are several research issues to perform these tasks.

○ Building a Normal Behaviour Profile

A variety of efforts to build more indicative profiles have been made. In other words, the successful establishment of normal activity profiles requires judicious consideration of various factors. As such efforts, cautious selection of audit level, audit data format, and audit data reduction techniques are already discussed. Apart from these, there are some other issues to be considered when anomaly detectors build profiles.

First, *how can anomaly detectors collect all possible normal history data?* This question rises from the initial assumption of anomaly detectors. Anomaly detectors define anomaly when they show significant deviation from normal activities. This is a close assumption. That is to say, if anything is not judged as normal activities, it will be deemed as anomaly. Therefore, only complete collection of all possible normal activities can support this assumption. There are two methods to cope with this question. (Hofmeyr et al.) Firstly, we can gather all possible normal cases by artificially generating synthetic normal cases. After analysing all possible normal cases in a synthetic way, all of them are simulated and collected. This is very useful when replicating results or comparing results in different parameter settings. However, this cannot test the false positive case of anomaly detector. In other words, even though we can test whether anomaly detector can detect anomaly or not, but we cannot test whether detected anomaly is real intrusion or not. Secondly, we can collect normal cases by tracing the real normal activities in a live user environment. This case is more realistic in terms of dealing with real environment. However, this raises different questions. How can we guarantee that collected data do not include abnormal case? How long we have to collect real data to cover all possible normal cases? The answers about these questions should be considered to build good profiles

Second, *how can anomaly detectors define normal activities?* After collecting normal activities, we need to define these normal activities in a general term or rule in order to test whether a new event is anomalous or not. This is achieved by identifying general features to describe collected normal activities. The size of collected auditing data is generally very large and its structure is not simple. In addition, when normal activities are collected in a live user environment, some auditing targets show abnormal activity, but which is not intrusion. We can regard this irregular activity as noise of normal activity history data and this data should be disregarded as normal activity. Thus, we need a sophisticated method to unravel general features of normal activities. Various models have been attempted to perform this task and the details are described in the next section. Among them, which model will characterize normal activities more precisely by filtering noise data out and identifying general feature of normal activities?

Third, *how can anomaly detectors represent identified features of normal activities?* After identifying features of normal activities, we face another question: how can anomaly detectors represent recognised features? This is a question of efficiency. Which representation of normal activities will be more efficient to detect anomaly? Even though anomaly detectors are competent to identify the feature of normal activities, they can make poor decisions unless they employ efficient representation schemes to describe identified features.

Fourth, *how can anomaly detectors update profiles?* Computer system environments are not static. They are being constantly changed by users, vendors and system administrators. The normal activities of auditing targets are also being continuously changed according to this environment. Therefore, profiles should be dynamically adjusted. There are several questions to update profiles.

Firstly, how often should anomaly detectors update profiles? The update of profiles in a real-time when a new event is generated will clearly show the most precise information. However, frequent update of profiles will require extra computation of IDSs and can cause delayed processing time. Therefore, the number of profile update must be determined by considering this problem. Secondly, when profiles are updated, how much of information should remain and how much of information should be pruned? For example, when a new event is generated and this event can imply new activity, which has not been recorded in current profiles. In this case, is this event considered as anomaly or normal activity, but just which has not been collected before. If this event is deemed as normal activity and the new information implies from this event is contradictory to previous information, how should anomaly detector update profiles? Does it delete previous information completely and add new information? Or does it add the information compromising these two information?

○ Detection of Excessive Deviations from Normal Behaviours

The second task of developing anomaly detection systems is how to detect excessive deviations from normal behaviour. The completion of this task also raises several research issues as follows.

First, *how can anomaly detectors determine anomaly from observed deviations from normal activities?* If anomaly is detected by investigating significant deviations from normal activities, how much of deviations will be significant enough to determine anomaly? Most initial approaches adopt intuitively defined thresholds (Denning, 87).

Second, *how can anomaly detectors cope with correlationships among a number of features?* There can be a number of features to describe normal activities. Anomalous patterns of overall system and user activities can be detected by not only anomaly of individual features but also correlationships among them. The way of considering a number of features collectively depends on the intrusions which current IDSs target to detect.

Third, *how can anomaly detectors detect intrusions from detected anomaly?* This question is not a sheer question of implementing anomaly detector. Rather, this is a question of building IDSs. This problem stemming from the initial assumption is a false positive error of anomaly detectors, which anomaly generated by legitimate users is regarded as intrusion. In order to solve this problem, we need a supplementary component.

There are several issues to be considered to build anomaly detectors like above. These issues are taken into account in various approaches and each approach shows strengths and weaknesses depending on specific issues. In the next sections, these various approaches are discussed with respect to these issues.

나. Statistical Approach

Denning (Denning, 87) suggested a number of disparate statistical approaches that are applicable to implement intrusion detection systems.

- Operational Model: this is the most rudimentary statistical model. The anomaly is detected merely by comparing a new observation to fixed thresholds. The thresholds are intuitively defined based on history data. When a number of features are considered to determine the anomaly of overall system, this model is only able to probe the anomaly of individual feature. This model is simple and easy to implement, but the interrelationships among features are ignored. Thus, it is difficult to detect sophisticated intrusions.

- Mean and Standard Deviation Model: this model defines the thresholds by estimating a confidence interval. A confidence interval is a set of statistical estimates such as mean and standard deviation, which can be accepted as normal. If a new observation outside a confidence interval of significant level, it can be regarded as anomaly. A significance level represents the probability of wrongly determining anomaly. This is more quantitative when thresholds are defined. However, a specific data distribution is assumed when confidence intervals are defined. This trait causes a practical problem to implement.

- Multivariate Model: many intrusions can be detected by interrogating the interrelationship among several features concurrently. While operational model and mean and standard deviation models consider only specific feature, multivariate model probes a number of features collectively. It aims to elucidate the hidden meaning of correlation among features.

These statistical approaches are prevalently adopted to establish IDSs. Apart from these, variegated statistical approaches such as Markov-Chain model, time series model and Bayesian network model can be considered (Denning, 87), (Kumar, 95). In this section, a developed IDS harnessing mean and standard deviation model, HAYSTACK is described.

○ HAYSTACK

HAYSTACK was designed to help Air Force Security Officers to detect the misuses and the anomaly of Unisys 1100/2200(Mykerjee et al., 94). This system provides the short summaries of user behaviours, anomalous events, and security accidents. HAYSTACK focused on detecting several types of intrusions such as attempted break-ins, masquerade attacks, penetration of the security system, leakage of information, denial of service, and malicious use. HAYSTACK is a batch system that reports the daily-based audit trail analyses to security officers. Everyday HAYSTACK transfers audit trails from a mainframe to a separate PC specialised in intrusion detection processing. Intrusion Detector residing on a separate PC provides daily reports within a few hours. In addition, HAYSTACK employs an independent audit trail pre-processor. The audit trail pre-processor running on the mainframe extricates the necessary information and reformats it.

HAYSTACK employs three main operations. First of all, HAYSTACK reports the notable single events that modify the security status of an overall system. When it observes that any single event modifies the Security State of a target system such as access controls, user-ids, and group-ids, HAYSTACK reports it with explanatory messages. The second operation is that HAYSTACK monitors particularly some system resources that are regarded as being fragile by a security officer. The system reports to a security officer when these system resources are accessed. However, these reports might be generated too often.

As the third way, HAYSTACK uses a statistical analysis for two different subgoals. One is that a statistical component analysis users collective session behaviours. This operation reports a weighted multinomial suspicion quotient which indicates how much user groups collective session behaviours conform to the expected behaviours of known intrusions as compared to all other sessions. HAYSTACK profiles two dozen measures of the user's session such as time of work, number of files created, etc. These measures represent the activities of user groups for a specific session. The values of these measures are compared to the predefined thresholds. The thresholds represent 90 percent of historical measurements and threshold ranges are defined under the assumption that the values of measures follow Gaussian distribution. The measures whose values are outside the ranges of thresholds are weighted by the severity of individual known intrusions and these weighted scores are reported as suspicion quotient. Therefore, the suspicion quotient denotes the extent of anomaly of each session for single user group in terms of specific intrusions. The second statistical analysis is used to spot the alterations of users behaviours when their behaviours in recent sessions are compared to previous sessions.

2. Statistical Profile and Rule-Based Expert System

The initial idea of Dennings generic intrusion detection model is focused on building a statistical profile of normal activities and implies the occurrence of intrusions by a rule-based expert system. Even though the main logic to detect anomaly is based on statistical analysis, a rule-based expert system is required in order to perform tedious and complicated work. They are necessary to perform successful statistical analysis such as creation and update of profiles, system reaction when anomaly is detected. In this section, a system employing statistical profiles and a rule-based expert system, Multics Intrusion Detection and Alerting System (MIDAS) is described.

○ Multics Intrusion Detection and Altering System (MIDAS)

The Multics Intrusion Detection and Alerting System (MIDAS) is a real-time anomaly detection system for the National Computer Security Centres networked mainframe, Dockmaster, a Honeywell DPS-8/70 Multics computer system (Mykerjee et al., 94). MIDAS is a stand-alone Symbolics LISP machine. It is based on Dennings generic Intrusion Detection model (Denning, 87), which employs statistical profiles and a rule-based expert system. The statistical profiles provide the simple statistics of individual user and system activities. These statistics can be calculated through auditing user commands. A rule-based expert system makes a forward-chaining inference with explanations. The rules of expert systems are used to detect anomaly by matching the statistics in profiles, which are regarded as normal activities, with the thresholds predefined in a knowledge base, which are regarded as expected activities. Apart from the statistical profiles and a rule-based expert system, MIDAS deploys a audit data preprocessor, which transforms original audit trails into a canonical format, a network-interface-daemon, which maintains communications between MIDAS and target mainframes, and a user interface, which provides communication between MIDAS and security officers.

The rules of MIDAS are defined to particularly detect attempted break-ins, masqueraders, penetrators, Trojan horses, viruses and misfeasors. These rules are divided according to their auditing targets.

○ Detect Immediate Intrusion

These rules aim to detect immediate attacks, which can be detected merely by observing auditable events. These rules do not require any further statistical analysis. Instead, the clear anomalies of auditable events by themselves are investigated by these rules.

○ Detect User Anomaly

These rules probe the anomalous activities of users and hosts. All the activities of users and hosts for a session are statistically summarised in the user profiles. The rules examine at the end of session whether the observed behaviours of latest session are significantly different from the past behaviours or not. When a security officer does not judge the revealed anomaly as intrusion, the profiles are updated by considering them.

○ Detect System-Wide State Anomaly

MIDAS also investigates the global system security status. With the profiles of individual users and hosts, MIDAS maintains the system-wide profile describing the global normal behaviour of target systems. These rules can detect intrusions by observing the anomaly of global system behaviours.

3. Machine Learning Approach

The cornerstone of machine learning model is to automatically extract the normal activities of features, which are considered as being critical to detect anomaly, from garnered audit data. When history data of features are provided, a machine learning model attempts to identify the hidden rules to define the normal behaviours of features and these identified rules are used to determine whether a newly observed event is anomalous or not. There are several approaches employing different definition of critical features. In addition, different approaches adopt different forms of profiles. Some describes profiles in the form of rules and the others do in the form of numerical functions. In this section, three different approaches, which employ different features and different forms of profiles, are discussed.

○ Wisdom and Sense

Wisdom and Sense (W&S) is a rule-based anomaly detection system developed at the Los Alamos National Laboratory (Mykerjee et al., 94), (Liepins and Vaccaro, 89). It runs in real-time on a stand-alone UNIX workstation and probes the audit trails transferred from VAX/VMS hosts. W&S aims to detect malicious or erroneous behaviours by users, Trojan horses and viruses. W&S builds the rules, which represent the normal activities of users and hosts, by themselves. The rules can be defined based on the routine activities recorded on the audit trails. The anomaly is detected by identifying the deviated patterns from the normal activities described by the rules.

The first main operation of W&S is to generate the rules automatically from the garnered auditing data. The audit data is comprised of myriad of audit records and a single audit record shows a single event. A single audit record consists of a number of fields, which are the subjects and objects of event. The rules articulate the routine values of "test fields" when the values of other fields are given. The test fields are the auditing targets, which are deemed as being critical enough to distinguish anomaly from normal activities. For any test fields, W&S investigates all the combinations of observed values of the other fields.

Therefore, the left-hand sides of rules are comprised of all the observed values of other fields and the right-hand sides of rules are the values of test fields when the left-hand side conditions are provided.

The elementary units of the rules, fields, are designated by converting the original audit data into a number of categories. This is because most of fields are categorical data and the numerical values of continuous variables are also arbitrary (such as port number). Therefore, if the values of fields are categorical, W&S directly uses the fields, but if the values of fields are continuous, the system converts them into several acceptable categories using clustering algorithms. Whenever a new audit record is generated, a new rule is generated or previously generated rules are updated. While audit records are being generated, W&S continues building the forest of rule trees. The generated rules are pruned in order to maintain the reasonable size to handle and present the latest normal activities.

The second main operation of W&S is to detect anomaly using simple statistics. Whenever a new record arrives, W&S trawls through the rules and matches applicable rules. The system measures the figure of merit (FOM) for each event. The FOM of each event is the normalised sum of the grades of standard error of selected rule. The standard error is calculated by assigning passing strengths and failing strengths. The passing strength underpins the collected evidences when a new event satisfies the formerly generated rule and the failing strength shows vice versa. W&S makes a decision that a new event is anomalous when its FOM is over the predefined threshold.

○ Time-Based Inductive Machine (TIM)

Time-Based Inductive Machine (TIM) (Teng et al., 90) is a rule-based anomaly detection system. Time-based inductive learning is harnessed to generate the rules, which profile the normal temporal patterns from the observed behaviour of individual users and user groups. The rules are generated under the assumption that the sequences of events, which are snapshots of temporal processes recorded in profiles, follow a discernible pattern. This assumption allows TIM to consider the interrelationships and orders of events. The prototype of TIM is implemented on a VAX3500 with 32 megabytes memory. TIM detects anomaly within seconds of its occurrence.

The generated rules imply the forthcoming pattern, which indicates the future event, when the sequence of patterns, which represent previous events, are provided. The inductive learning analyses all the possible sequences of events and converts them into rules. The rules are pruned depending on their prediction accuracy before they are stored in the rule-base. A high number of correct inferences for each rule produces a high confidence value for the rule. For example,

E1 -> E2 -> E3 => (E4 = 95%, E5 = 5%)

, where E1 through E5 are events. This rule implies that the observed sequence of events, E1->E2->E3 would result in the occurrence of E4 with the probability 95% and the occurrence of E5 with the probability 5%. During the learning phase, the rules are adjusted dynamically and finally only the rules with high confidence values remain in the system. When a new event is generated, the inductively generated rules are retrieved. The rule, whose left-hand side matches the observed sequence, is selected and the implied event according to this rule is compared to the last event in the observed sequence. If this event is deviated far from the implied event of the rule, TIM alerts the anomaly to a security officer. However, if a new sequence of events does not match any rule in the rule-base, this is reported to a security officer. A security officer determines whether this sequence is anomaly pattern or normal pattern, but which is not observed before. In the case of the latter, a new sequence of events is converted into a new rule.

○ Machine Learning Experiments

Another approach of machine learning for a intrusion detection system (Lane and Brodley, 97a), (Lane and Brodley, 97b), (Lane and Brodley, 97c) has attempted by the COAST(Computer Operations, Audit and Security Technology)team at Purdue University. Like the other anomaly detection systems, this approach also learns user behaviour from the profiled history and detects abnormal behaviour by comparing it to the currently observed user behaviour. While many systems mainly profile all the possible behaviour of users, groups and target systems, this approach only profiles the input streams of an individual user. The underlying assumption of the adopted learning algorithm is that a user responds commands in a similar way under similar situations and this leads to repeated sequences of actions. The similarity between the sequence of actions (UNIX commands) of the current user's input stream and each fixed-length input sequence profiled in a user's command history is measured by using the newly suggested function. When the similarity calculated by the similarity function falls outside the thresholds, the input stream of a current user is regarded as anomaly. Lane and Brodley have attempted to verify the validity of their assumption and a newly suggested similarity function through a number of various experiments.

In the initial experiment, this approach profiles the sequence of commands only within the context of a single command interpreter. The input stream is segmented into a token stream with the negligence of file names and the preservation of command names and argument switches. This token stream is divided into predefined fixed length sequences and they are maintained in a dictionary. The novel numerical similarity measure is derived by the intuition which is that as the number of matching adjacent tokens without separation by interleaving tokens increases, the causal relationship between two sequences of token streams becomes stronger. Lane and Brodley stipulated this by the following example. If sequence Seq1 has k tokens in common with each of Seq2 and Seq3, but the common tokens are adjacent in Seq1 and Seq2 and the common tokens are separated in Seq1 and Seq3, then the similarity between Seq1 and Seq2 is higher than the similarity between Seq1 and Seq3.

The first experiment was performed in a batch style. The collected history data of user input streams are divided into two groups: training data and test data. A training data set is used to build a dictionary of history input streams. A dictionary consists of an instance database, a similarity measure and a set of system parameters. The anomaly detection of test data was performed by measuring the similarity between the historical sequences in a dictionary and each sequence in a test data set is measured. This provides a stream of similarity measures. The original stream of similarity measures in the experiment is noisy due to the normal deviations in a user's actions. Lane and Brodley filter the noise out by a smoothing filter. The smoothed stream shows a steady trend indicating normality and it allows a security officer to intuitively set an appropriate threshold. As a user types a new sequence, the similarity to the sequences in a dictionary is estimated and abnormality is detected when this similarity is below the threshold.

Lane and Brodley performed a number of experiments to prove their assumption of the adopted learning algorithm and the effects of system parameter selection such as the sequence length and the dictionary size. First of all, they built UNIX shell command histories of 4 users, which were amassed for 4 months. All these four users who provided their UNIX input streams are the postgraduate students at Department of Computer Science in Purdue University. Therefore, this experiment can be said to perform on the extremely experienced computer users rather than novice users. For each user, the number of collected tokens varies from minimum 7,769 to maximum 23,293. In addition, the collected data sets are divided into two groups for training and a test. The proportion of data for training and a test is 2/3 and 1/3 respectively. In the first experiment, Lane and Brodley showed the employed novel machine learning algorithm could distinguish 4 different users when test data was given. When a 12-length sequence of one user from the test data set was presented, the learning algorithm can distinguish the users as the detection accuracy from minimum 69.85% to maximum 99.19%. In the second experiment, the impact of sequence length on detection rate was investigated. As the result of the experiment performed on the first user, the detection accuracy increases as the sequence length increases and the false positive rate (the incorrect classification of original user as other users) increases as well. Furthermore, the optimal library size for the individual user was scrutinised. The experiment results verified the optimal size of library is user-specific. According to the characteristics of each user's behaviour, one user showed fairly regular behaviour pattern and the other one did not.

Although this approach shows a promising result in the initial experiment, a number of problems still remain. First, the system requires the optimal selection of several system parameters to reach good performance. These parameters are a sequence length, a window length for the training set, determining a threshold and a library size. Lane and Broadly have reported from their initial experiments that the system performance is very sensitive to these parameters. However, they have not provided how to optimize these parameters to bring in good detection. Second, the definition of the suggested novel similarity measure function and the selection of threshold are merely achieved by intuition. More sound mathematical proofs to advocate these decisions are necessary. Third, more investigation to choose an appropriate filter to smooth the original similarity measure stream is necessary. Finally, adaptive learning to take account of only recent sequences is also demanded.

4. Neural Network Approach

Neural networks are renowned for their competent learning capability. Attempts to use neural networks for anomaly detection have been performed. The learning mechanism of neural networks is used to build profiles of normal behaviours of users, systems or systems resources. In this section, two different models employing different audit level are presented.

○ Time Series Analysis

One of those attempts (Deber et al., 92) employs a recurrent neural network to learn a user's regular activity. Debar, Becker and Siboni represent a user behaviour pattern in terms of a time series pattern. The neural network learns a sampling window of n consecutive commands before moving this window along in time to obtain a new sample of n consecutive commands. After learning the time series pattern, which the consecutive commands generate, the neural network predicts the next command. Each input neuron of a neural network receives the user's command and after learning, the output neuron predicts the user's next command. The network consists of 60 input neurones that denote 60 commands by one user and 60 output neurones that show continuous values indicating the probability of occurrence of each command. As a result of training on 6550 commands, the network showed up to 92.25% prediction accuracy for the first command. This experiment was performed to test the validity of a neural network as an IDS by investigating whether the prediction of a specific user's next command contributes to distinguish intruders from authorised users. As a front module to determine whether a new audit record is suspicious or not, the deployment of a neural network was suggested based on the test results which a neural network reasonably distinguishes a specific user's pattern.

○ Keystroke Dynamics

Another attempt (Obaidat and Macchiarolo, 93) claimed to adopt neural networks as a learning module for an intrusion detection system. This attempt aims to identify a user's individual keystrokes. The input patterns are comprised of the time intervals between successive keystrokes. The underlying assumption is each person has his/her own typing technique. In order to audit the time intervals between successive keystrokes, Obaidat and Macchiarolo used assembly language based on the IBM 386 PC, which can provide the time duration between keystrokes by using software interrupts. 15 sequential values of time intervals between keystrokes are set up as one vector to represent each user's typing technique, and 40 such vectors are collected for one user from 6 users. The network used in this experiment is a backpropagation network, which consists of 15 input nodes, 6 output nodes and 3 layers. Each vector of a user's keystrokes is encoded by 15 input nodes and each output node represents a specific user. Obaidat and Macchiarolo have verified the possibility of using neural networks by showing 97.8% classification accuracy in the test.

The strengths of the neural network approach are as follows: first, they don't assume any unrealistic statistical assumptions, second, they automatically take into account the correlation of different input audit measures, third, they cope very well with noisy data. However, they also have several drawbacks. First, the difficulty of optimizing the parameters, the topologies of networks. The prediction capability of neural networks are very susceptible to the number of input nodes, output nodes, layers and the choice of learning rates. A large amount of effort spent in trial and error is required to reach satisfactory results. Second the weakness of explanation about their decisions. The learning mechanisms of neural networks have not been completely clarified. They are unable to provide sufficient grounds for their decisions. Third, the tests have been performed only on a small amount of training and test data. The performance of neural networks is particularly sensitive to the characteristics of their training and test data. Therefore more thorough investigation is required.

5. Evolutionary Approach

As the first attempt to use an evolutionary algorithm for detecting network anomaly, Heady et al. proposed a classifier system monitoring network traffic (Heady et al., 83). This system consists of three main steps: network traffic monitoring and pre-processing, extracting the rules describing normal activities of network traffic and flagging the deviations from those activities and evolving the extracted rule while they detect various intrusions.

The first step, network traffic monitoring is performed by a network traffic monitor. It extracts timestamp, packet size, source-destination address pair, and network protocol descriptor. These events are used to characteries each network connection. In order to perform the other two steps, a classifier system maintains aggregation and event detection rules, database update rules, threshold rules and prediction rules. Aggregation and event detection rules count statistics of events and flag significant events. Database update rules update the past event profiles. When the aggregation rules fire, the update rules update the statistics of events by considering new counts. Threshold rules maintain thresholds of individual events and fire if the statistics of any event, which are counted by aggregation rules, exceeds defined thresholds. Finally, prediction rules are applied to one or more threshold rules to perform multivariate analysis. If all of the threshold rules fire as abnormal, then prediction rules report the network state to be abnormal. The final step of a classifier system is evolving the prediction rules by a genetic algorithm. All the prediction rules, which were fired to report abnormal network states, are rated according to their prediction results. If their prediction is right, they will get reward, otherwise they will be penalized. A genetic algorithm uses this score as its fitness function and applies mutation and crossover operators. As the result, the weakest rules are removed and a number of new rules are generated from the remaining strongest rules using crossover and mutation.

This work shows good adaptability of evolutionary computing approach for network intrusion detection. However, it is only very initial stage of work. It missed the evaluation of learning time, which will be the crucial point of employing this approach. Also, it used only limited events and intrusions for test. It did not show how much of various intrusions could be detected using evolutionary concept. In other words, it did not show how much the evolved prediction rules from initial aggregation rules could detect unknown intrusions.

6. Computer Immunology

While most of the above intrusion detection systems have developed based on the rigorous research in the computer security community, a novel approach was proposed by the researchers in the artificial intelligence community. They have investigated how a natural immune system works well for protecting humans from dangerous foreign pathogens such as bacteria, viruses, and parasites and modelled a computational immune system to protect a computer system from computer viruses and intrusions (Somayaji et al., 97), (Kephart, 94), (Forrest et al., 97), (Forrest *et al.*, 94). This idea started from that several researchers view computer viruses as a form of artificial life (Spafford, 94).

One of them, Forrest (Forrest et al., 97) has attempted to build a computational immune model for the intrusion of a computer system. She regarded the intrusion detection problem as distinguishing "self" from "other" which is the cornerstone of a natural immune system. This view can be said to be very similar to the view of anomaly detection. She suggested two approaches to implement this basic notion. In the first one (Forrest *et al.*, 96), she profiles the normal behaviours for privileged system processes and compares them to currently observed behaviours. This idea is not much different from a typical anomaly detection idea. The only difference of her experiment is that she profiled the system calls of privileged system processes rather than user commands or system usage. She argued the profiles of system calls resulted in a massive reduction of audit data. In the second approach (Forrest et al., 97)(Forrest *et al.*, 94), she follows the protecting mechanism of a natural immune system more closely. Like the first idea, a normal behaviour of a user or a system, self is profiled. A system generates a set of detectors, which fail to match some normal behaviour. While a system is executing, if a detector is activated, it is implied there is a change on behaviour and this behaviour is deemed as an abnormal behaviour. In (Forrest *et al.*, 94), this idea was tested for detecting computer viruses.

In (Forrest *et al.*, 96), the validity of the first idea of computer immunology was assessed using several different tests. They selected *sendmail* as the privileged process to be profiled. In order to profile the system calls of *sendmail's* normal executions, they retrieved the system calls which belonged to a k+1 size window and recorded chronologically the system calls within a window. By sliding a window along the trace of system calls, the various combinations of system calls could be profiled. When the patterns of *sendmail's* normal executions were profiled, special efforts were made to maintain the generality of the profile. Therefore they artificially created 112 different types of messages which would be sent by *sendmail.* For various operating systems they reprofiled the system calls of *sendmail* process for each of 3 different window sizes. The results obtained were combined into a single built profile which was then subjected to two tests in order to confirm its validity. In the first test, each of the normal system calls of *sendmail* were compared to the normal system calls of other processes and approximately showed 5-32% mismatching. Forrest, Hofmeyr and Longstaff interpreted this result as the only information of the order of system calls can distinguish the behaviour of different processes. In the second test, Forrest, Hofmeyr and Longstaff simulated three different types of system process behaviour. First, successful intrusions using *sendmail* vulnerability, second, failed intrusion attempts using *sendmail* vulnerability and third, error conditions. Although discrepancies among the mismatching rates with *sendmail* normal system calls depended on specific type of abnormal behaviour as expected, the test results did not show this expectation. The mismatching rates of the first and the second types of process behaviour showed at average over 2%. However, this rate is similar to the mismatching rates of the third type of process behaviour, 2.3% and 2.7%. Thus, normal system call profiles did not show any clue to distinguish error conditions from intrusions.

This first computer immunology approach does not show any distinguishing feature of human immune systems such as distributed anomaly detection. Instead, it tried to show the concept, self and non-self on a single host and this is not much different from the conventional anomaly detection mechanism. The interesting point of this work is that it showed the system call sequences of privileged process could be a good indicator to describe self of a monitored host. Even though it is not the first attempt, it proves again this audit level will be quite promising in terms of building a light-weight IDS. However, it needs to be tested on more extensive intrusions in order to prove whether this audit level is good enough to replace other heavy-weight audit level such as auditing all application programs and auditing all user commands typed.

제5절 Misuse Detection Systems

Misuse detection approach defines suspicious misuse signatures based on known system vulnerabilities and a security policy. This approach scrutinises whether these misuse signatures present or not in the auditing trail files. The idea of misuse detection approach is that known intrusions can be detected if signatures implying their presence is defined and those signatures are perceived (Denning, 87). The main issue of implementing misuse detection systems is how to define security signatures in a more efficient way. In this section, several approaches to considering this issue are discussed (Porras, 92).

1. Rule-Based Expert System Approach

Conventional misuse detection systems deploy a rule-based expert approach. The initial idea of misuse detection requires the definition of misuse signature and interrogates them from observed audit data. The nature of rule-based expert system supports this idea naturally. The misuse signatures and the monitoring mechanism to observe the presence of misuse signatures can be defined in a knowledge base as a form of rules. However, the conversion of misuse signatures originally recorded in audit records into rules is not simple task. Various models to get around this task have been reported. In this section, these various models are introduced.

○ State Transition Analysis Tool (STAT)

The State Transition Analysis Tool (STAT) (Ilgun et al., 95), (Porras, 92) was propounded as an efficient tool to build a rule-based expert system for misuse detection and was implemented for UNIX in USTAT (Unix-specific STAT)(Ilgun, 92). The fundamental goal of STAT is helping knowledge engineers to build a rule-based knowledge base. Even though a number of vulnerabilities are known, it is not easy for knowledge engineers to translate them into precise rules. STAT is a graphical tool to elucidate the key actions to cause an intrusion that should be described in the rules in a knowledge base. In this approach, an intrusion is conceived as a sequence of states and their transition. This notion is deduced from the assumptions which are: first, all intrusions are completed when an intruder gains minimum prerequisite access to a target system and second, all intrusions aim to obtain some privileges which are not held by them. Therefore, the states associated with one intrusion represent the minimum signatures to cause an intrusion. The successful completion of each state at each step permits an intruder to transit to the next sate in a sequence of states. The incessant successful transition finally leads an intruder to compromised states which he/she desires to access.

In order to clarify the minimum intrusion signatures efficiently, STAT designates the above notion by using a state transition diagram. A node indicates a state and an arc illustrates its transition. The initial state, which is the first node, signifies the previous state just before the start of an intrusion and the last node specifies the compromised state after a successful intrusion. Between these two states, the minimum prerequisite intermediate states are drawn according the order of their occurrences. Knowledge engineers are required to describe the rules, which monitor only the states represented in a state transition diagram, rather than all an intruder's typed commands corresponding to all the events in an audit profile. However, this scheme only supports to specify totally ordered sequences of states. More sophisticated tools such as representing a partially ordered sequences of states are demanded. Intrusions stem from the variegated combinations of key signatures.

○ Specification-Based Monitoring

One of the problems an intrusion detection system should be tackled is how to manage a tremendous amount of audit data. A myriad of ways have been reported and a specification-based monitoring (Ko, 96), (Ko et al., 94) is one of them. The cornerstone of this approach is to define the security specifications of privileged programs, which stipulate the expected behaviours of these programs and audit only their activities by probing their violations in respect of the presented specifications. The operating systems of a computer system usually have privileged programs, which are allowed to bypass the security mechanisms of operating systems. Nevertheless operating systems delegate these privileges to programs so as for them to perform their specific jobs, insidious intruders are able to get around the security mechanisms underpinned by operating systems. The concise specification of privileged programs requires low complexity and it makes IDSs detect intrusions in real-time. The presented specifications grasp the valid operation sequences of the execution of privileged programs. If the observed operations during the execution of privileged programs are not mapped to these valid operations, a system is discerned as being intruded. Furthermore, Ko (Ko, 96), (Ko et al., 94) suggested a novel language in order to describe specifications more effortlessly. The language is devised based on simple predicate logic and regular expressions and the novel grammar of the language underpins tracing a concurrent program execution.

This approach obviously reduces a tremendous of audit data by monitoring only the executions of privileged programs. However, this idea only outlines detecting some parts of intrusions related with privilege programs. And this method cannot guarantee detect all the intrusions associated with the selected privilege programs. The reduction of audit data costs an intrusion detection system relinquishing the monitor of more variegated intrusions.

○ Advanced Security Audit Trail Analysis on UNIX (ASAX)

ASAX developed jointly by the University of Namur and Siemens-Nixdorf Software S.A. (Habra et al., 92), (Habra et al., 91) is a rule-based expert system for misuse detection which particularly outlines the universality, power and efficiency of audit trail analysis. From the point of view of universality, they developed a normalized audit file format (NADF) which can be transformed from all the different types of audit formats. This format consists of chronological sequence of audit records and each record consists of an audit data identifier, data length, and data value. Furthermore, the system maintains auxiliary files to preserve the information about the relationship between external and internal formats of audit data. This allows the system to manage the queries described in the external format. With respect of power, a language, RUSSEL, which is specialized in processing a large sequential file efficiently and have the power of rule-based language was developed for ASAX. In other words, this language adopts predefined control structure, which is appropriate for making inference from a chronological sequence of audit data. Finally, ASAX was developed considering its efficiency of audit trail analysis. As the principal requirements of achieving efficiency, ASAX focuses on the following two points: the analysis of audit trail should be completed by one pass and the repetitive steps should be reduced as little as possible. RUSSEL language was designed carefully to underpin the first point by encapsulating the necessary information about the past. The second point was achieved by efficient implementation techniques. The detailed implementation techniques can be referred from (Habra et al., 91).

2. Model-Based Approach

Garvey and Lunt (Garvey and Lunt, 91) suggested a model-based intrusion detection system which is a variant to a rule-based misuse detection system deployed at NIDES. In the case of typical rule-based misuse detection system, the definition of the rules to describe known misuse scenarios costs developers a large proportion of time and effort. The model-based IDS is proposed particularly to get around this weakness.

This approach builds a knowledge base, which consists of the specifications of various intrusion scenarios and models. These scenarios are formalised by the sequence of events possibly recorded in audit profiles rather than by the language to represent rules. The system endeavours to probe the current activities in order to pinpoint evidence supporting the intrusion scenarios. If the system can find evidence to support the occurrence of a particular intrusion scenario, one intrusion model or scenario supported by this evidence is initiated to be active. The system finally can alert the occurrence of this intrusion as more evidence enough to support the active scenario is amassed. While the system seeks more evidence, it updates the likelihood of the occurrence of a particular intrusion scenario and deems this likelihood to be over a predefined threshold. This capability to verify of the occurrence of intrusion is achieved by an evidential reasoning.

This approach has several strengths over typical rule-based misuse detection systems. Firstly, it helps developers to deal with enormous amount of audit data. The model allows the system to concentrate only on interrogating the corresponding data to the active intrusion scenarios at one moment. Secondly, the ground for a system's decision is presented in a more intuitive explanation. The intrusion is detected not by the language specifying rules but by the events that are directly observable in audit trails. Therefore a system can easily provide explanations about its decisions.

3. Pattern Matching Approach

The different approach for a misuse detection system, a pattern matching (Kumar, 95), (Kumar and Spafford, 94) was attempted by Kumar. He regards the key signatures of intrusions as specific patterns and adopts a pattern matching to identify the intrusion signatures form audit profiles. A Coloured Petri Network (CPN) which is a prevailing graphical tool in a pattern matching field is used to describe intrusion signatures. Each node in a network represents a state and an edge describes its transition. There can be one or more initial states led to a unique final state indicating a particular intrusion and a token is assigned to each initial state. When one state is in transit, the next state is determined by evaluating the expressions attached to each transition. The evaluations of expressions are performed by assigning tokens to the local variables of expressions. In other words, the successful assignment of tokens to the local variables of expressions determines the next state. This approach supports a partial order matching by allowing multiple initial states for a unique final state.

In addition, Kumar extensively surveyed notified vulnerabilities and classified them according to structural interrelationships among observable system events. Furthermore, he identified the common structural interrelationships of events comprising a specific intrusion based on the categories class and developed a particular computational model to detect intrusion for each class. This classification is undoubtedly efficient for a pattern matching scheme. This is because vulnerabilities are classified from the point of view of observable events rather than resultant vulnerabilities. Therefore, an efficient matching can be achieved by applying a corresponding computational model to an observed event. Pattern matching shows how attacks can be classified as patterns which match against occurrences in a system. There patterns can encode dependencies between system conditions (i.e. event x must happen and so must y) and also temporal conditions (i.e. event x must happen before event y while condition z is true). This is a powerful method of detecting intrusions, but it relies on the patterns being generated beforehand. If the patterns are incomplete then there may be hole in the systems defences. Again, the patterns may have to be re-generated if the systems behaviour changes due to a policy or operational change.

The pattern matching approach has several strengths. Firstly, it is portable. The pattern matching approach can handle a heterogeneous set of hosts without rewriting intrusion signatures. This is because the patterns defining intrusions are abstract enough to cover different hosts and security policies. In addition, the adoption of partial order matching allows thus scheme to deal with non-consecutive intrusions. For example, some intrusions are performed under different user accounts and different times. These intrusions can be detected by collecting a number of non-consecutive signatures. The pattern matching scheme performs its partial match when it detects the first signature and keep track of them over several login sessions. However, this advantage causes a problem. It requires considerable overhead and can result in the severe problem to operate IDSs in real-time.

4. Evolutionary Algorithm Approach

○ Genetic Algorithm Approach

The validity of Genetic Algorithms were tested by Me (Me, a), (Me, b) as a complement to a model-based approach. The genetic algorithm in his experiment was used as a tool to search efficiently for the evidences supporting the intrusion models of a model-based misuse detection system. Genetic algorithm aims to find events, which have the greatest risk to a system amongst the subsets of all the possible intrusion models. The selected subset, which is considered as the greatest risk to a system, can be interpreted as ones, which are the most likely to occur. The system needs not to be committed to match all the observed events to the corresponding events in all the intrusion models of a knowledge base. Rather the system is expected only to pinpoint the events related with the selected models by a genetic algorithm. This allows a system to save a large amount of computation to collect evidences for a model-based misuse detection system.

The genetic algorithm starts its evolution by setting up a population, which is initialized by a set of randomly selected possible solutions. Each member is evolved through crossover and mutation and evaluated its fitness to an optimal solution by a fitness function. Me sets up the population of initial solutions as a N-dimensional hypothesis vector, H, which maximizes the product W * H. Here, N is a number of possible intrusion types. The element of H, Hi, is turned on by 1 if the intrusion i is presented and turned off by 0 otherwise. W is an N-dimensional weight vector, which is proportional to the risk with each intrusion. In order to make the hypothesis realistic, a constraint is added; the number of events for a particular intrusion which is observed from a audit trail should be greater than or equal to the number of events which are required to cause that the specific intrusion. The finally gained solution through the evolution of a genetic algorithm represents the one set which collects different types of intrusion models and these models are deemed as they can subvert the system as much as the greatest risk.

It was believed that a genetic algorithm would be an advantageous component for an intrusion detection system through the above experiment. Regardless of the satisfactory test results, several questions are raised. First one is the definition of a set showing the greatest risk. This set is defined simply by considering the predefined extent to risk and the absence and the presence of each intrusion model. This definition totally disregards the correlation of different intrusion models. A more sophisticated fitness function, which should consider a number of intrusion models, collectively is ignored. Second question is the definition of constraints. The order of events of an intrusion model is disregarded when the constraint is defined. The intrusion is not taken place merely by the collection of required events. It is caused as a result of the sequential feedback from the required events.

○ Genetic Programming Approach

Another attempt to use an evolutionary algorithm has been presented by (Crosbie and Spafford, 95b). They proposed autonomous multi-agent systems for network intrusion detection (). In this proposal, each agent is required to be adaptive to dynamically changing network environments. As one approach to implementing adaptive agents, Crosbie and Spafford used a genetic programming (GP). In this work, a number of GP agents evolve by learning the normal and abnormal building blocks from prepared training scenarios, which are mixture of both intrusive and non-intrusive activities. This GP agent evolves as batch style learning and the evolved best agent is selected for a stand-alone IDS.

The agent implemented using GP consists of a set of operators (which are logical, arithmetic, and conditional) and a set of primitives. The primitives are extracted event from monitored network traffic such as total number of socket connections, average time between connections, minimum time between connections, maximum time between socket connection, destination port to which the connection is intended and source port from which the connection originated. Combining these primitives and operators in any way as a parse tree form generates a single agent. The fitness value of generated agent is evaluated at each evolution step based on its performance. To evaluate the performance, training data were prepared by simulating four different types of intrusions on network traffic data and assigning the probability of each intrusion. This predefined probability indicates the degree of an intrusion. The absolute differences between the agents reported suspicious degree and the training scenarios assigned probability determines agents fitness value.

This work emphasized the autonomous adaptability of GP and it showed good adaptability. The evolved prediction rules have only selected events from presented initial events. It showed the possibility of using GP to perform feature selection process. However, this work has a critical problem, which requires extensive and deep training scenarios. This means it has to simulate all intrusions to collect right training data. From practical point of view, it is very difficult to simulate all intrusions. Moreover, it requires assigning the probability of each intrusion and it is not easy to evaluate the degree of risk.

5. Hybrid Systems

Many host-based IDSs and network-based IDSs adopt both anomaly detection and misuse detection concurrently. This is because each technique has different strengths and drawbacks. In this section, the hybrid systems that employ both an anomaly detector and a misuse detector are introduced.

가. Statistical Profile and Rule-Based Hybrid Systems

○ Information Security Officer's Assistant (ISOA)

The Information Security Officers Assistant (ISOA) is a real-time hybrid system running on a UNIX-based workstation (Mykerjee et al., 94). This system automatically reports the alarms of suspicious security status and allows security officers to analyse the details of them interactively in a graphical and textual form. ISOA monitors the activities of individual users and hosts by using audit trails. ISOA establishes the profiles of indicators, which signifies the presence of intrusions. Those indicators are determined from both the specific signatures of intrusions, whose penetration paths are known, for misuse detection and the approximately implied flags of intrusions, whose penetration paths are unknown, for anomaly detection. Furthermore, the indicators embody hierarchical meaning to signify intrusions. Therefore, the observance of indicators at each monitoring step increases or decreases concern level of target systems. The ISOA extracts only the events, which are recorded on audit trails, related with indicators.

The statistical component and the rule-based expert system are reciprocal. They analyse audit trails concurrently in order to evaluate the concern level of target system security. According to the concern level which is constantly modified by observing the evidences of intrusions, the analyses of audit trails are structured. The global security status of individual users and hosts are appraised by this hierarchical structure. The indicators of anomalous activities are determined by significant deviations of observed events from expected measures. The thresholds, which determine the significant deviations of observed events from expected measures, are defined by a simple statistical analysis. Whenever the indicators of anomaly are flagged, it is recorded in profiles as a historical abstract of monitored behaviour. The rule-based expert system can make an inference about the meaning of indicator flags based on this historical abstract of monitored behaviour. Furthermore, the rules employed in the expert system define the relationship among individual events, which indicate more sophisticated intrusions.

The anomaly detector of ISOA probes the audit trails in two different schemes: real-time preliminary analysis and batch secondary analysis. The real-time preliminary analysis is performed when audit trails are generated by operating system. The predefined indicators stored in profiles are examined promptly after a single event is generated. When the observed events show their activities outside the ranges of indicator threshold, this prompt investigation reports timely alarms to security officer. For the time being the further scrutiny can be instantly required according to the judgement of security officer. The batch secondary analysis is invoked at the end of a user login session or when required for further analysis. The expected users and hosts activities for a session are compared against currently observed statistics of user and host session activities. These examinations inform security officers of suspicious sessions and make them to scrutinise more details.

○ Next Generation Real-Time Intrusion Detection Expert System(NIDES)

The Next Generation Real-Time Intrusion Detection Expert System (NIDES)(Lunt and Jagannathan, 88), (Javitz and Valdez, 91), (Lunt, 93), (Anderson, 93), (Jackson et al., 94) which was developed by SRI International is a hybrid system adopting a statistical profile-based anomaly detection system and an expert system-based misuse detection system. NIDES aims to be a general-purpose IDS which is independent of a heterogeneous set of hosts, application environments, audit level, system vulnerability or types of intrusions. A statistical component residing on NIDES profiles the activities of individual users, remote hosts and target systems. The profiling is performed by maintaining the statistics of various measures representing these activities. NIDES requires only a small amount of storage for the profiles by storing merely the statistics of intrusion detection measures rather than preserving all historical audit data. The measures can be dynamically adjusted according to the object to be profiled. In addition, the profiles are updated daily and thus a statistical component can adaptively learn the most recent behaviours of monitored objects. The statistical component and the expert system component are loosely coupled and thus they operate in parallel. The alerts reported independently from each component compensate for each other. The security officer can make the final decision from these two complementary reports.

As a new audit record arrives, it is converted into a vector of intrusion detection measures, which are employed in profiling. This vector is compared with the corresponding measures of the profiles in order to estimate the score. The estimated score reflects the extent of similarity to the normal pattern of the specific measure observed of a new record. If the score is large, it implies the audited behaviour is far deviated from the normal behaviour recorded in the profile. A human security officer flexibly assigns the threshold of the score providing the criterion of abnormality. For example, a security officer can determine a yellow level warning when the total score falls within 1% of past data and a red level warning when the total score falls within 0.1% of history data. A multivariate method allows the statistical component to estimate the total score showing the overall behaviour by taking into account each of the individual scores of single measures. Furthermore, the statistics of profiles are dynamically updated using the specification of a half-life. The assumption under adopting a half-life is that the activities of most recent days contribute more than the activities of more remote days. The appropriate length of profile is determined by the suitable selection of half-life.

An expert system adopted for misuse detection makes its forward-chaining inference based on the set of rules in a rule-base. The rules describing intrusion cases are defined from the knowledge of past intrusions, known system vulnerabilities, the intuition about suspicious behaviour and the installation-specific policy. The rules represent the minimum requirements of the most prevalent vulnerabilities such as logins, user privilege and file access rather than encompassing all known vulnerabilities. The auditing trails provided by the auditing subsystem of UNIX are the input data to be matched with the condition parts of the rules. However, UNIX auditing subsystem's inability to report the arguments of UNIX shell commands interferes with firing several rules describing some vulnerability. Apart from two main components: statistical anomaly detector and rule-based misuse detector, NIDES employs audit data generating components, a user interface component, and other various data communication components.

제6절 Literature Review of Network-Based Model

Network-based IDSs are developed to detect network intrusions, which attempt to subvert computer systems across the network rather than an attempt to access only a single host. Compared to single computer systems, network/distributed systems tend to be more vulnerable. This is because network typically includes more resources to be protected, network systems are usually configured for resource sharing, and a global policy applied to all the hosts, which commonly comprise a heterogeneous set, is rare. Moreover a network intruders attempt to intrude upon multiple points across the network. (Heady et al., 83), (Ko et al., 93). It is formidable to protect network systems, but unavoidable. In this chapter, the current research issues of developing network-based IDSs are discussed and the network-based IDS that have been developed are introduced.

1. Current Research Issues of Network-Based Model

The vulnerability of network systems motivates the development of network-based IDSs. In contrast to host-based IDSs, the new auditing target of network-based IDSs requires the following functions. By scrutinizing these functions, we can comprehend the current research issues of network-based IDS.

○ Robustness

Network-based IDSs should have *multiple points, which are robust enough against the attack on IDSs.* The critical weak point of IDSs is the subversion on IDSs by intruders. If intruders already know the existence of IDS and subvert IDSs, then all efforts to develop IDSs are futile. Furthermore multiple points of IDSs must have *their own unique detection procedures.* This is because intruders knowledge on a specific point should not allow them to be able to subvert other points (Balasubramaniyan *et al.*, 97), (Crosbie and Spafford, 94), (Hofmeyr et al.), (Somayaji et al., 97).

○ Fault Tolerance

Similarly, network-based IDSs must be *tolerant to any system faults.* In other words, when a single point of network-based IDS is faulty, an entire intrusion detection mechanism should not be crippled even though it may causes degradation of overall system performance (Balasubramaniyan *et al.*, 97), (Crosbie and Spafford, 94), (Hofmeyr et al.), (Somayaji et al., 97).

○ Efficiency

Network-based IDSs should be *simple and lightweight enough to impose a low overhead on the monitored host systems and network.* As a network grows, a number of resources to be handled by an IDS rapidly increases. In this case, a single IDS is expected to perform monitoring, data gathering, data manipulation and decision making. It imposes a large overhead on a system and places a particularly heavy burden on CPU and I/O. This causes severe system and network performance degradation. In addition, an IDS gathers an excessive amount of raw data from all the hosts and this takes up a great deal of disk space (Balasubramaniyan *et al.*, 97), (Crosbie and Spafford, 94), (Crosbie and Spafford, 95a).

○ Adaptability

Network anomaly detector should be dynamically adjusted in order to detect evolving network intrusions. Computer system environments are not static. Users, vendors and system administrators are constantly changing them. The normal activities of network and intrusions are also being continuously changed according to this environment (Hofmeyr et al.), (Somayaji et al., 97).

○ Scalability

It is necessary to *achieve reliable scalability to gather the high-volume audit data correctly from distributed hosts.* Multiple hosts in a network generate high-volume of audit data and this audit data is dispersed among the various systems. In the case of the monolithic model, the audit trail collection procedure is distributed and its analysis is centralized. However, it is very difficult to forward all audit data to a single IDS for analysis without losing the data. Even if it scales all audit data correctly, it may cause severe network performance degradation. Furthermore, in an inter-network environment of multiple administrative domains, network administrators in different domains may be unwilling to share all the information with others (Balasubramaniyan *et al.*, 97), (Crosbie and Spafford, 94), (White et al., 96), (Porras and Neumann, 98).

○ Global Analysis

In order to detect network intrusions, network-based IDSs should *monitor collectively multiple events generated on various hosts to integrate sufficient evidence and to identify the correlation between multiple events.* Many network intrusions often exploit the multiple points of a network. As a simple example, if an intruder attempts to guess a password of any user account, he or she usually moves to different hosts to hide his or her attempts. Thus, from a single host, they might appear to be a just normal mistake. But if they are collectively monitored from multiple points, they clearly appear to a single attack attempt (Mounji et al., 94), (Ko et al., 93).

○ Accountability

Network-based IDSs should be able to *trace the physical origin of intrusions.* Attackers typically hide their physical location, thus preserving anonymity. They usually login to a series of hosts to confuse their point of origin rather than directly subvert a target system. Besides, the hosts which intruders login to can be in different administrative domains, with administrators who may not know or trust one another in advance. All of these problems make it difficult to chase the origin of attackers and let intruders continue attacking (Staniford-Chen and Heberlein, 95), (Ko et al., 93), (White et al., 96), (White and Pooch., 96).

○ Configurability

Network-based IDSs should be able to *configure themselves to the local requirements of each host or each network component.* Individual hosts in a network environment are heterogeneous. They may have different security requirements. In addition to hosts, different network components such as routers, filters, DNS, firewalls, or various network services may have various security requirements (Balasubramaniyan *et al.*, 97), (Crosbie and Spafford, 95a), (Crosbie and Spafford, 94).

○ Extendibility

Network-based IDSs should be able to *extend the scope of IDS monitoring easily and simply regardless of operating systems of new hosts.* When a new host is added to an existing network environment and especially when this new host runs a different operating system that has a different format of audit data, it is not simple to monitor it in a consistent manner with existing IDSs. However, the network environment is very dynamic. New hosts running different operating systems are often added (Balasubramaniyan *et al.*, 97), (Crosbie and Spafford, 95a), (Crosbie and Spafford, 94), (Paxson, 98), (Porras and Neumann, 98).

2. Monolithic Approach

As discussed in Taxonomy, the monolithic approach is to deploy a central server to monitor multiple hosts. Most of network-based IDSs at the early stage are to employ the monolithic approach which monitor the small number of hosts, which is generally under hundreds In this chapter, the network-based IDSs employing monolithic approach are discussed with respect of the desired functions of network-based IDSs.

가. Infancy of Network-Based IDSs

Even though the truthful network-based IDS should perform the network level of auditing (refer to 2.4.1 audit level), the network-based IDSs at the early stage do not monitor network activities. They monitor only user activities, commands or application program activities or system resource activities. However, they are able to transfer the host audit trails from multiple monitored hosts to a central site for processing. This aims to perform global analysis. Host-based IDSs, ISOA and IDES (Mykerjee et al., 94) are upgraded are able to support multiple hosts by employing this approach. This approach provides the function that the various formats of audit data transited from different OSs are merged into a single specified canonical format. And thus it allows a central IDS to perform a global analysis.

Another host-based IDS, ASAX (Mounji et al., 95), specification-based monitoring (Ko, 96) and IDIOT (Kumar, 95) has been also upgrade by employing this approach. In addition, it allows local hosts to perform local analysis and a central server performs only global analysis. It shows more improved efficiency and scalability.

나. Monitoring Network-Traffic Data

After the addition of simple audit trail transferring module to typical host-based IDSs, researchers interest shifted to how to detect intrusions attempted across the network. It requires auditing network traffic simultaneously with auditing local hosts and users.

○ **NADIR (Network Anomaly Detection and Intrusion Reporter)**

NADIR is the first IDS employing the network traffic information. This was developed for Los Alamos National Laboratorys Integrated Computing Network (ICN) (Jackson et al., 91), (Mykerjee et al., 94). NADIR monitors the weekly network activity of individual users and whole network. Each local host sends raw network audit records to a single intrusion detection server, NADIR. The collected network activity information includes the generated authentication attempts, network classification level and the network component which user attempts to access.

○ **NSM (Network Security Monitor)**

The Network Security Monitor is developed at the University of California, Davis (Mykerjee et al., 94), (Heberlein et al., 90). NSM models the network and hosts in a hierarchical-structured Interconnected Computing Environment Model (ICEM) which has 6 layers: packet, thread, connection, host, connected network, and system layer. The lowest layer represents the network packet and the highest layer represents the state of entire network. The network traffic audited via ICEM provides input data to the expert system. This input data includes the simple network traffic summary of individual connection such as start time, number of packets that each host sends for one connection period, bytes of data which each host sends for *one connection period*. Additionally, profiles of expected traffic behaviour such as expected data paths and service profiles, knowledge about capabilities of each of the network services, level of authentication required of each of the services, level of security for each of the machine, and signatures of past attacks are provided. The expert system analyses these input data and reports the security state of a particular connection over a specific time period.

This is the first approach to use the network connection-oriented profiling. Also, its hierarchical-structured ICEM allows NSM to adjust the granularity of analysis according to the performance of machines and the amount of traffic. In other words, NSM provides the analysis of host-to-host activities at the lowest level, services activities at the next level, connection activities at the next level. A security officer can choose any level of analysis.

○ **DIDS (Distributed Intrusion Detection System)**

The Distributed Intrusion Detection System is jointly developed by UC Davis, Lawrence Livermore National Laboratory, Haystack Laboratory and the US Air Force (Mykerjee et al., 94), (Snapp et al., 91). This is the evolution of NSM. NSM cannot detect an intruder that breaches a system via a dial-up line, and thus may not generate any network activity. Similarly, it is completely vulnerable for encrypted network traffic. These deficiencies are caused from which NSM audits only network traffic. Instead DIDS audits not only network traffic but also individual host and user activities like other host-based IDSs. More significantly DIDS aims for combining distributed monitoring and data reduction with centralised data analysis. It puts the first step forward handling arbitrarily wider area from the LAN environment.

DIDS consists of four components: host monitor, LAN monitor, DIDS director and communication protocols between components. The host monitor resides on an individual host and analyses the audit records from the specific hosts operating system. This performs only simple analysis and sends notable events to the DIDS director for further analysis. The LAN monitor runs on each LAN segment and investigates all of network traffic. This is a subset of NSM. The DIDS director is composed of communication manager, an expert system and a user interface. The communication manager is responsible for the transfer of data between the director and each of the security status of each individual host and overall system based on the reports from host monitors and LAN monitors. Another new feature of DIDS is its accountability. It employs the network-user identification (NID) operation. When the first time a user appears at the monitored network environment, DIDS allocate a unique NID to a user. Even after a user rlogin to different hosts, DIDS can track him or her by identifying his or her NID.

DIDS attempted to avoid system performance degradation by delegating local analyses to local monitored hosts. This architecture showed better scalability and not to impose too many overheads on monitored hosts. It handles a fairly large number of hosts, but as the network size grows, this architecture still cannot guarantee safe scale. Even though host monitors and LAN monitors helps reduce audit data volume, which needs to be transferred to a DIDS director, they only perform simple local analysis and still needs to send a large volume of notable events to a DIDS director. In other words, most of detection decision still depends on a central decision-maker, a DIDS director and it limits the size of network that DIDS monitors.

Most of network-based IDS that are currently being used employ a DIDS style architecture and audit data. They employ a central intrusion detection server, which analyse notable events generated by monitored hosts. They are Bro (Paxson, 98), NID (Network Intrusion Detection) (NID), NFR (Network Flight Ranger) (Ranum et al., 97), Shadow (Paller, 98) and these systems are freeware. Apart from these systems, many commercial systems such as Cisco NetRanger, ISS Real Secure (Paller, 98) also employ a similar architecture with a better graphic user interface and good summarized detection reports.

Among them, Bro and NFR provide functions to customize network profiles. They filter network packets, which arrive at monitored hosts, into a series of events according to a configuration defined by a user. A users configuration describes the significant events that IDS will analyse for detecting specific intrusions. After generating events, they execute interpreters for specific scripts that describe a sites specific security policy. Both Bro and NFR provide configurability and extendibility via specific script language. However, this approach requires a user a lot of efforts to write a site-specific security policy in a provided script language. Despite of this deficiency, they provide various events for off-analysis by implementing real-time packet filtering and reassembling. Therefore, they can be significant front engines to other network-based IDSs.

3. Co-operative Approach

The general approach to co-operative mode attempts to distribute a number of responsibilities of a single central server to a number of co-operating IDSs.

These are usually independent of each other. Each IDS is responsible for monitoring only a local host and thus it is unable to detect network intrusions by itself. However, if a number of IDSs operate concurrently and co-operate with each other, they can make a coherent inference and finally make a global decision. In this section, the network-based IDSs which employ co-operative approach are introduced.

○ GrIDS (Graph-Based Intrusion Detection System)

The GrIDS project at University of California, Davis, (Staniford-Chen et al., 96) focuses on the detection of large-scale network attacks such as Internet worm, malicious sweeps, co-ordinated attacks and etc. These large-scale attacks form differentiated network activity patterns from normal network activities and GrIDS uses network activity graphs to identify them.

GrIDS views a whole large network as the aggregation of a number of sub-network. Any host belongs to a sub-network called a department and a department can has a parent department or child departments or both. A data source module executing at any host monitors host and network activity and it sends reports of detected anomalous activity to a graph engine. The reports from data sources are converted into a graph by user-defined rule sets. A graph engine is responsible for building a network activity graph of a department. Each network activity graph represents host and network activities in a department. The generated graph is evaluated whether it shows anomalous graph pattern or not. This graph representation allows GrIDS to perform global analysis for detecting large-scale network attacks. Furthermore, the network activity graphs generated by graph engines are passed up to the graph engine of a parent department. The graph engine in a parent department in turn builds graphs that have a coarser resolution. The graph at high level shows only summarized information about lower level graphs rather than showing a complicated topology of combined big network. This hierarchical approach makes GrIDS scalable.

However, it shows weaknesses at its configurability and extendibility. When the topology of current network is changed, it will cause the change of network hierarchy and the whole aggregation rule sets should be changed. But, this is not simple work. Also, when a graph engine apart from them locating at the lowest level is attacked or crashed, it causes the failure of all other IDSs locating above the originally crippled IDS. Likewise, it shows disadvantages at its robustness and fault tolerance as well.

○ EMERALD (Event Monitoring Enabling Responses to Anomalous Live Disturbances)

Based on the early efforts in developing the host-based IDS, NIDE (Next-generation Intrusion Detection Expert System), SRI International proposed a hierarchical network-based IDS, EMERALD (Porras and Neumann, 98), (Porras and Valdes, 98). The principal design goal of EMERALD is monitoring a large enterprise network with thousands of users connected in independent administrative domains. In order to satisfy this, it employs a hierarchical building block approach.

The network surveillance is achieved through three hierarchical layered analyses: service, domain and enterprise-wide analysis. On the service analysis level, EMERALD handles the misuse of individual components and network services within a single domain. On the domain analysis level, it detects the misuses visible across multiple network services and components. On the enterprise-wide level, it monitors the co-ordinated misuses across multiple domains. This hierarchical analysis is achieved through executing service monitors, domain monitors and enterprise-wide monitors respectively on local hosts, the entry points of independently administered sub-network and gateways. The detection decisions made by each monitor are disseminated to other monitors at any level for performing global analysis.

Each monitor employs a statistical profile engine for anomaly detection, a signature analysis engine for misuse detection, a resolver for the co-ordination of other monitor reports and detection responses. In addition to these main components, each monitor adopts a resource object that presents the analysis targets and corresponding profile events. This object-oriented approach provides EMERALAD with good configurability and extendibility. Also, it provides a message-handling component that supports the interoperability between various monitored hosts.

Although the hierarchical building block approach of EMERALD provides most of network requirements such as efficiency, scalablility, extendibility, configurability and global analysis, its hierarchical manner leaves some problems to support complete robustness and fault tolerance.

○ Co-operative security managers

Co-operative security managers (White et al., 96), (White and Pooch., 96) propose another distributed approach by placing intrusion detection responsibility on each local host. Its principal design goal is to guarantee safe scale. It avoids employing any central control server and a hierarchical structure. Instead, the several background processes on each local host provide local intrusion detection, distributed (network) intrusion detection, intruder tracking, intruder handling and user interface functions. In particular, a distributed intrusion detection component and an intruder tracking handling component plays a significant role to support distributed detection. The distributed intrusion detection component co-ordinates local decisions made by local intrusion detection component and the intruder tracking component finds the origin of intruder through the communications of CSMs on other hosts. However, the detailed mechanisms of each component have not described yet. Therefore, it is difficult to evaluate its configurability, extendibility, adaptabliliy and efficiency.

○ Autonomous Agent for Intrusion Detection (AAFID)

Autonomous Agents for Intrusion Detection (AAFID) launched by the COAST project team (Balasubramaniyan *et al.*, 97), (Balasubramaniyan *et al.*, 98), (Crosbie and Spafford, 94), (Crosbie and Spafford, 95a), (Crosbie and Spafford, 95b) is the first attempt to use autonomous agents for network intrusion detection. Similar to CSM, AAFID operates without any central server. Instead, each independent autonomous agent is responsible for monitoring only a small aspect of the overall system and a number of agents operate concurrently. Through the co-operation among them, AAFID can make a coherent inference and finally make a global decision.

AAFID consists of main five components: agents, transceivers, monitors and user interfaces. On each host, any number of agents, a transceiver, a monitor and a user interface operate concurrently. Each agent monitors a certain aspect of a host and reports abnormal behaviour to a transceiver running on the same host. A transceiver is a communication point of a number of agents running on the same host. It receives reports from agents and performs the appropriate action after receiving reports. Also, it starts and stops agents. While a transceiver controls the agents on a local host, a monitor controls the communication between different local hosts.

Currently, the second prototype of AAFID is released and this completes only communication mechanism between components and hosts. Therefore, it has not been proved yet that the overall architecture actually supports the presented research claims. These are good extendibility, configurability, scalability, global analysis, robustness and adaptability. However, before proving these claims, this architecture has crucial deficiency at maintaining efficiency. It places too many overheads on local hosts. In particular, they plan to implement evolving agents to sustain adaptability (Crosbie and Spafford, 95b), (Crosbie and Spafford, 94). However, it already requires local hosts to process computationally expensive work such as lots of communication mechanisms, auditing mechanism and analysis of audit trails. Even though the good efficiency of AAFID was claimed in (Balasubramaniyan *et al.*, 98), it has not shown a clear solution to resolve this problem.

○ NetStat (Network-based State Transition Analysis Technique)

NetStat is a network-based IDS, which is extended from STAT (State Transition Analysis Tool) (Vigna and Kemmerer, 98). STAT is a graphical tool to elucidate the key actions to cause an intrusion that should be described in the rules for misuse detection. The extended NetStat provides a network state transition diagram to describe network topology, network service configuration and network intrusion scenarios. NetStat system automatically determines which events have to be monitored and where the monitors need to be located via the analysis using a NetStat diagram.

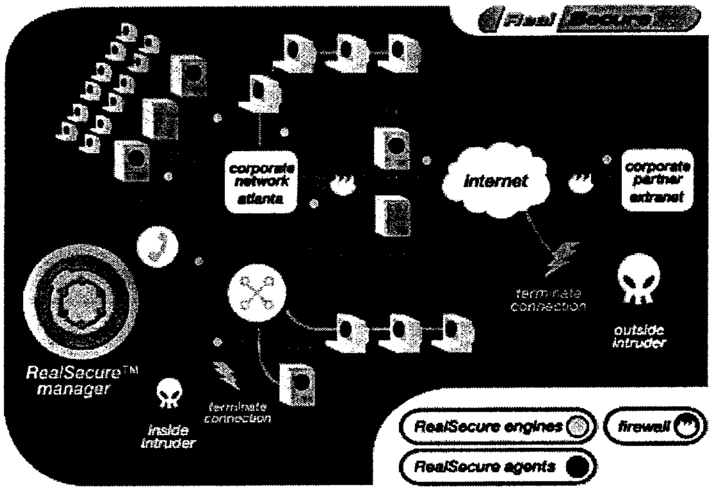
It consists of a central server which interacts with a security officer and a collection of distributed probes. The central server has a network fact base, a state transition scenario base and an analyser. The network fact base contains the information about network topology and network services. The state transition scenario base stores the set of state transition diagrams representing the intrusion scenarios to be detected. When intrusions to be detected are determined, the analyser accesses the network fact base and the state transition scenario base. By querying these databases, the analyser generates a set of probes. Each probe is equipped with the information about which events have to be collected, where each probe should locate, which information about the network topology and services is required and the descriptions of selected intrusions. The corresponding information to each probe is customized according to a particular sub-network, which the probe is intended to monitor. A set of probes, each of which is customized to a specific sub-network, is disseminated to specified monitor points. While they monitor a specific sub-network, they also interact each other for perform global analysis.

The overall architecture of NetStat system is somewhat similar to that of the proposed artificial immune system in this research even though it does not use the analogy of nature. Like the proposed artificial immune system, it will show good robustness and fault tolerance by distributing a large number of probes and their uniqueness customized to the specific local points. Also, it does not seem to show any serious problem in satisfying other requirements of network-based IDS, efficiency, scalability, global analysis, configurability and extendibility. However, the advantage of the artificial immune system over NetStat is its autonomous adaptability. The artificial immune model automatically maintains its adaptability such as providing the automatic event selection via the clonal selection and the gene library maintenance. Even though the state transition diagram helps NetStat to adjust intrusion detection rules, it still has to implement an analysis engine, which converts presented state diagrams into actual detection rules. However, this is an avoidable problem of implementing misuse detectors. Rather criticising this approach, we can perhaps suggest that a misuse detector of NetState style can be a good reciprocal component to the artificial immune model in order to detect known intrusions.

제7절 국외 상용화 제품

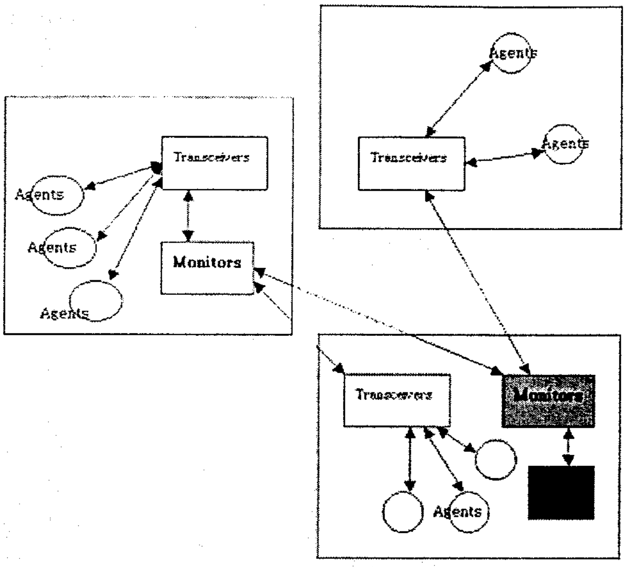
○ RealSecure

ISS(Internet Security System)의 침입탐지를 위한 가장 널리 상용화된 보안 상품 중의 하나다. network-based와 host-based 침입 탐지를 동시에 할 수 있다. 사용자는 제공하는 filter들을 그 시스템에 맞게 on/off만 할 수 있고, 사용자 정의 filter는 지정할 수 없다.



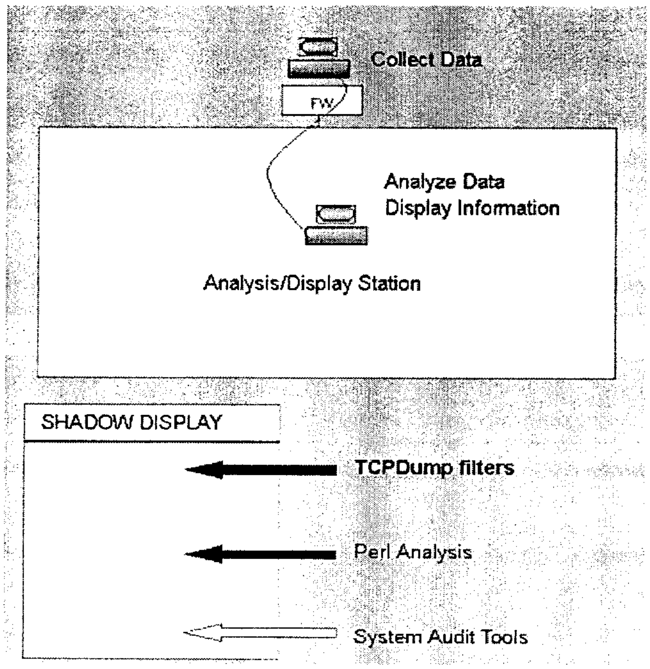
○ AAFID(autonomous Agents For ID)

purdue 대학에서 만든 IDS 시스템으로서, 기존의 IDS문제점을 보완하기 위해서 각각의 agents가 독립적으로 수행가능하고 전체 모니터링하는 부분과 연동 가능한 구조를 가지는 autonomous agents로 구성된다.



○ Shadow 1.6

shadow의 전체적 구성은 packet collecting 부분 여러개와 packet analysis & display하는 부분을 centralize하게 하나로 구성된다.



Emerald(Event Monitoring Enabling Responses to Anomalout Live Disturbances)
Unix(특히 SunOS 5.5 ~ 5.8)에서 host based IDS solution으로서 JAVA를 이용한 GUI환경과 user defined policy 기능을 제공한다. Solaris BSM을 사용하여 구현했고 실시간 탐지와 audit data를 이용한 추후 탐지가 가능하다.

제8절 국내의 상품화

현재, 국내에서도 침입탐지시스템의 중요성에 대한 인식이 높아져가고 있다. 인터넷데이터센터(IDC)나 인터넷 쇼핑몰, 그리고 기업의 정보보호 차원에서도 침입탐지시스템은 기존의 침입차단시스템(Firewall)과 함께 정보보호를 위한 필수 구성요소 자리잡아 가고 있다.

국내에서 IDS 에 관한 학계의 연구는 주로 Java를 이용한 에이전트에 대한 IDS 연구와, 데이터마이닝 기법을 적용한 IDS 연구가 가장 활발하다. [2] [3]
한국정보보호센터에서는 침입차단시스템(Firewall)에 이어 침입탐지 시스템에 대한 평가를 실시하고자 평가기준을 개발, 지난 7월 29일 고시하고 편재 평가에 앞서 평가자문을 해주고 있다. 현재, 국내에서 자체기술로 침입탐지 시스템을 개발하였거나, 개발중인 업체는 약 20여개이며, 이들중 절반이상이 평가를 받기위해 평가자문을 신청한 상태이다.

정통부에서는 침입탐지 시스템의 시장규모가 400~500억에 이를 것이라고 발표하였으나, 공공기관 등의 수요가 많기 때문에 정부예산이 변수로 작용할 것으로 보이며, 업계에서는 내년도 침입탐지시스템 시장을 약 200억정도로 추정하고 있다.

[2] 악성 이동 에이전트에 대한 실시간 CORBA 기반 침입 탐지 시스템의 설계 (고려대 1999)
[3] 데이터 마이팅 기법을 적용한 침입탐지 모듈설계 (조선대 1999)

국내 침입탐지 시스템 제품현황

항 목 회사명	시스템 설치 및 설정									침입탐지 및 대응
	제품명	시스템 환경	출시시점	시스템 1대가격 (만원)	하드웨어 요구사항	소프트웨어 요구사항	알람종류	원격관리 제공유무/ 방법	타기종간 호환성	
데이터게이트 인터네셔널	시큐레이다 1.0	호스트+ 네트워크 기반	2000.7. 출시	400	Sun Ultra5 (Manager)	Solaris 2.6	콘솔문자, 콘솔알람, 이메일	Yes/ 웹브라우저	No	공격발생시간, 공격위험성, 공격방법, 취약점 개선방법
디엠디	Network Police 1.0	네트워크 기반	2000.9. 출시예정	1,500 -2,500 (사용자수)	PentiumII, 333Mhz, RAM 64MB, HDD 150MB	Win 9x, Win 2000ME	콘솔문자, 콘솔 알람, 호출기 혹은 핸드폰, 이메일, SMS	No	Yew/ IETF IDWG Data Model	세션 차단으로 대응후 공격의 위험성을 경보함
디지탈이지스	Aegis IDS 1.0	네트워크 기반	2000.8. 출시	컨설팅 수행후 결정	Pentium-III, 500Mhz, RAM 64MB, HDD 18G, Fast Ethernet NIC	Linux, Solaris 2.5.1 이상	콘솔문자, 콘솔 알람, 호출기 혹은 핸드폰, 이메일	Yes/ Aegis IDS Remote Control Agent 사용	Yes/ Aegis IDS Common Alert File을 통한 Firewall 과의 연계	관리자가 지정한 대응행동을 취하고 탐지된 공격에 대한 설명과 취약점 개선방법 제시
시큐아이닷컴	Secu(i) IDS-R	라우터 기반	2000.9. 출신예정	응답안함	라우터	Win95/98/ 2000/NT 계열	콘솔문자, 콘솔 알람, 이메일	Yes/ Console 이용	Yes/SNMP	공격 위험성 대책관련 근거
아이에스피테크	ISP-IDS	호스트+ 네트워크 기반	2000.10 .출시예정	3,000	SUN 계열 W/S	Solaris 2.x	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/ SSL 이용	Yes/ TCP/IP 기반 자체프로토콜	해당 세션 단절후 관리자에게 통보
LG전자	Safe Zone	호스트+ 네트워크 기반	응답안함	3,500	CPU333Mhz, RAM 521MB, HDD9G	Solaris 2.5,Win95	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/ 관리 콘솔에서 원격제어	No	공격을 막을 수 있는 방법, 취약점 개선방법
웰넷정보통신	iCOP Pro2000	호스트+ 네트워크 기반	2000.7. 출시	3,500	Pentium 200Mhz, HDD기가급, 사운드카드	Win 2000/ NT4.0	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/iCOP/Center 사용	Yes / OPSEC 지원	공격의 위험성, 공격을 막을 수 있는 방법, 취약점 개선방법

윈스테크넷	SNIPER	네트워크 기반	2000.4. 출시	응답안함	Pentium 500Mhz, 256MM, HDD9G, 10/100 NIC	Red Hot Linux Kernet 2.2	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/ 웹을 이용한 C/S 방식, IAP를 이용한 통제관리	Yes/ 지원관제센터 시스템과 연동하기 위한 IAP	위험요인, 공격유형 설명, 해결방안
융시스템	NTracer2000	호스트기반	2000.7. 출시	응답안함	Pentium-II, RAM 64MB, HDD 100MB	Win 2000/ NT4.0	콘솔문자, 콘솔알람, 로깅	Yes/ TCP/IP기반통신	No	사건에 대한 이벤트 설명수준
인젠	NeoWatcher@ESM	네트워크 기반	2000.8. 출시	4,500	Pentium-II, 500Mhz, RAM 512MB, HDD 2GB, NIC2EA	Win 2000	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/ 전용관리 프로그램 (NeoAdmin)	Yes/ CISCO Router, SSL	공격방법 설명, 방어 방법 설명, 공격영향
정보보호 기술	Trust Sensor for Network 1.0 (Manager)	네트워크 기반	2000.7. 출시	3,400	Pentium-II, RAM 256MB, HDD 9GB	Win 95/98/2000/NT	콘솔문자, 콘솔알람, 이메일, 강제종료후 공격차단통보	Yes/ 네트워크 세그먼트에 Sensor 설치, 원격에 Manager 설치	Yes/ OPSEC, SSL	공격에 대한 설명, 공격위험성, 탐지에 대한 False Positive 및 ㄹ민ㄷ Negative, 공격대응방법, CERT 등 관련기관 권고, Firewall과 연동후 결과보고
케이원시스템	NetPolice	네트워크 기반	2000.9. 출시예정	2,000	Pentium-II, RAM 256MB	Red Hot Linux 6.1	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/ 웹브라우저	Yes/ TCP/IP	탐지 데이터를 분석한 통계자료를 통해 취약점과 개선점 컨설팅
펜타시큐리티시스템	Siren3.0	호스트+ 네트워크 기반	2000.7. 출시	2.000	Sun Ultra, RAM 128MB, HDD1G	Solaris 2.5.1, 2.6.7, Win 2000/NT 4.0, Linux 지원	콘솔문자, 콘솔알람, 호출기 혹은 핸드폰, 이메일	Yes/ 다중콘솔	No/ 단, 로그 파일 통해 타시스 ㅔ과의 인터페이스 가능	취약성 설명, 취약성 발견시기, 공격영향, 대응방법

	침입탐지 및 대응		관리 편이성 및 보고기능										기술지원과 서비스		
	분석방법	자체 Stealth 기능	DB업데이트	업데이트 주기	업데이트정보사용자전송	관리자인증메커니즘	보고서종류	침입세션재현	사용자 침입패턴추가	대규모전산망관리기능	통계자료제공	추적기능	무료서비스기간	연중수시	탐지패턴업데이트비용
데이터게이트 인터네셔널	룰베이스, 프로파일베이스	No	원격수동	비정기적	Yes	Timestamp, ID, 패스워드	공격시간별통계, 공격종류별통계. 요약, 상세보고서	Yes	Yes	Yes	Yes	무	12	Yes	무료
디엠디	룰베이스	Yes	자동	비정기적	No	관리자등급별 계정관리	공격시간별 통계, 공격종류별 통계, 침입에 의한 영향편입, 서비스별 트래픽, 호스트별 트래픽, 블랙리스트, 경보내역, 공격에 대한 대응	Yes	Yes	No	Yes	무	12	Yes	무료
디지털이지스	룰베이스	Yes	자동	비정기적	Yes	Aegis IDS Remote Control Agent와 Aegis IDS Engine간의 암호화	공격시간별통계, 공격종류별 통계, 상위10위 공격 패턴, 공격대응, 감사자료, Aegis IDS 엔진상태, 불량 메일송수신자 리스트	No	Yes	Yes	Yes	Yes	12	Yes	무료
시큐아이닷컴	룰베이스	무	로컬수동	비정기적	Yes	Login ID/PIW, IDS 사용자별 권한 설정	공격시간별통계, 공격종류별 통계	No	Yes	Yes	Yes	무	무	무	무
아이에스피테크	룰베이스, 패턴매칭, 상태전이	No	자동	정기적 2주	Yes	콘솔에서만 패스워드 인증 및 SSL을 이용한 원격관리인증	공격시간별통계, 공격종류별 통계, 침입내역분석	No	Yes	No	Yes	무	12	Yes	무
LG전자	룰베이스	No	원격수동	정기적 4주	Yes	패스워드 방식	공격시간별통계, 공격종류별 통계, 서비스별 통계, IP별 통계, 프로토콜별 통계	No	Yes	Yes	Yes	무	6	Yes	8~15%
웰넷정보통신	룰베이스, 프로파일베이스	Yes	자동	비정기적	Yes	자체사용자 관리도구 제공	공격시간별통계, 공격종류별 통계, 침입에 의한 영향평가, 공격호스트별 통계, 타겟 호스트별 통계, 프로토콜별 통계	Yes	No	Yes	Yes	Yes	12	Yes	무료
윈스테크넷	룰베이스	Yes	자동	비정기적	Yes	SSL을 이용한 Time Stamp 암호화기법사용	공격시간별통계, 공격종류별 통계, 침입에 의한 영향평가, 상세세션, Net 사용량	Yes	Yes	Yes	Yes	무	12	Yes	무료
융시스템	룰베이스	No	로컬수동	비정기적	No	관리자 프로그램 실행시 패스워드 입력	사건발생 회귀분석, 사용자별 사건발생 회귀분석, 컴퓨터별 사건발생 회귀분석	무	Yes	무	No	무	12	No	무

	침입탐지 및 대응		관리 편이성 및 보고기능										기술지원과 서비스		
	분석방법	자체 Stealth 기능	DB업데이트	업데이트 주기	업데이트 정보 사용자전송	관리자인증메커니즘	보고서종류	침입 세션 재현	사용자 침입 패턴 추가	대규모전산망 관리 기능	통계 자료 제공	추적 기능	무료 서비스기간	연중 수시	탐지 패턴 업데이트 비용
인젠	룰베이스, 프로파일베이스	Yes	자동	정기적 4주	Yes	ID, 패스워드 기반 인증, 코딩된 인증파일	공격시간별통계, 공격종류별 통계, 프로토콜별 사용통계, 서버별 사용통계, 시간.일.월별 사용통계, 웹사용통계, 다운된 서비스서버 통계, 사용자정의	Yes	Yes	Yes	Yes	무	12	무	무료
정보보호기술	보안정책기반분석, 통계적분석, 시그너처기반분석	Yes	원격수동, 로컬수동	정기적 3개월	Yes	ID, 패스워드 기반 인증, Manager와 Sensor간의 SSL을 이용한 인증 및 데이터보호	공격시간별통계, 공격종류별 통계, 정기.비정기 보고서	No	Yes	Yes	Yes	무	12	Yes	20%/년
케이원시스템	페트리넷	Yes	원격수동	정기적 4주	Yes	무	공격시간별통계, 공격종류별 통계, 침입에 의한 영향평가, 침입IP별, 취약IP별, 취약 Port별	무	No	Yes	Yes	무	6	Yes	계약체결시 결정
펜타시큐리티시스템	롤베이스, Staelseical Method	No	SE에 의해 혹은 관리자가 다운받아 서버에 저장한후 서버에서 자동분배	비정기적 1개월	Yes	암호화	공격시간별통계, 공격종류별 통계, 호스트.그룹.전체, 호스트.OS별 통계, 일.주.월.년간 통계	No	No	Yes	Yes	No	12	Yes	무

이중 데이터게이트의 “시큐레이다”는 국내 보안환경의 필요성에 맞게 설계된 침입탐지 제품으로 네트워크 및 시스템에서 시도되는 해킹의 실시간 탐지와 경보, 공격유형 분석과 대비 및 공격차단등의 필수적인 기능 제공과 관리자가 효율적으로 보안시스템을 운영하고 관리할 수 있도록 하는 관리적 유용성을 제공한다.

시큐레이다에서 제공하는 기능은 가장 최근의 침입탐지시스템의 특징을 나타내고 있어, 국내 침입탐지시스템의 기능을 알수 있다.

○ 시큐레이다

기능

① 네트워크 정보를 추출하기 위한 패킷 수집 및 분석 기능

명확하고 단순한 침입 형태에 대한 자체 판정

(서비스 거부 공격, 취약한 네트워크 서비스 공격 등)

② 침입 패턴을 기반으로 침입 여부를 판정하는 기능

수집된 정보의 신속하고 정확한 분석을 통한 실시간적인 침입의 판정

기존의 침입 형태를 통하여 새로운 침입의 형태를 유추

(특정 서비스관련 침입패턴 탐지, 웹 관련 공격 등)

③ 시스템 기반의 침입을 탐지할 수 있는 기능

시스템의 오용 및 남용에 의한 파일시스템에 대한 침입 탐지

(불법적인 관리자 권한 취득 시도, 허용되지 않은 접속 시도, 등)

④ 웹 서비스를 기반으로 침입을 보고할 수 있는 사용자 인터페이스 기능

침입의 내용과 상태의 효과적인 보안 경보 전달

웹 서비스와 Java를 이용한 보안 경보 및 보안 관련 통계 자료전달

보고서 양식을 통한 침입의 결과, 분석 자료 및 통계의 출력

제5장 A Network-Based Intrusion Detection System 개발

An intrusion detection system (IDS) is an automated system for the detection of computer system intrusions. The main goal of an IDS is to detect unauthorised use, misuse and abuse of computer systems by both system insiders and external intruders. In parallel to rigorous investigation into intrusion prevention such as firewall and cryptography, the significance of research into IDS has been growing and various approaches have been suggested and developed (Balasubramaniyan et al., 1997), (Mykerjee et al, 1994). As one novel approach, a few computer scientists have proposed simple computer immune models for intrusion and computer virus detection (Forrest et al., 1997), (Forrest et al., 1996), (Kephart, 1994), (Somayaji et al., 1997). The promising initial results from these models motivate computer scientists to understand human immune systems more fully.

This research aims to unravel the significant features of the human immune system, which would be successfully employed for a novel network intrusion detection model. Several salient features of the human immune system, which detects intruding pathogens, are carefully studied and the possibility and the advantages of adopting these features for network intrusion detection are reviewed and assessed.

제1절 Human Immune Systems and Network Intrusion Detection

Early IDSs operated at the *host level*, whereas contemporary systems tend to be *network-based* (Mykerjee et al, 1994). *Host-based* IDSs monitor a single host machine using the audit trails of a host operating system and *network-based* IDSs monitor any number of hosts on a network by scrutinising the audit trails of multiple hosts and network traffic.

Both host-based IDSs and network-based IDSs mainly employ two techniques:

anomaly detection and *misuse detection* (Mykerjee et al, 1994). The anomaly detection approach establishes the profiles of normal activities of users, systems, system resources, network traffic and/or services and detects intrusions by identifying significant deviations from the normal behaviour patterns observed from profiles. The misuse detection approach defines suspicious misuse signatures based on known system vulnerabilities and a security policy. This approach probes whether these misuse signatures are present or not in the auditing trails. These two techniques have different strengths and weaknesses and should be reciprocal in a complete IDS (Mykerjee et al, 1994).

This research focuses on presenting the analogy between human immune systems and network-based IDSs. Somayaji *et al.*(1997) present more general principles and suggest various possibilities for a computer immune system. In contrast, this work concentrates on the design of competent *network-based IDSs*, and analyses the several outstanding features of the human immune system with this specific problem in mind.

1. Requirements of network-based IDSs

Before presenting the human immune system features, it is necessary to comprehend which functions are required to design a competent network-based IDSs. A careful examination of the literature allows the significant functions to be distilled into seven points:

○ ***Robustness***

it should *have multiple detection points, which are robust enough against the attack and any system faults on IDSs* (Balasubramaniyan et al., 1997), (Forrest et al., 1997). The critical weak point of an IDS is its failure and subversion by intruders. If intruders already know the existence of an IDS and can subvert it, then the effort to develop the IDS was futile.

○ ***Configurability***

it should be able to *configure itself easily to the local requirements of each host or each network component* (Balasubramaniyan et al., 1997), (Somayaji et al., 1997). Individual hosts in a network environment are heterogeneous. They may have different security requirements. In addition to hosts, different network components such as routers, filters, DNS, firewalls, or various network services may have various security requirements

○ ***Extendibility***

it should be *easy to extend the scope of IDS monitoring by and for new hosts easily and simply regardless of operating systems* (Balasubramaniyan et al., 1997), (Somayaji et al., 1997). When a new host is added to an existing network environment and especially when this new host runs a different operating system that has a different format of audit data, it is not simple to monitor it in a consistent manner with existing IDSs.

○ **S*calability***

it is necessary to *achieve reliable scalability to gather and analyse the high-volume of audit data correctly from distributed hosts* (Balasubramaniyan et al., 1997). In the case of the monolithic IDSs, the audit trail collection procedure is distributed and its analysis is centralised (Mykerjee et al, 1994). However, it is very difficult to forward all audit data to a single IDS for analysis without losing the data. Even if it scales for all audit data correctly, it may cause severe network performance degradation.

○ ***Adaptability***

it should be *dynamically adjusted in order to detect dynamically changing network intrusions* (Balasubramaniyan et al., 1997), (Somayaji et al., 1997). Computer system environments are not static. Users, vendors and system .administrators are constantly changing them. Therefore, the normal activities of networks and intrusions are also continuously changing according to this environment.

○ ***Global Analysis***

in order to detect network intrusions, it should *collectively monitor multiple events generated on various hosts to integrate sufficient evidence and to identify the correlation between multiple events* (Balasubramaniyan et al., 1997), (Mykerjee et al, 1994). Many network intrusions often exploit the multiple points of a network. Thus, from a single host, they might appear to be just a normal mistake. But if they are collectively monitored from multiple points, they clearly can be identified as a single attack attempt.

○ ***Efficiency***

it should be *simple and lightweight enough to impose a low overhead on the monitored host systems and network* (Balasubramaniyan et al., 1997), (Forrest et al., 1996), (Somayaji et al., 1997). A single IDS is expected to perform monitoring, data gathering, data manipulation and decision making. It may impose a large overhead on a system and could place a particularly heavy burden on CPU and I/O, resulting in severe system and network performance degradation.

Even though various approaches have been developed and proposed until now (Balasubramaniyan, 1997), (Mykerjee et al, 1994), no existing network-based model satisfies these requirements completely.

2. the Design goals of network-based IDSs

Upon analysis, the requirements identified above can be used to derive three main design goals of an effective network-based IDS. They are being distributed, self-organizing and lightweight.

가. Distributed

The first design goal is being distributed. A distributed network-based IDS delegates its responsibilities to a number of distributed components. A number of independent intrusion detection processes monitor only a small aspect of the overall system. They operate concurrently and co-operate with each other. If a network-based IDS is distributed, it will satisfy the following requirements.

○ ***Robustness***
for a distributed network-based IDS, the failure of one local intrusion detection process does not cripple an overall IDS even though it causes the minimal degradation of overall detection accuracy.

○ ***Configurability***
a single intrusion detection process can be simply tailored to local requirements of a specific host without considering the various requirements of other hosts.

○ ***Extendibility***
even when a new host running a different operating system is added to a network, it is easy to add a new intrusion detection processes on this new host. This is because intrusion detection processes are independent and thus existing

processes do not need to be modified when a new intrusion detection process is added.

○ ***Scalability***

because audit data collection and its analysis take place in the same place, at a monitored local host, the high volume of audit data is distributed amongst many local hosts. Hence, distributed IDSs are more scalable than IDSs based on a single central server.

나. Self-Organization

The second goal is being self-organising. Without a central controller having predefined information, a self-organising network-based IDS automatically learns intrusion signatures which are previously unknown and/or distributed. This is achieved through the interaction with changing network environments, various security requirements and other intrusion detection processes. If a network-based IDS is self-organising, it will satisfy the following requirements.

○ ***Adaptability***

it is highly adaptive because there is no need for manual update of its intrusion signatures as network environments change.

○ ***Global analysis***

the overall intrusion detection system simply provides the global analysis. This is because it is self-organising from the interactions among a large number of various intrusion detection processes.

다. Lightweight

The third design goal is being lightweight. A lightweight network-based IDS does not impose a large overhead on a system or place a heavy burden on CPU and I/O. If a network-based IDS is lightweight, it will satisfy the last requirement.

○ ***Efficiency***

by placing minimal work on each component of the IDS, the main jobs that should be performed by local hosts and networks are not adversely affected by the monitoring.

3. Overview of human immune systems

Before we can identify which features of human immune systems may prove useful in the design of an effective network-based IDS, it is necessary to investigate the major mechanisms of the human immune system. An overview is presented in this section (largely based on (Paul, 1993), (Tizard, 1995)). The overall human immune system is implemented through the interactions between a large number of different types of innate and acquired cells rather than the function of one particular human organism. From a large number of different cells, lymphocytes (white blood cells), play a central role. Their main mechanism is distinguishing self cells, which are the cells of human body, from non-self cells, which are dangerous foreign cells. Each lymphocytes is specialized in reacting to a limited number of structurally related harmful foreign cells, known as antigens. Lymphocytes have the specific binding areas, called receptors, which have complementary shapes to the determinants of antigens, called epitomes. A specific antigen is recognised by its epitopes binding to lymphocyte antibody receptors.

Lymphocytes are classified into two main types: B-cells and T-cells. B-cells are antibody secreting cells and T-cells kill antigens or help or suppress the development of B-cells. Both B-cells and T-cells have their own unique genetic structures. Both B-cells and T-cells are expressed by several chains of DNA (*gene libraries*) and each chain has a variable domain and a constant domain. The genes in a variable domain are highly variable from one to another and this determines the specific binding area to antigens. The genes in the constant domain are invariable and show various biological effects when B-cell antibody receptors bind to antigen epitopes. B-cells and T-cells are developed in the bone marrow and the thymus respectively. At the bone marrow and the thymus, several gene libraries uniquely corresponding to domains of B-cells and T-cells contain the candidate genes to express B-cell and T-cell receptors. A specific receptor is generated by selecting gene segments randomly from gene libraries and joining them. Furthermore, in order to generate diverse receptors, they adopt a progressive series of genetic operators during their development processes. These include gene rearrangements, choosing different joining sites, somatic mutation, class switching and others (the details of these genetic operators are presented in (Tizard, 1995)).

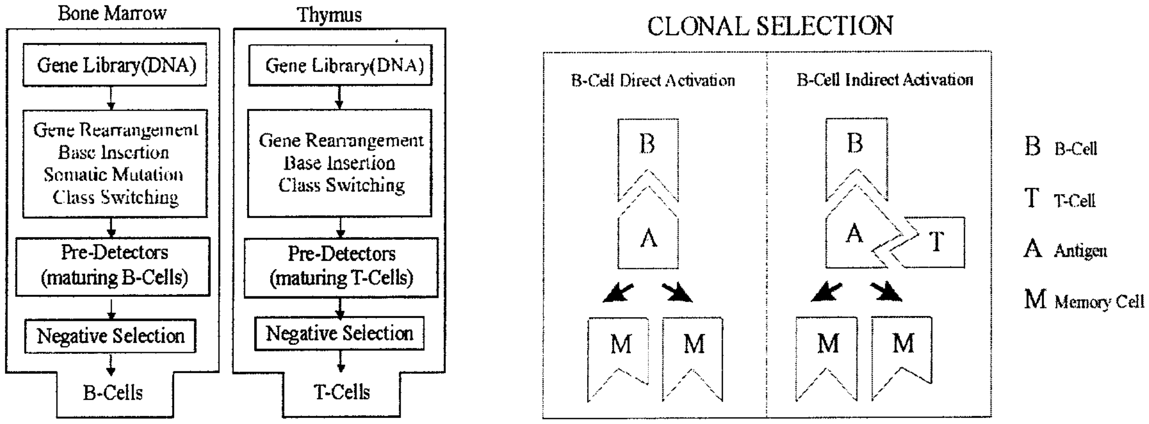


그림 20 : Development of B-cells and T-cells (left). Clonal selection (right).

Before leaving the bone marrow and the thymus, maturing B-cells and T-cells have to pass the last test, negative selection. In B-cell and T-cell development process, totally new cell receptors can be generated via various genetic operators. Therefore, it leaves the possibility for randomly generated receptors to bind to self cell epitopes. To prevent this, when maturing B-cells and T-cells bind to self cells circulating through the bone marrow and the thymus, they are killed instead of being released into a body. Figure. 20 (left) shows the development of B-cells and T-cells in the bone marrow and the thymus.

Mature B-cells and T-cells that pass the negative selection are released from the bone marrow and thymus. Both B-cells and T-cells continuously circulate around the body in the blood and encounter antigens for activation and evolution. The antibodies of B-cells, which recognize harmful antigens by binding to them, are activated directly or indirectly. When B-cell antibody receptors bind to antigen epitopes with strong affinity above a threshold, they are directly activated. On the other hand, B-cell antibody receptors can bind to antigen epitopes with weak affinity below a threshold. In this case, B-cells need the help of T-cells and Major-Histocompatibility Complex (MHC) molecules to be activated. MHC molecules have two important functions to help B-cell activation. Firstly, they bind to the fragments of antigens specially hidden inside cells, (not visible on the cell surface) and secondly, they transport these fragments to the B-cell surface. When B-cell antibody receptors bind to antigen epitopes with weak affinity, MHC molecules try to find some hidden antigen inside cells. When MHC molecules find them, they transport them on the surface of B-cells. The receptors of T-cells are genetically structured to recognise the MHC molecule on the B-cell surface. Thus, T-cells can bind to MHC molecules on B-cell surfaces. When the T-cell binds to MHC molecule with strong affinity, it sends a chemical signal to the B-cell which allows it to activate, grow and differentiate. What does make the T-cells determine the B-cell activation? One major difference between B-cells and T-cells is that only B-cells perform somatic mutation, which is a very high rate of mutation, to increase its diversity when they are developed in the bone marrow. Hence, B-cells have more various and new receptors that T-cells. In addition, the thymus is centrally located while the bone marrow is distributed. Thus, most of self cells pass through the thymus and hence the negative selection in the thymus is more reliable than that in the bone marrow.

With or without the assistance of T-cells, B-cells are activated and this activation is immediately followed by clonal selection. The activated B-cells are divided into a number of clones that have the same antigen-binding properties as parent B-cells or mutated antigen-binding properties. On the other hand, if any antigen cannot activate B-cells within a limited time, they rapidly die off. Therefore, based on the existing antigens, only the fittest B-cell antibodies survive. Because antigens constantly change, the efficiency of detection is maintained by the evolution of B-cell antibodies via clonal selection. Furthermore, when antigens activate B-cells, they produce memory cells for the reoccurrence of same antigens in the future. Because of these memory cells, antigens that have been identified previously are detected much quicker (known as the secondary response). Figure 20 (right) shows the generation of memory cells via clonal selection.

In contrast, idiotype antibodies, which are anti-antibodies, can activate antibody receptors. Immune systems let antigens and anti-antibodies compete to bind to antibodies and the winning anti-antibodies can suppress binding between antigen and antibody. The inhibition of idiotype antibody against antigen contributes to regulate an appropriate level of immune responses. Immunologist, Jern, proposed immune network theory (Dasgupta and Attoch-Okine, 1997), (Farmer et al., 1986) based on understanding the role of the idiotype antibody. He views immune systems as a functional network of lymphocytes and the network at any moment has the dynamic state of internal interactions of antibodies and antigens. The continuous chain of differentiation by antigens and suppression by idiotype antibodies can form a large-scaled network. When this network finally reaches the equilibrium status between suppression and stimulation, it determines the overall immune system.

4. Human Immune System features for Network-based IDS's

By performing a careful analysis of the complex capabilities of human immune systems summarised above, it is possible to identify several significant features for network-based intrusion detection. Upon investigation, it becomes clear that specific features can act together in order to satisfy each of the three design goals of competent network-based IDSs: being distributed, self-organising and lightweight.

가. Distributed

The human immune system is distributed. The following mechanisms allow the human immune system to detect antigens in a truly distributed way.

○ Immune Network

the human immune system is implemented through the interactions between a large number of different types of cells. Instead of employing a central co-ordinator, human immune systems sustain the appropriate level of immune responses by maintaining the equilibrium status between antibody suppression and activation using idiotype antibodies (Dasgupta and Attoch-Okine, 1997), (Farmer et al., 1986) .

○ Unique Antibody Sets

the human immune system generates various groups of antibodies to detect different antigens. Its evolution mechanism through natural selection of gene libraries and clonal selection maintains a number of different sets of antibodies. Therefore, each antibody set is unique and independent. These properties do not require any central co-ordinator and they allow the human immune system to detect antigens in a local antibody level (Somayaji et al., 1997).

나. self-organisation

The overall immune response is composed of three evolutionary stages: gene library evolution generating effective antibody, negative selection eliminating inappropriate antibodies and clonal selection cloning well-performing antibodies. These three stages are self-organizing rather than being directed by a central organ or predefined information.

○ Gene Library Evolution

antibodies recognize antigens by the complementary properties that only antigens, not self-cells, show. Thus, some knowledge of antigen properties is required to generate competent antibodies. The human immune system learns this knowledge by its evolution over time and hence provides us with efficient and knowledge-rich DNA. Because of this evolutionary self-organisation process, our gene libraries act as archives of information on how to detect commonly observed antigens (Tizard, 1995).

○ Negative Selection

as the second stage, this eliminates inappropriate and immature antibodies, which bind to self. The important constraint that the immune system has to satisfy is not to attack self cells. Instead of having any global information about self cells, this constraint satisfaction is performed in the thymus and bone marrow by presenting self cells, and removing any antibodies which attack these cells (Forrest et al., 1997), (Paul, 1993).

○ Clonal Selection

as the third stage, this process clones antibodies performing well. In contrast, antibodies performing badly die off after a given life time. Thus, according to currently existing antigens, only the fittest antibodies survive. Similarly, instead of having the predefined information about specific antigens, it self-organises the fittest antibodies by interacting with the currently existing antigens (Paul, 1993), (Tizard, 1995).

다. lightweight

The human immune system is lightweight. The following mechanisms allow it to be lightweight and are focused on three ideas: i) how a vast number of antigens can be detected with a smaller number or antibodies, ii) how the known antigen information can be reused efficiently and iii) how numerous antibodies can be generated with a limited number of genes. Approximate binding, memory cells and gene expression provide the answers to these questions respectively.

○ Approximate Binding

The immune response activates when the affinity of antibody and antigen binding is above a certain threshold. In other words, a single antibody can detect any number of antigens as long as their affinity is above the threshold. This approximate binding contributes to increase the generality of immune systems (Forrest et al., 1997).

○ Memory Cells

memory cells store the genetic information of previously detected antigen epitopes and respond efficiently and quickly when they meet the same antigens in the future (Somayaji et al., 1997), (Tizard, 1995). Because memory cells have a longer life span than ordinary antibodies, they retain immunity without the need to create the same antibodies again.

○ Gene Expression

the immune system maintains antibody diversity in order to ensure the effective detection of a wide range of antigens. In an antibody development process, known as gene expression, several genetic mechanisms are employed to generate diverse antibodies from the gene libraries. The main idea of these mechanisms is that a vast number of new antibodies can be generated from new combinations of gene segments in the gene libraries (Paul, 1993), (Tizard, 1995).

In summary, this analysis shows that the human immune system is distributed through its immune network and unique antibody sets. It is self-organizing because of the three evolutionary processes of gene library evolution, negative selection and clonal selection. It is lightweight because of the generality of approximate binding and gene expression, and the efficiency of memory cells.

Since the human immune system is distributed, self-organising and lightweight, it clearly fulfils the design goals for network-based intrusion detection systems. Perhaps most importantly, the mechanisms used by human immune systems satisfy the three goals in an elegant and highly optimised way and this motivates future research harnessing such processes. Because of this study, it is thought that the application of computer immune systems to network-based intrusion detection is likely to provide significant benefits over other approaches.

제2절 An Artificial Immune Model for Network Intrusion Detection

Even though various approaches have been developed and proposed, no network-based IDS has satisfied all its requirements (Kim and Bentley, 1999a). This chapter roposes a novel approach to building a network-based IDS, which is inspired by a human immune system. (Kim and Bentley, 1999a) carefully studied the several salient features of human immune systems and showed the possibility and advantages of adopting these features for network intrusion detection. This chapter presents a more specific artificial immune model, which actually monitors a real-network, and describes the main components of this model. The next section starts by recalling three types of network-based IDS's.

1. Taxonomy of Network-based IDS's

According to the overall architecture, we categorise network-based IDSs into three groups: *monolithic*, *hierarchical* or *co-operative*.

○ MONOLITHIC Approach

The monolithic approach employs a central intrusion detection server and simple host audit programs running on multiple local hosts. Monitored local hosts transfer their collected audit trails to an intrusion detection server and then this server performs audit trail analysis. Most network-based IDSs which have been developed until now use this approach and run in real small-scale networks (Mykerjee et al, 1994). However, such methods show some critical deficiencies in their scalability, robustness and configurability. Firstly, as a network size grows, a huge number of audit trails needs to be transferred from local hosts to a central server. This causes severe degradation of the network performance and it is difficult to guarantee scalability. Secondly, if a central intrusion detection server is subverted or fails, the overall IDS becomes crippled. Thirdly, a single intrusion detection server should *uniformly* configure itself to the *various* local requirements of each host.

○ HIERARCHICAL Approach

The *hierarchical approach* was proposed to overcome the problems of the monolithic approach. It was designed to monitor large-scale networks, which have more than several thousand hosts. It defines a number of hierarchical monitoring areas and each IDS monitors a single area. Instead of transferring all the collected audit data from local hosts to a central IDS, each single IDS at any level of monitoring area performs local analysis and sends its local analysis results up to the IDS at the next level in the hierarchy. Thus, IDSs at higher levels only need to analyse transferred local reports collectively. The Graph-based Intrusion Detection System (GrIDS) (Staniford-Chen et al., 1996) and Event Monitoring Enabling Responses to Anomalous Live Disturbances (EMERALD) (Porras and Neumann, 1997) project propose this approach to monitor large-scale networks and they are still in progress.

The hierarchical approach seems to show better scalability by allowing local analyses at distributed local monitoring areas. However, other problems raised from the monolithic approach still remain. When the topology of the current network is changed, it causes a change of network hierarchy and the whole mechanisms to aggregate local analysis reports must be changed (Mykerjee et al, 1994). In addition, when a monitor residing at the highest level is attacked or crashed, then all network-wide co-ordinated intrusions, which are identified only by the global analysis of local results collected from distributed monitors at lower levels, easily escape detection.

○ CO-OPERATIVE Approach

The *co-operative* approach attempts to distribute the responsibilities of a single central server to a number of co-operative host-based IDSs. Each IDS is responsible for monitoring only a small aspect of a local host and a number of IDSs operate concurrently and co-operate with each other. Moreover, they can make a coherent inference and make a global decision. The difference of this approach from the hierarchical approach is that there is no hierarchy among distributed local IDSs. Therefore, the failure and subversion of any IDS does not always prevent the detection of co-ordinated attacks. The Co-operative Security Managers (CSM) project (White et al., 1996) and the Autonomous Agent For Intrusion Detection (AAFID) project (Balasubramaniyan et al., 1998) proposed this approach. In these proposals, it is claimed that most of problems encountered by the two approaches previously mentioned would be resolved. These projects are still in progress and the validity of this claim remains unproven. In particular, this approach raises a different problem, namely the maintenance of efficiency. It places too many overheads on monitored local hosts such as many communication mechanisms, auditing mechanisms and analyses of audit trails and these can be a significant encumbrance to them.

To summarise, various architectures of network-based IDSs have been proposed and here they have been grouped into three different approaches. However, each approach shows different problems and no network-based model completely resolves the encountered problems.

2. Artificial Immune Model Overview

The human immune system has been successful at protecting a human body against a vast variety of foreign pathogens or organisms (Tizard, 1995). This remarkable property is attractive to computer security researchers and artificial intelligence researchers. Based on the studies by immunologists, a growing number of computer scientists have proposed several different computer immune models (Dasgupta and Attocj-Okine, 1997). The main idea of these models is distinguishing self, which is normal, from non-self, which is abnormal. In this paper, with respect to network intrusion detection, we view the normal activities of monitored networks as self and their abnormal activities as non-self. Many sophisticated network intrusions such as sweeps, co-ordinated attacks and Internet worms are detected by monitoring the anomalies of network traffic patterns (Porras and Valdes, 1998) . Most network-based IDSs monitor network packets and their identified anomalies show critical signatures of these network intrusions (Mykerjee et al, 1994), (Tizard, 1995). Thus, the artificial immune model is designed for distinguishing normal network activities from abnormal network activities and expected to detect various network intrusions[4].

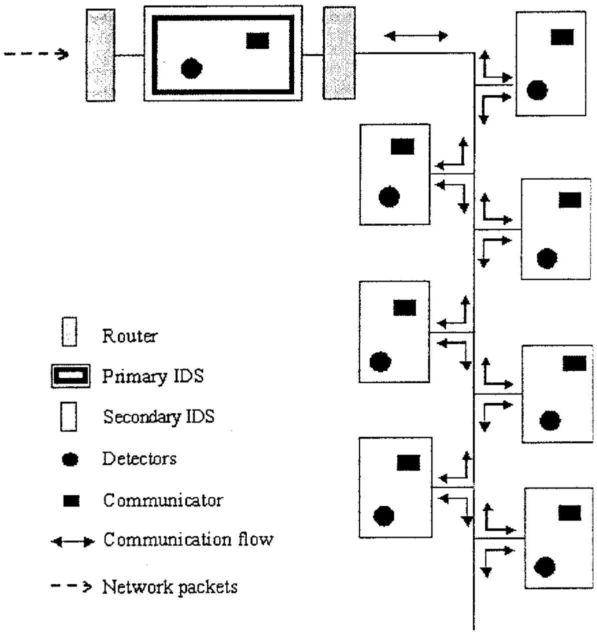


그림 21 : The Physical Architecture of an Artificial Immune System

The overall architecture of the novel artificial immune model developed as part of this work is presented in Figure 21. The artificial immune model for network

[4] Most network-based IDSs operating in real network environments monitor the audit trails generated by a local host together with the network activities. This kind of approach is more reliable at detecting various intrusions. Even though the artificial immune model proposed in this paper restricts its monitoring scope to network activities, it should be extended by monitoring local audit trails and this extension might be possible by employing one suggestion, a host-based computer immune system, introduced in (Somayaji et al., 1997).

intrusion detection consists of a primary IDS and secondary IDSs. For a human body, at the bone marrow and the thymus, various detector cells, called antibodies, are continuously generated and distributed to secondary lymph nodes, where antibodies reside to monitor living cells. The distributed antibodies monitor all living cells and detect non-self cells, called antigens, invading into secondary lymph nodes. For the artificial immune model, the primary IDS, which we view as the bone marrow and thymus, generates numerous detector sets. The architecture shown in Figure 21 is assumed to monitor a single network domain. Therefore, all the input network packets transferred to a monitored single network domain firstly arrive at the first router This assumption can be extended for monitoring large-scale networks which include a number of different domains. It is achieved simply by installing a single primary IDS on each domain and monitoring each domain independently. Each individual detector set describes abnormal patterns of these network traffic packets. It is unique and transferred to each local host. We view local hosts as secondary lymph nodes, detectors as antibodies and network intrusions as antigens. At the secondary IDSs, which are local hosts, detectors are background processes which monitor whether non-self network traffic patterns are observed from network traffic patterns profiled at the monitored local host. The primary IDS and each secondary IDS have communicators to allow the transfer of information between each other.

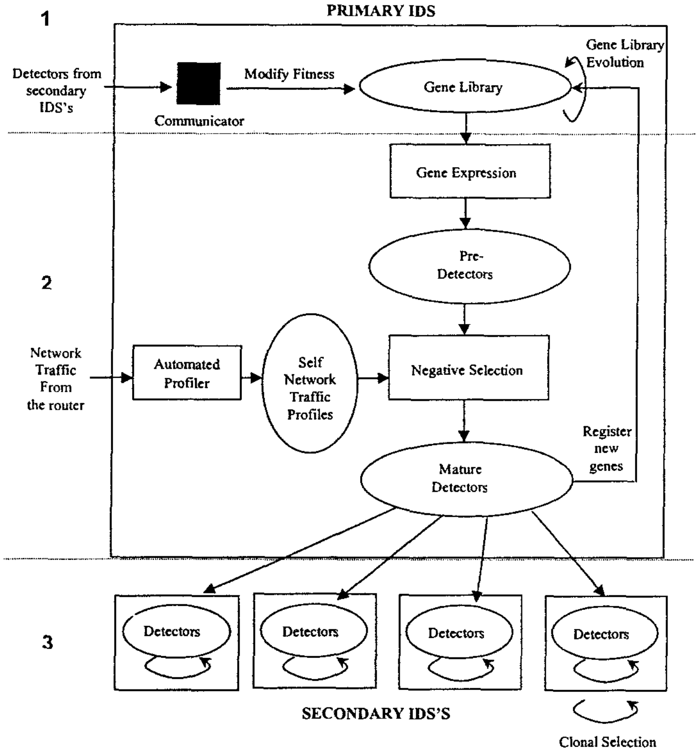


그림 22 : Conceptual Architecture of the Artificial Immune Model

Kim and Bentley (Kim and Bentley, 1999a) identified three main goals for designing an effective network-based IDS's: being distributed, self-organising and lightweight. Furthermore, they showed that the several sophisticated mechanisms of the human immune system allow it to satisfy these three goals. For the proposed artificial immune system, these mechanisms are embedded in three evolutionary stages: gene library evolution, negative selection and clonal selection. While the currently existing computer immune models focus on the use of a single significant stage according to their perceived purpose (Dasgupta and Attocj-Okine, 1997), (Forrest et al., 1997), (Mykerjee et al, 1994), the new artificial immune model proposed in this paper combines these three significant evolutionary stages into a single methodology. The overall conceptual architecture of the proposed artificial immune model is shown in Figure 22. In Figure 22, stage one indicates gene library evolution, stage two presents negative selection and stage three shows clonal selection. The functions in each stage and how these three stages operate together for performing network intrusion detection are described in the following two sub-sections: Primary IDS and Secondary IDS's.

가. Primary IDS

The primary IDS performs the first two evolutionary processes: gene library evolution and negative selection. At the gene library evolution stage, it aims to gain general knowledge on effective detectors. At the negative selection stage, it aims to generate a number of diverse detectors, which do not match self, and transfer a number of unique detector sets to distributed local hosts. In order to achieve these tasks, it contains the following components (shown in Figure 14).

At the first stage, a gene library is generated and maintained by an evolution process[5)]The *gene library* of the artificial immune model stores the potential genes of detectors and diverse genetic mechanisms generate new detectors. The potential genes are the selected fields of profiles to describe anomalous network traffic patterns. They are selected after understanding the detailed mechanisms of network protocol and their security holes (Porras and Valdes, 1998) . The initial genes might be set by the values of these fields that are observed when a previously known intrusion is simulated. They can be described by the number of packets, bytes, specific errors, etc of typical network services for a specific short period or one connection time (Mykerjee et al, 1994), (Porras and Valdes, 1998). If a new detector, which is generated from initial genes and transferred to a local host, detects anomalous network traffic activity, the genes comprising this detector will be added to the gene library. But, if the genes are already stored in the gene library, the fitness values of these genes are increased. If this process continues, the size of the gene library will grow. However, if the size of the gene library is limited, whenever the size is above a fixed length, the genes that have lowest fitness values will be removed from the gene library. This mechanism drives the artificial immune model to perform *gene library evolution.* This process allows the artificial immune model to learn knowledge of currently existing intrusions regardless of whether they were

3 It should be noted that this evolutionary process is a simulation of the natural evolutionary process for gene libraries. In nature, the DNA (gene libraries) of an organism cannot change within the lifetime of that organism. Evolution operates on populations of organisms, evolving gene libraries based on which organisms survive (i.e., how effective their immune systems are, throughout their lives). This is clearly computationally expensive, so in this model we treat the gene library as a population in itself and evolve it with a single artificial immune system. However, unlike gene library evolution, the other two evolutionary processes within the model operate in a conceptually similar manner to natural immune systems..

detected previously or not, making it self-organizing. Furthermore, its self-organizing feature allows it to be lightweight. This is because it does not have to contain all the information of intrusions that have been detected so far. Instead, it holds only the smaller and limited number of genes which currently survive.

At the second stage, the *gene expression* process generates various *pre-detectors* via rearrangement of selected genes, the selection of various gene-joining points, mutation of genes, which are randomly selected from the gene library. These mechanisms can lead to the generation of a vast number of possible pre-detectors from combinations of genes (Tizard, 1995). This process permits the artificial immune model to detect numerous intrusions using a smaller number of detectors, making it lightweight. The *automated profiler* produces a self network traffic profile of raw network traffic packets transferred from the first router. However, the raw network traffic volume is huge and the normal activity patterns are hidden. The automated profiling component reduces the huge volume of raw network packets into a self profile. The fields of the *self network traffic profile* are identical to those of the generated pre-detectors. In other words, specific values of these fields can determine whether the observed network activities are normal (the self-profiles), or anomalous (the pre-detectors). However, some pre-detectors can be false detectors because they have novelty generated via mutation in the gene expression process. These false pre-detectors are removed by the *negative selection* process, which matches them to a self network profile produced by an automated profiler. If the field values of pre-detectors match the field values of the self network traffic profiles, we can consider these new pre-detectors as false detectors which wrongly identify self as anomalies, and thus they are eliminated (Forrest et al., 1997). This process removes false pre-detectors by presenting self without any global information about self and hence it shows the property of self-organization.

Finally, the surviving detectors from negative selection become *mature detectors*. Before each detector set is transferred to an individual local host, the genes made up of mature detectors are newly registered in the gene library. Unique sets of detectors and self network traffic profiles are selected from these mature detectors based on each network connections in order to transfer them to local hosts. This selection guarantees the uniqueness of individual detector sets. These unique detector sets detect network intrusions independently in a local host level (Somayaji et al., 1997) and permit the artificial immune model to be distributed. The selected detector sets and self network traffic profiles are transferred to the second router and it distributes them to their corresponding secondary IDSs.

In order to perform above processes, the primary IDS needs to communicate with the secondary IDS's. For example, the former needs to transfer mature detectors to the latter and the latter needs to send newly found useful genes to the former. The *communicator* controls any type of communication between the primary IDS and the secondary IDS's.

나. Secondary IDS

The secondary IDS's perform the last evolutionary process: clonal selection. Its main tasks are detecting various intrusions with a limited number of detector sets and cloning the identical detectors that are performing well, producing memory detectors and driving the gene library evolution in the primary IDS. These tasks are achieved by the operations of several components: self network profiles, unique detector sets, network traffic anomaly detection, clonal selection of detectors, memory detectors and a communicator.

In order to perform *network traffic anomaly detection*, the detectors of *unique detector sets* and *self network profiles* transferred from the primary IDS are compared. First of all, the match strength between the field values of a detector and the self profile is measured. When this strength is over a pre-defined threshold, this process informs it to the communicator. This approximate binding helps make the artificial immune model lightweight. This is because one detector can bind to a number of different intrusions if only their match strength is over the threshold (Somayaji et al., 1997).

After detecting anomalies, the secondary IDSs perform *clonal selection*. When a new detector detects an abnormal network traffic activity, this detector remains as a *memory detector* in a secondary IDS and clones itself. The cloned detectors can be transferred to other hosts. They act as misuse detectors. They detect quickly the same intrusions in the future, which have previously detected. Furthermore, the genes of this detector will be added to the gene library in the primary IDS if they do not exist in the gene library or the fitness values of these genes will be increased otherwise. This drives the gene library evolution in the primary IDS. As the anomaly detection of detectors in local hosts continues, each local host will have more memory detectors and the number of detectors that need to be transferred to each local host will decrease. This process allows the model to be self-organised and lightweight. Instead of having the predefined information about specific intrusions, it self-organises the fittest detectors by detecting the currently existing intrusions. In addition, the evolved gene library and memory cells decrease the efforts to create various new detectors, helping to make the model lightweight.

The final decision of whether a network intrusion has occurred is made according to the collective decisions from several local hosts. The artificial immune model employs the agent communication mechanism suggested by (Balasubramaniyan et al., 1998). When suspicious activity is detected by anomaly detection process at any secondary IDS, it sends a signal to a *communicator*. The communicator increases the risk level and sends a signal to the communicators in other hosts and the primary IDS. Other communicators, which receive the signal, increase the risk level. If suspicious activities are found from several hosts within a short time, the risk level in each host and the primary IDS will be rapidly increased. When this risk level becomes above a certain threshold, a communicator can inform the breach of network intrusion to a security officer through a user-interface.

다. Summary OF Artificial Immune Model

The artificial immune model described above consists of the primary IDS and the secondary IDSs. It combines three evolutionary stages. *Gene library evolution* simulates the first stage of evolution, which learns knowledge of currently existing antigens. This process allows the model to be lightweight and self-organizing. *Gene expression* and *negative selection* form the second stage of evolution, generating diverse pre-detectors and selecting mature detector sets by eliminating false pre-detectors in a self-organizing way. The transfer of unique detector sets to the secondary IDSs also occurs at this stage, making the model distributed. *Clonal selection* is the third stage of evolution, detecting various intrusions with a limited number of detector sets using approximate binding, and generating memory detectors. This generality and efficiency results in the model being lightweight. In addition, this process drives the gene library evolution in the primary IDS. These three processes are co-ordinated across a network to satisfy the three goals for designing effective IDS's: being distributed, self-organizing and lightweight (Kim and Bentley, 1999a).

3. Discussion of Artificial Immune Model

To provide an indication of the advantages of this approach, the new artificial immune model suggested in this paper is now analysed with respect to the requirements of a network-based anomaly detector. Kim and Bentley (1999a) described the seven requirements of a competent network-based IDS. The proposed artificial immune model is assessed with respect to these seven requirements.

The proposed artificial immune model is distributed by using a unique detector set in a local secondary IDS for detecting local intrusions and employing communications among secondary IDSs for detection network intrusions. This distributed feature allows the model to be robust, configurable, extendible and scalable.

Firstly, the artificial immune model is ***robust***. The failure of any detector set residing at any local host does not cripple an overall artificial immune system even though it may cause some minor degradation of detection accuracy. Each detector set can still detect network intrusions even after the failure of the primary IDS. This is because each local host already has detector sets, which were transferred before the failure. Besides, if an intruder breaks through a local host and gains the information about how detectors describe anomalous behaviour, this intruder might attempt to use this information to disguise his or her activities. However, the uniqueness of each detector set makes this kind of attempt difficult.

Secondly, it is ***configurable***. Even though detectors are generated in the primary IDS, their usefulness is proved at a local level by employing clonal selection in each secondary IDS. Furthermore, this local level clonal selection drives the gene library evolution in the primary IDS. In other words, the generated detectors co-evolve to detect various intrusions and this co-evolution is led by the self profiles and existing intrusions in each local level. Therefore, the artificial immune model configures local requirements in a self-organised way disregarding various requirements of other hosts.

Thirdly, it is ***extendible***. When a new local host is added to a network, it simply needs to generate another detector set for the new host and install a secondary IDS consisting of an automated profiler, anomaly detection process, clonal selection process and a communicator without considering other hosts. These components are totally independent from the components at other secondary IDSs and thus they ensure that the artificial immune model is easy to extend. Fourthly, it is ***scalable.*** At initial stages, an artificial immune system might need to generate a large detector set. However, as it detects anomalies more and more, each local host will be equipped with more and more memory detectors and eventually will require very few new detectors to be transferred. Nevertheless, this requires the occurrence of a number of various intrusions within a practically short time. Therefore, the overall artificial immune mechanisms may be simulated by presenting a number of intrusions for a short time and this is used for the initial learning process before the launch of real intrusion monitoring by the artificial immune model.

In addition, the artificial immune model is self-organising by performing gene library evolution, negative selection and clonal selection. This property of self-organisation makes the model both ***adaptable*** and capable of ***global analysis***. Firstly, the negative selection process allows detectors to consider dynamically the self information at any moment. The clonal selection and the gene library evolution generate various detector sets that are the fittest for the recently encountered intrusions. Therefore, the newly generated detectors always dynamically learn knowledge about currently existing intrusions and self.

Furthermore, when a new intrusion is detected, these new abnormal patterns will be registered to the gene library of the primary IDS and remain as the memory detectors at the secondary IDSs. Therefore, the artificial immune model still can be highly adaptive. Secondly, global analysis is achieved via the communication between the primary IDS and the secondary IDSs and this communication mechanism is simple and autonomous, which does not require a global communication controller.

Finally, the artificial immune model is lightweight by detecting various intrusions using approximate binding and memory cells, performing gene library evolution and gene expression [6]. This lightweight feature provides good *efficiency*. Firstly, the approximate binding permits one detector to detect a number of different intrusions. Consequently, the model needs to generate a much smaller number of detectors than the number of intrusions that are expected to be detected. Secondly, as mentioned above, clonal selection generates memory detectors within local hosts. As the number of memory detectors increases, the number of new detectors required will decrease, resulting in a reduction of computation time. More importantly, as the detection of intrusions continues, a gene library collects useful genes. Through gene library evolution, these genes define detectors that have already proved their usefulness by identifying anomalies. Since such detectors use only the most useful features of the profile at any one time, this removes the need for each local host to perform feature selection during profiling. This feature certainly reduces the overheads of local monitored hosts compared to the co-operative approach. The final example of efficiency in the system is provided by the gene expression process. This process allows the artificial immune model to generate a huge number of detectors from a small number of genes in the gene library.

[6] Even though the novel evolutionary approach of the artificial immune model allows the secondary IDSs to be lightweight, it may impose some more work on the primary IDS. To resolve this problem, it may be designed as a parallel array of the primary IDSs (Carlberg, 1998) . For example, the first router which receives all network input packets outside a network domain can split network packets into groups of flow based on each connections. Then a number of different flow groups can be sent to each primary IDS. Each primary IDS will have the identical components that have been introduced in this paper and it generates specific detector sets and self profiles based on each connection. The specific detector sets and self profiles generated by an individual primary IDS are sent to the second router and this router can transfer them to a specific secondary IDS (a local host) within a domain.

제3절 Negative Selection of an Artificial Immune System

This chapter proposes the use of negative selection of artificial immune system for developing an effective network-based IDS. An overall artificial immune model for network intrusion detection presented in (Kim and Bentley, 1999b) consists of three different evolutionary stages: negative selection, clonal selection, and gene library evolution. Among these stages, the first stage, negative selection, is investigated in this chapter. We present practical problems of a negative selection algorithm when it is applied on network-based intrusion detection. This chapter is organised as follows; section 1 discusses a negative selection algorithm originally devised by Forrest, Hofmeyr, and Somayaji (1997). Section 2 describes details of network traffic packet data used in this work. Then, in section 3, detailed implementation points including genotypes, phenotypes, genetic operators and fitness functions are provided.

1. Related work

The basic idea of the human immune system is the ability to distinguish self, which is normal, from non-self, which is abnormal. For a human body, various detector cells, called antibodies, are continuously generated and distributed to a whole body. The distributed antibodies monitor all living cells and detect non-self cells, called antigens, invading into a human body. This main procedure is performed by three evolutionary stages described above and each stage plays its different and significant role in making the overall immune system function successfully (Kim and Bentley, 1999a).

가. Negative Selection of the Human Immune System

An important feature of the human immune systems is its ability to maintain diversity and generality. It is able to detect a vast number of antigens with a smaller number of antibodies. In order to make this possible, it is equipped with several useful functions (Kim and Bentley, 1999a). One such function is the development of mature antibodies through the gene expression process. The human immune system makes use of gene libraries in two types of organs called the thymus and the bone marrow. When a new antibody is generated, the gene segments of different gene libraries are randomly selected and concatenated in a random order, see figure 23. The main idea of this gene expression mechanism is that a vast number of new antibodies can be generated from new combinations of gene segments in the gene libraries.

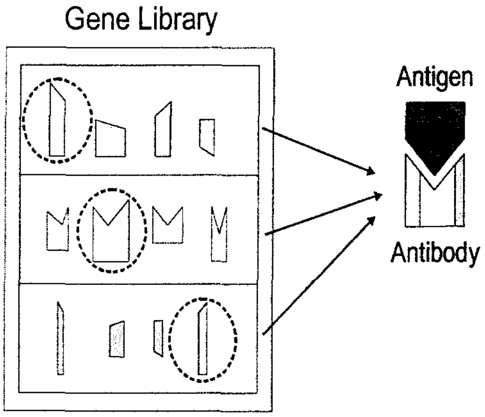


그림 23 : Gene Expression Process

However, this mechanism introduces a critical problem. The new antibody can bind not only to harmful antigens but also to essential self cells. To prevent such serious damage, the human immune system employs negative selection. This process eliminates immature antibodies, which bind to self cells passing by the thymus and the bone marrow. From newly generated antibodies, only those which do not bind to any self cell are released from the thymus and the bone marrow and distribute throughout the whole human body to monitor other living cells. Therefore, the negative selection stage of the human immune system is important to assure that the generated antibodies do not to attack self cells.

나. Negative Selection Algorithm

Even though the clear role of negative selection in a human immune system is to eliminate harmful antibodies, it shows some other important features, which can help us to devise a more effective anomaly detection algorithm. Conventional anomaly detection algorithms generally establish the normal behaviour of a monitored system and spot significant deviations from the established normal characteristics. The antigen detection mechanism by antibodies follows this conventional anomaly detection algorithm in a way, but it shows some other strengths over this conventional algorithm.

Forrest et al (1994), (Forrest, Hofmeyr, and Somayaji, 1997) proposed and used a negative selection algorithm for various anomaly detection problems. This algorithm consisted of three phases: defining self, generating detectors and monitoring the occurrence of anomalies. In the first phase, it defines self in the same way that other anomaly detection approaches establish the normal behaviour patterns of a monitored system. In other words, it regards the profiled normal patterns as self patterns. In the second phase, it generates a number of random patterns that are compared to each self pattern defined in the first phase. If any randomly generated pattern matches a self pattern, this pattern fails to become a detector and thus it is removed. Otherwise, it becomes a detector pattern and monitors subsequent profiled patterns of the monitored system. During the monitoring stage, if a detector pattern matches any newly profiled pattern, it is then considered that new anomaly must have occurred in the monitored system.

This negative selection algorithm has been successfully applied to detect computer viruses (Forrest 1994), tool breakage detection and time-series anomaly detection (Dasgupta, 1998). Besides these practical results, Dhaeseleer et al. (1997) showed several advantages of negative selection as a novel distributed anomaly detection approach. One of the formidable features is that this novel approach does not define specific anomalies to be detected and thus it does not require the prior knowledge of anomalies. This feature allows it to be able to detect previously unseen anomalies.

In addition, the detection is distributed and local. This trait originates from the aggregation of distributed and independent detector detection. That is to say, an individual detector contains only a subset of the patterns needed to describe all existing anomalies, and it monitors only small parts of the system. Therefore, each detector recognises only the anomalies of the small section of the system that it monitors, and the overall abnormal status is diagnosed by the collection of independent detection results. Moreover, this distributed detection by local detectors provides robustness within the system. The anomaly detection problem for computer security such as computer virus detection and intrusion detection especially requires robustness of the detection algorithm. It has to be robust enough to withstand the attack and any system faults. The multiple detection points by independent detectors and the uniqueness of each detector allow it to be robust (Kim and Bentley, 1999a), (Forrest, Hofmeyr, and Somayaji, 1997).

However, the current negative selection algorithms show several drawbacks. The most significant problem is the excessive computational time caused by the random-generation approach to building valid detectors. This results in the exponential growth of computational effort with the size of self patterns (Dhaeseleer et al., 1997). Moreover, it is very difficult to know whether the number of generated detectors is large enough that can satisfy the acceptable detection failure probability. Dhaeseleer derived a formula presenting an appropriate number of detectors when an acceptable failure probability is given and claimed that the derived formula allows the negative selection algorithm to tune its detection accuracy against the cost of generating and storing detectors. However, this work has been accomplished under some unrealistic assumptions: it does not take into account false positive error and independence between self patterns. Furthermore, he only considered binary patterns and a simple r-contiguous bit matching rule. Nevertheless, it is not easy to estimate the appropriate number of detectors when the negative selection algorithm employs numerical patterns and a more sophisticated matching rule. Therefore, this difficulty may force the negative selection algorithm to adopt an arbitrary number of detectors and this may cause an unexpected low detection accuracy or the inefficient computation by generating more than sufficient number of detectors.

2. Negative Selection Algorithm for Network Intrusion Detection

Among the three evolutionary stages comprising the artificial immune system, during the negative selection stage, the system generates diverse pre-detector patterns and selects mature detector patterns by eliminating false pre-detector patterns by binding them to self patterns (Kim and Bentley, 1999b).

To apply the negative selection algorithm, firstly, we need to generate pre-detectors and this requires the creation of a gene library containing various genes. For the human immune system, the immature antibodies are generated via the gene expression process, in which the gene segments of different gene libraries are randomly selected and rearranged in a random order. From this process, the genes of the gene libraries contain the genetic information that determines the specific structure of antibody binding area, which will be the complementary structure of existing antigen binding area. These genes are usually inherited from ancestors genes. To be more precise, the genes of the gene library of the human immune system initially have some knowledge about the antigens that had attempted to attack ancestors body. Returning to our problem, the genes of the initial gene library of the artificial immune system, which will be the genes of pre-detectors, can be the selected fields of profiles to describe anomalous network traffic patterns. The initial genes might be set by the values of these fields that are observed when a previously known network intrusion is simulated. However, the simulation of network intrusion can be a difficult task if network administrators and users of the monitored network are not co-operative. For this reason, we employ Forrest et als negative selection algorithm (Forrest et al, 1994) to generate pre-detectors, which does not initially require any network intrusion simulation.

As described in section 2. this algorithm consists of three stages: defining self, generating detectors and monitoring the occurrence of anomalies. In the first step, we define self by building the profile of normal network activities. After understanding the detailed mechanisms of network protocol and their security holes, we can define the fields of profiles. The details of these fields are described in the next section. In general, the fields of the created profiles represent the normal activities of TCP/IP protocol for each single connection. In the second step, the negative selection algorithm randomly generates the pre-detectors, whose fields are the same as those of self profiles but the values of these fields are randomly generated. The generated field values of these pre-detectors are compared to those in the self profiles. If the values of the common fields of both 'self' and 'pre-detector' are similar enough, this pre-detector is removed. The scheme to measure this similarity is discussed in section 3.4. The surviving pre-detectors become detectors which contain some specific values of originally defined fields of the self profile. In the third step, we continuously generate the profiles of current network activities in the same way and compare their field values with the detectors field values. If the values of the same fields of both any new self and a detector are similar enough, this self pattern is regarded as the signature of network intrusion.

3. Network Traffic Data for Negative Selection

In this section, the details about network traffic data used for this work are described.

가. Data Gathering

The data chosen for this research is available at http://iris.cs.uml.edu:8080/network.html. This is a set of *tcpdump* data and was collected for a part of an Information Exploration Shootout, which is a project providing several datasets publicly available for exploration and discovery and collecting the results of participants. The network packet capturing tool, *tcpdump*, was executed on the single gateway that connects an intra-LAN to external networks. It captured TCP packet headers that passed between the intra-LAN and external networks as well as within the intra-LAN. Five different data sets were generated. The TCP packet headers of the first set were collected when no intrusion occurs and the other four sets were collected when four different intrusions were simulated. These intrusions are: *IP spoofing attack*, *guessing rlogin* or *ftp passwords, scanning attack* and *network hopping* *attack*. The details of attack signatures and attack points of the four different attacks are not available. This data originally had the fields of *tcpdump* format such as time stamp, source IP address, source port, destination IP address, destination port and etc.

나. Data Profiling

Since *tcpdump* is not designed for security purpose, its primitive fields are not enough to build a meaningful profile. Consequently, the first stage of our data profiling program is to extract more meaningful fields, which can distinguish normal and abnormal. Many researchers have identified the security holes of TCP protocols (Porras and Valdes, 1998) and so the fields used by our profiles are selected based on the extensive study of this research. They are usually defined to describe the activities of each single connection.

The automated profile program was developed to extract the connection level information from TCP raw packets. The TCP packet headers in the original file were collected according to chronological order. These original data were dumped into MS SQL-Server DBMS and the automated profile program was implemented in JAVA using JDBC accessing SQL-Server.

○ Profile Fields

Each connection is established between a source port executing on a source host and a destination port operating on a destination host. For TCP protocol, each time the source port process of a source host intends to communicate with the destination port process of a destination port, it establishes connection between them. For each TCP connection, the following fields are extracted:

- Connection identifier: each connection is defined by four fields, initiator address, initiator port, receiver address and receiver port. Thus, these four fields are included in the profile first in order to identify each connection.
- Known port vulnerabilities: many network intrusions attack using various types of port vulnerabilities. There are fields to indicate whether initiator port or receiver port potentially hold these known vulnerabilities.
- 3-way handshaking: TCP protocol uses 3-way handshaking for a reliable

communication. When some network intrusions attack, they often violate the 3-way handshaking rule. Thus, there are fields to check the occurrences of 3-way handshaking errors.

- Traffic intensity: network activities can be observed by measuring the intensity over one connection. For example, number of packets and number of kilobytes for one specific connection can describe the normal network activity of that connection.

So, in total, self profile fields have four types of 35 different fields.

○ Profiling Categories

Even though the network profile fields were extracted to describe a single connection activity, the data used in this research is too limited to apply this initial profile. The limit is that the data was collected for a quite short time, around 15~20 minutes. During this brief period, most different connections were established only once. An insufficient quantity of data was collected to build different connection profiles. Therefore, it is necessary to group different connections into several *meaningful* categories until each category can have a *sufficient number* of connections to build a profile. Consequently, a total number of connections for each potential profile category were counted.

First of all, the data was categorized into two different groups: inter-connection and intra-connection. Inter-connection is the group of connections that were established between internal hosts and external hosts, and intra-connection is the group of connections that were established between internal hosts. Furthermore, to preserve anonymity, all internal hosts have a single fake address 2 and any extra information about external hosts and network topology is not provided. Therefore, the profiles according to specific hosts are insufficient. Instead, in this research, only the profiles of specific ports on any hosts are considered.

According to various possible categories, the established connection number of each profile was counted. From each case, apart from a profile class that has more than 100 connections, other profile classes were again grouped into other different classes until each class has more than 100 connections. Finally, 13 different self profiles were built. Their class names and the number of established connections are shown in table 14.

Inter-connection	
Class	Number of Connection
{(2, *), (*, 80)}	5292
{(2, *), (*, 53)}	919
{(2, *), (*, 113)}	255
{(2, *), (*, 25)}	192
{(2, *), (*, well-known)}	187
{(2, *), (*, not well-known)}	756
{(2, 53), (*, *)}	940
{(2, 25), (*, *)}	352
{(2, 113), (*, *)}	145
{(2, well-known), (*, *)}	114
{(2, not well-known), (*, *)}	6050
Intra-Connection	
{(2, *), (2, well-known)}	190
{(2, *), (2, not well-known)}	189

표 44 : Self Profiles

In Table-44, the class column of inter-connection is shown as: {(a,b),(c,d)}, where a is an internal host, b is a internal port number, c is a external host address and d is an external port number. Hence, the connection is established between (a,b) and (c,d). For the class column of intra-connection, a is an internal host address, b is an internal port number, c is an internal host address and d is an internal, port number. Table-44 indicates any host address and any port number. In addition, well-known shows the ports in the range 0 to 1023 are trusted ports. These ports are restricted to the superuser: a program must be running as root to listen to a connection. The port numbers of commonly used IP services, such as *ftp, telnet, http*, are fixed and belong to this range. But, many common network services employ an authentication procedure and intruders often use them to sniff passwords. It is worthwhile to monitor these ports separately from the other ports. Therefore, if the number of connections for any profile category, which is based on a specific port on any hosts, is not sufficient, these categories are regrouped into two new classes, a well-known port and a not well-known port.

4. Implementation

This section describes the detailed implementation of the negative selection algorithm that is proposed in this work. It introduces the genotype and phenotype representation, the genetic operators and finally the fitness functions which are based on the similarity between a detector pattern and a self pattern.

가. Genotypes and Phenotypes

In this section, the details about handling continuous values suitable for genotype encoding, genotype representations and mapping between genotypes and phenotypes are described.

○ Discretisation

As seen above, each network activity profile has 35 fields. From these 35 fields, the values of 28 fields are continuous and the values of the other 7 fields are discrete. Specifically, the continuous values of 28 fields show a wide range of values. In order to handle this various and broad range of values, a simple discretisation algorithm is required.

There are many discretisation algorithms available (Freitas, 1997). Most of these algorithms require long processing times. To make a system to report the occurrence of intrusions immediately, a simple discretisation algorithm that requires less computing time is used in this work. This algorithm consists of two steps. In the first step, an overall range of real values for each field is sorted. In the second step, according to a given total cluster number, which is a variable, the number of records for each cluster is uniformly determined. In other words, the lower bound and higher bound of each cluster are determined by ensuring that each cluster contains the same number of record. We intentionally choose the simplest discretisation algorithm here in order to use the formula which controls the appropriate number of detectors when a false negative error is given **(Dhaeseleer et al., 1997).**

○ **Genotype Encoding**

Genotypes consist of 35 genes where each gene represents each field of a detector. As described above, the profile built for this work has 35 fields and this number determines the total number of corresponding genes in the detectors. Each gene indicates cluster number and it has an alphabet of cardinality 10 with values from 0 to 9. For example, the gene g1 indicates the first field of a profile, the number of packets sent by an initiator. This field has integer values and so these values were discretised into n clusters based on a predefined cluster number. When the discretisation is performed, a cluster table is generated. It contains intervals of clusters indexed by ascending sequential numbers. Thus, each field in a self profile has its corresponding cluster table and the corresponding gene represents the cluster number stored in the cluster table. Since the total number of clusters is given before starting a negative selection algorithm, a whole chromosome still consists of a fixed length of genes, each having an alphabet of cardinality 10. Finally, for a nominal type of field, such as the 'well-known source port', which indicates whether a given source port is well known or not, a cluster interval of a cluster table is defined simply by the meaning of each group.

○ Measure Similarity

While generation of detectors and self-profiles and application of genetic operators are performed at the genotype level, measurement the similarity between a selected detector pattern and self pattern is operated at the phenotype level. This is another difference between most work using a negative selection algorithm (Forrest, Hofmeyr, and Somayaji, 1997), (Dasgupta, 1998). Such work usually performed this evaluation procedure on a genotype level using simple r-continuous bit matching rule.

In this work, in order to measure the similarity, genotypes for each generation are mapped onto phenotypes. The new similarity measurement is as follows:

- The genotype of each detector is mapped onto phenotype according to instructions in cluster tables and offset tables.
- The degree of similarity between the phenotype of the detector and the selected self is measured.

Phenotypes mapped from evolved genotypes are represented in a form of detector patterns. A field of a detector phenotype is represented by an interval having a lower bound and a higher bound while a field of a self phenotype is described by one specific value. Hence, the first step of measuring the similarity checks whether a value of each field of a self pattern belongs to a corresponding interval of a detector phenotype. When any value of a self pattern field is not included in its corresponding interval of a detector phenotype, these two fields are not matched. In this case, the degree of similarity is measured by the distance from the value of a self pattern field to the closer value out of the lower bound and higher bound. These two bounds comprise an interval of the corresponding field value of the detector pattern. After assigning this distance as a similarity score of an individual field of the detector pattern, a total similarity score of a given detector pattern is calculated by summing all these individual similarity scores. It should be noted that before summing them up, each score must be normalised.

5. Experiments : Investigate the Feasibility of Independent Negative Selection

○ Experiment Design

The problems of the negative selection algorithm are exemplified through a series of experiments that apply it on the first data set. The negative selection algorithm used in these experiments mainly followed the implementation details which are used in (Forrest etal., 1994). However, there are several things that are different from Forrests implementation details and those are explained in the previous section 3.4 Implementation. Some implementation details have been kept the same as Forrests (Forrest et al., 1994). For example, the same matching function, the r-continuous matching function for measuring the similarity is used. Its matching threshold is defined as 9. In order to define this number, the formula to approximate the appropriate number of detectors when a false negative error is fixed (Dhaeseleer et al., 1997) , (Forrest et al, 1994) is used.

It was shown that the longer matching threshold drives the creation of more general detectors, but it also causes a larger number of random detector generation trials, which need to avoid the matching a self profile (Dhaeseleer et al., 1997), (Forrest et al, 1994). Thus, we can derive an approximate appropriate matching threshold number by varying the expected false negative error and random detector generation trial number. Even though this formula is clearly useful to predict the appropriate number of detectors and its generation number, its predicted number showed how infeasible this approach is for applying it on a more complicated search space. For instance, when the expected false negative error rate is fixed as 20%, its predicted the detector generation trial number is 51 and the appropriate number of generated detectors is 21935 for the matching threshold is 3. Similarly, when we define the matching threshold is 4, it predicted 535 for the former and 955 for the latter. None of these cases seem to provide any feasible test case in terms of computing time. In addition, it was observed that when we fixed the matching threshold number as four and ran the system, the system could not manage to generate any single valid detector after one day. Thus, we generated valid detectors by setting the matching threshold number that allowed a system to generate a valid detector in a reasonable time.

○ **Experiment Result**

It was observed that the average time of successful detector generation took about 70sec CPU time and the average number of trails to generate a valid detector was 2.6 when a matching threshold was nine. Even though this number gave reasonable computing time to generate a valid detector set, very poor detection accuracy by generated detectors was shown. The maximum 1000 valid detectors were generated and the detection accuracy was measured per every 100 detectors. The observed detection accuracy was less than 20% for four different intrusion data sets and one artificially generated random test set. This result was gained as the average of five runs.

In contrast to the promising results shown in Hofmeyrs negative selection algorithm for network intrusion detection (Hofmeyr, 1999) , the experiment result of this research raises doubt whether this algorithm should be used for network intrusion detection. These contradictory findings can be explained by the fact that Hofmeyrs encouraging result originated from the adoption of limited profile features which a negative selection algorithm can handle, while the experiment of this research used the more complicated but more realistic profile features that a negative selection algorithm struggles to solve. More importantly, Forrest

(Forrest et al, 1994), (Somayaji et al., 1997) and Hofmeyer(1999) view that the network intrusion detection of artificial immune system is achieved mainly by the sole function of negative selection stage than the co-ordination of three different evolutionary stages However, Hofmeyer(1999) attempted to adopt the notion of clonal selection for his network-based IDS, but his system did not employ the full functionality of that stage such as clonning detectors when they detect intrusions. This is somewhat different from our view.

Consequently, the initial results of our experiments motivated us to re-define the real role of negative selection stage within an overall network-based IDS and design a more applicable negative selection algorithm, which following a newly defined role. As much of the other immunology literature addressed (Tizard, 1995), the antigen detection powers of human antibodies rise from the evolution of antibodies via a clonal selection stage. While Forrest et als negative selection algorithm allows it to be an invaluable anomaly detector, its infeasibility is also caused from allocating a rather overambitious task to it. To be more precise, the real job of a negative selection stage should be restricted to tackle a modest task but reflecting the closer role of negative selection of human immune system. That is simply filtering the harmful antibodies rather than generating competent ones.

제4절 The Clonal Selection of an Artificial Immune System

Even though new antibodies surviving negative selection are assured to be self-tolerant, their efficacy to detect antigens is unknown when they are released from the bone marrow and the thymus. This is because new antibodies are randomly generated and they are verified only not to be self. They might hold non-self patterns but not antigen patterns. In order to exclude these ineffectual detectors, the human immune system adopts the evolution of antibodies towards the existing antigen patterns (Tizard, 1995). During this evolution process, the human immune system uses its own unique niching strategy to maintain generality and diversity of antibodies as one part of clonal selection process (Forrest et al, 1993).

1. Clonal Selection of the Human Immune System

B-cells and T-cells, which are antibody secreting cells, employ various gene shuffling mechanisms and somatic mutation in their development stage as their strategy for maintaining the diversity of antibodies. Besides these mechanisms, human immune systems adopt different strategies. As one species evolves via natural selection, human immune systems also evolve through a process called *clonal selection.* The genes, which determine the specificity of antibodies, continuously evolve toward having the capacity to detect more prevalent pathogens at any moment and reproduce new lymphocytes with high affinity for those specific pathogens (Playfair, 1996), (Roitt and Brostoff, 1998), (Tizard, 1995) .

When B-cells are developed in bone marrow, only a limited energy source is given to them. They are active immediately after release from bone marrow, but they are rapidly exhausted and die. However, they do not simply disappear if they are activated by binding to specific antigens. When B-cells are directly or indirectly activated by antigens, they are divided and differentiated into a number of clones of antibody secreting cells, plasma cells, before they are exhausted. Plasma cells have the same antigen-binding properties as the receptors of parent B-cells or they can have mutated antigen-binding properties. As B-cells bind more to specific antigens, they have more chances to be selected for cloning. Similarly, as they bind less to specific antigens, they have fewer chances to be selected to clone and eventually tend to decrease. According to the different population of existing specific antigens, only the fittest antibodies survive.

Furthermore, when B-cells are activated by antigens, they produce memory cells for the reoccurrence of same antigens. The life of B-cell and its clone, the plasma cell is relatively short. However, some of the B-cell clones survive as memory cells. Some of the memory cells are exposed antigens and differentiated into plasma cells without undergoing somatic mutation and other memory cells undergo somatic mutation to be differentiated into plasma cells. Particularly the plasma cells generated without somatic mutation allow the secondary response of immune systems. When new pathogens are detected by B-cells, they generally take some time. We call this primary response. Compared to the primary response, the secondary responses by memory cells are very fast and efficient. Moreover, the memory cell provides an associate memory property (Dhaeseler, 1997). The new antigens have a structure that is not the same but is similar to the structure of previously detected antigens, and can be detected by memory cells. This is because the binding of antibody and antigen is approximate. For example, when a body is infected by cowpox, the immune system takes time to detect and eliminate it. But if somebody is infected by smallpox after being infected by cowpox, he/she is rapidly cured by the secondary response of immune system. This is because cowpox and smallpox are similar enough to induce secondary response by memory cells.

2. The Clonal Selection Algorithm

As discussed in the previous section, the clonal selection of a human immune system shows several significant mechanisms to maintain diversity and generality of antibodies. Among those, for the artificial immune system, we focus only on its natural selection mechanism in order to supplement for the efficiency drawback of negative selection algorithm. By using the evolution process of clonal selection, the computational time of negative selection due to random generation can be reduced. In addition, the problem of tuning the appropriate number of detectors may be solved by multimodal convergence feature of a niching strategy.

Forrest et al (1993) presented the niching strategy of their artificial immune system which follows the analogy of the human immune systems. They explored whether it is able to i) detect common patterns of randomly presented antigens and ii) to discern and maintain the diverse antigen population. In their model, they created one population of antibodies and one population of antigens randomly. They used the GA to evolve the antibody population under a constant antigen population. This algorithm was devised by observing the following unique features of the human immune systems:

① From a specific antibody's point of view, antigens are usually encountered

sequentially.

② From an specific antigen's point of view, it responds with only a subset of

antibody secreting cells.

③ There is competition among antibody secreting cells which bind given antigens.

④ The antibody secreting cells are evolved by somatic mutation.

Conforming to the niching strategy of the human immune system which is brought by the clonal selection, for each generation, their modified GA selects an arbitrary size of random sample from the antibody population and a single random antigen from the antigen population. After each antibody in the sample is matched against a selected antigen, the fitness score of only one antibody showing the highest match score is increased while the fitness scores of the others remain the same. In summary, the observed properties of the human immune systems are turned into the following clonal selection algorithm:

① A single antigen is randomly chosen.

② A fixed size of the antibody population sample is chosen at random.

③ Each antibody in the sample is compared with the randomly selected antigen.

④ The antibody in the sample which showed the highest match score has added the

match score to its fitness value. The fitness values of other antibodies remain the

same.

⑤ This process is repeated for many antigens.

Using this algorithm, Forrest et al (1993) showed antibodies evolved to be generalists that match to most antigens to some extent. Their analysis of this result showed that antibodies evolved towards finding common schema that is shared among many antigens. Through the various experiments, they observed that this algorithm could sustain multiple inconsistent antibody patterns, which appear as the multiple peaks at a search space, and the similarity among antigens does not affect this capability. Moreover, they compared this niching strategy of the artificial immune system with the fitness sharing algorithm (Smith, Forrest, and Perelson, 1993). From this comparison, they reported that as the result of antibody sampling mechanism, the niching strategy of the artificial immune system controls its generality via the antibody sample size. To be more precise, when the sample size decreases, the selective pressures are moved towards generating a population of more general antibodies.

Even though the negative selection algorithm provides several strengths for network intrusion detection, it is necessary to resolve the excessive computational time caused from the random generation approach. Dhaeseleer (1997) introduced more efficient detector generation algorithms: a linear-time algorithm and a greedy algorithm. The basic idea is to provide an efficient method to enumerate all candidate detectors and thus allowing the negative selection algorithm to select valid detectors from this complete candidate detector set. However, this algorithm can be used only for a binary immune system using a simple r-continuous-bits matching rule. This is because they enumerate all possible valid detectors by counting the recurrence of all the potential r-continuous-bit binary strings unmatching self strings. Dhaeseleer also suggested the use of a non-binary alphabet immune system as an important future investigation because it is more natural in many cases. Furthermore, as analysed in the previous chapter, even when this formula can be applied (the case which uses a binary encoding and r-continuous matching function), it turned out infeasible when it is applied on real network traffic data due to its excessive quantity.

As the result, instead of using one of these algorithms, the negative selection algorithm for network intrusion detection introduced in this work should be replace by the niching strategy of Smith, Forrest, and Perelson's (1993) artificial immune system to build a valid detector set.

3. The Clonal Selection Algorithm for Network Intrusion Detection

In the first phase, the clonal selection algorithm build self/non-self profiles. In this research, the raw network traffic packets were gathered and these packets were parsed and built into self/non-self profiles according to its connection label.

These profiles are equipped with previously identified fields, which can distinguish normal and abnormal network activities. Then, the profiles are encoded in an appropriate data representation. In the second phase, when all the self/non-self profiles are encoded, the clonal selection algorithm starts generating detectors. We slightly modify Forrest et al's clonal selection algorithm() by adding the negative selection as one operator to it. The second phase of this algorithm for generating a detector set can be summarized as follows:

For each connection profile and its corresponding detector set:

- *D* detector patterns are generated at random and their fitness values are initialised with zeroes.
- A sample of *N* detector patterns is randomly selected from the generated *D* detector patterns.
- A single intrusion pattern is randomly selected from the non-self profile.
- Each detector in the sample is compared to all the self patterns from the self profile and the degree of similarity is measured. If any detector matches any self pattern completely, this detector is replaced by a new detector with ranom genes.
- Each detector in the sample is compared to the selected intrusion pattern and the degree of similarity is measured.
- The fitness value of the single detector in the sample that shows the largest similarity is increased. The fitness values of other detectors remain the same.
- The processes 2-5 are repeated (for typically three times the number of antibodies (Smith, Forrest, and Perelson, 1993)).

- P_b% detector patterns are selected as parents and genetic operators such as crossover, mutation are applied to generate new detectors.

- P_w% detector patterns are deleted to make space for children.

- A new detector population is created by including the selected parent detectors and the offspring detectors generated in 8

- Processes 2 - 9 are repeated until the fitness values cease to change.

After finishing the second phase by performing above, the clonal selection algorithm builds new self profiles by parsing newly captured network packets. In the third phase, the detector patterns in each detector set are compared to the patterns in each corresponding new self profile. If any detector pattern matches the new self pattern, the algorithm generates an alarm signal.

This niching strategy controls the generality of each detector according to a detector sample size. For practical reasons, we expect this algorithm to create more general detectors so that each detector can match more than one intrusion. This means that even though each detector cannot bind to one intrusion exactly, it can match a number of intrusions to some degree. This approach is more likely to be suitable for network intrusion detection. This is because, as we can clearly see in the next section, the length of each self chromosome used in this work and the search space which these self chromosomes form is much larger and complex than the search spaces handled in most of work using a simple negative selection algorithm (Dhaeseleer, 1997). Furthermore, we expect the computation time of the clonal selection to be less due to using evolution rather than random search. Finally, the appropriate number of detectors will also be naturally determined based on the multimodal convergence of evolution process.

4. Network Traffic Data for Clonal Selection

When the negative selection algorithm was developed, the first data set. Even though this data allows us to test the first prototype of the system, the amount of data was not enough to describe the connection-oriented network behaviour.

This is because it was collected only for a very short period (which is approximately 15 minutes.) This led us to find a better quantity of data set. The second data set was provided by the DARPA Intrusion Detection Evaluation Program (http://www.ll.mit.edu/IST/ideval/index.html). This program was set up and managed by MIT Lincoln lab to survey and evaluate the state of art in intrusion detection research. It gathered about 4 gigabytes of compressed network packets of 7 weeks and simulated a wide range of various network intrusions. These intrusions are categorised into four groups: Denial-of-service intrusions, unauthorised access from a remote machine, unauthorised access to local superuser privileges by a local unprivileged user and surveillance and probing attacks. In other words, for each intrusion class, a number of different intrusions but using a similar attack scenario are simulated. Compared to the first data set, this data set provides more extensive types of intrusions.

The data from these two data sets originally had the fields of network packet capturing tools format such as time stamp, source IP address, source port, destination IP address, destination port and etc. However, the primitive fields of captured network packets are not enough to build a meaningful profile. Consequently, it is essential to build a data-profiling program to extract more meaningful fields, which can distinguish normal and abnormal. Many researchers have identified the security holes of TCP protocols (Porras and Valdes, 1998) and so the fields used by our profiles are selected based on the extensive study of this research. They are usually defined to describe the activities of each single connection.
The automated profile program was developed to extract the connection level information from TCP raw packets and it was used to elicit the meaningful fields of the first data set. For the second data set, Lee(1999) provided the pre-processed data set which extract the meaningful fields from DARPAs original data.

For each TCP connection, the following fields are extracted for both two data sets:

- Connection identifier: each connection is defined by four fields, initiator address, initiator port, receiver address and receiver port. Thus, these four fields are included in the profile first in order to identify each connection.

- Known port vulnerabilities: many network intrusions attack using various types of port vulnerabilities. There are fields to indicate whether an initiator port or a receiver port potentially holds these known vulnerabilities.
- 3-way handshaking: TCP protocol uses 3-way handshaking for a reliable communication. When some network intrusions attack, they often violate the 3-way handshaking rule. Thus, there are fields to check the occurrences of 3-way handshaking errors.
- Traffic intensity: network activities can be observed by measuring the intensity over one connection. For example, number of packets and number of kilobytes for one specific connection can describe the normal network activity of that connection.
- Connection Content: these fields describe the activities of network connection user. One example is the number of failed logins or whether a *su* command is attempted and succeeded etc.

Thus, in total, network profile fields have 35 different ones for the first data set and 41 different fields for the second data set. While the first set includes only the first four types of fields, the second set adds the connection content type of fields. The details of these fields are introduced in Appendix.

5. Implementation

This section describes the detailed implementation of the clonal selection algorithm that is proposed in this work. This stage is developed in order to build an initial gene library. It introduces the genotype and phenotype representation, the genetic operators and finally the fitness functions which are based on the similarity between a detector pattern and a self pattern.

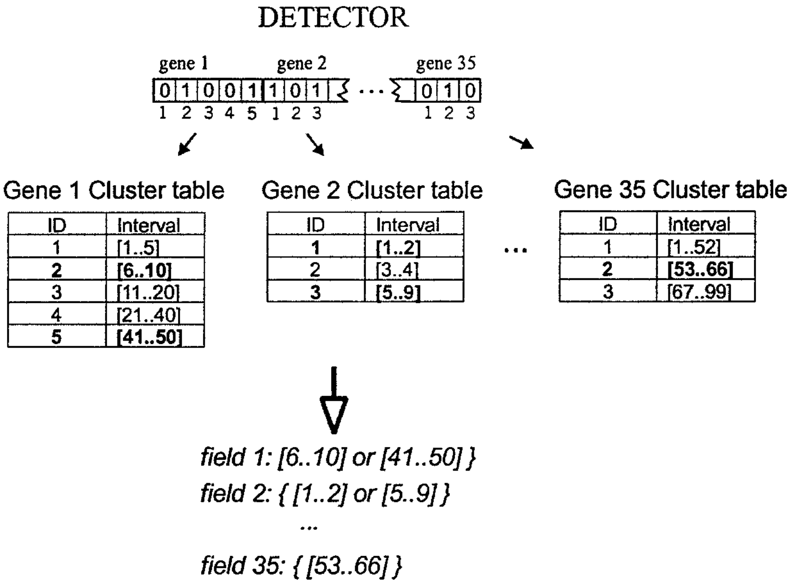
가. Genotypes and Phenotypes

The clonal selection algorithm employs the evolution mechanism of the classification rules, which classify non-self from self. One of natural ways of expressing classification rules is as a set of disjunctive normal form (DNF) rules. The *if-part* of each rule is a conjunction of one or more conditions to be tested and the *then* side of the rule describes the class label assigned to the rule. In the context of this research, the generated single detector will have a conjunctive rule as its phenotype. Therefore, the non-self patterns that can be detected by generated detectors will be a disjunction of these conjunctive rules.

There is some recent work which employed especially GP for the evolution of classification rules including our previous work financial fraud detection (Bentley, 2000a), (Bentley, 2000b). It takes advantage of the flexibility of GP that can adopt various functions sets, such as negation, larger-than, etc. This flexibility allows its phenotypes, classification rules, to express more complex conditions. However, this research trades the strong expressive power of GP for the simplicity of standard GA that can use bit string representation as its genotype. This is because one significant aspect of this research is the analysis of the pitfalls that the previously proposed AIS algorithms might have. In order to do so, we follow the closest genotype representations to theirs, but practically powerful and economic enough to express given phenotypes. The genotype and the phenotype representations which are used in this work is not claimed as the best way to do so. However, it is suitable for the purpose of this research.

○ Genotype Encoding and Phenotype Interpretation

Genotypes consist of 41 genes where each gene represents each feature of a detector. The profile built for this work has 41 features and this number determines the total number of corresponding genes in the detectors. Each gene comprises existing feature values. For instance, in the case of Protocol_type feature, its valid feature values are tcp, udp and imcp. Thus, this feature has only three possible values and leads the corresponding gene to have only three nucleotides. As seen in figure 16, each nucleotide is a binary bit whose value one represents to include the corresponding feature value in the condition part of a classification rule and whose value zero indicates vice versa. This kind of genotype representation allows a single feature of each detector rule to have more than one value, which can be combined by OR operator by assigning a separate bit to each existing feature value. In addition, the valid genes of each detector rule are combined by AND operator. Thus, the genotype in figure 16 can be interpreted as "IF (field1 belongs to ([6..10] or [41..50]) and field2 belongs to ([1..2] or [5..9]) and .... and field5 belongs to [53..66])THEN it is intrusion". It should be noted that in the case that all the existing nucleotides of a single gene have identical values, which would be all ones or all zeroes. It will result in the exclusion of this feature from the condition part of a given classification rule. This is because both cases (the former means any value of this gene and the latter implies none of valid gene values) can be interpreted that any value of this gene is irrelevant to the class decision. Furthermore, for the case that all the



ID	Interval
1	[1..5]
2	**[6..10]**
3	[11..20]
4	[21..40]
5	**[41..50]**

ID	Interval
1	**[1..2]**
2	[3..4]
3	**[5..9]**

ID	Interval
1	[1..52]
2	**[53..66]**
3	[67..99]

field 1: [6..10] or [41..50] }
field 2: { [1..2] or [5..9] }
...
field 35: { [53..66] }

그림 24 : **Detector Representation for Clonal Selection**

This kind of genotype representation is not novel. This encoding scheme has been very popular and its descriptive power has been proved to be sufficient enough to handle practical rule evolution problem. It was proposed by De Jong(De Jong et al., 1993) for this work to use GA for concept learning, which also aims the evolution of attribute-based classification rules. This approach can be compared to another classic genotype encoding schemes for the same domain As being mentioned GP as an alternative approach before, there are some other approached to express classification rules for their evolution. For instance, real-value encoding, fuzzy rule encoding, etc. can be more powerful and flexible encoding scheme. However, for the same reason described earlier, the direct comparison of genotype encoding scheme, which is employed in this research, to these method is not conducted. We narrow our attention to the binary encoding methods that are used in Michigan approach for the classification rule evolution.

This different method is that assigning an appropriate number of bits which sufficiently cover the valid range of gene values (Goldberg, 1989). For instance, if the feature 1 has three values and it requires two bits, which is large enough to indicate three different cases, 01, 10, 11. For this method, the required number of bits to encode each gene is 2** when ** is a number of existing gene values. The advantage of De Jongs genotype encoding method over this method is that it allows a disjunction of feature values for each feature in a single detector rule while the other method represents only one valid value for each feature per detector rule. By doing so, one single detector rule has more expressive power and it becomes more general, which means that it can match more intrusion patterns. As a consequence, the number of detector rules, which is required to detect a given number of intrusion patterns, will become smaller. The lightweight detectors are required for the network intrusion detection context and this strength of De Jongs method certainly advocates the lightweight feature of generated detector.

This genotype representation is a popular approach to the evolution of classification rules, but the somewhat different method that was used by other AIS algorithms for concept pattern recognition. Forrest et al (1994) and Potter (1997) developed the AIS for concept pattern learning and they used the different genotype encoding method which is introduced above. Even though they used a rather restricted genotype description, they managed to endow the lightweight feature to generated detectors by using a match threshold concept, which looks closer to the approximated binding procedure of human immune systems. The brief description of this method is that each antibody and antigen are represented by a fixed length of binary strings and when a predefined number of bits are complementary, these two string are considered to match each other. A given single antibody string can match more than one antigen string as long as the number of their complementary bits is above a specified threshold. In this case, a matching threshold value affects the degree of antibody generality, but it is difficult to determine its appropriate value. To overcome this problem, Potter lets this value to evolve together when an antibody evolves by his clonal selection algorithm.

Even though the method which maintains the lightweight property of detectors employed by Forrest et al(1994) and Potter(1997) looks closer to human immune systems, the expressive power of each detector rule is weaker than De Jongs encoding power. For example, a single detector encoded by De Jong's method can be represented by (number) detectors encoded by Forrest et al and Potters encoding schemes. Thus, even though they provide the generality of generated detector via the threshold match method, this work advocates the De Jongs genotype encoding scheme to generate an even lighter detector. In summary, this method trades the lightweight feature of a generated detector for a larger search space This conclusion does not imply that a threshold matching function is generally poorer at showing its generality when it is compared to logical representation. Notably GPs unique tree structure employing various functions provides a strong expressive power and it used a matching threshold concept such as a sphere threshold or a linear threshold. However, we still prefer De Jongs approach since its phenotype, which is logical representation, more natural to read and thus it provides strong intelligibility of generated detector rules. The intelligibility is one of significant IDS features for a security officer to operate IDSs.. However, when the overall architecture of AIS for network intrusion detection developed in this thesis is recalled, our main concern to make a system light originates from a possible burden of network traffic by transferring a large number of detectors from a primary IDS to secondary IDSs. Therefore, the lightweight feature of each detector is considered as more significant than longer computation time to generate detectors due to larger search space. This problem can be resolved by employing the parallel arrays of primary IDSs and this suggestion was already made in chapter 2.

○ Discretisation

As seen in the previous section, each network activity profile has 41 fields. From these 41 fields, the values of 32 fields are continuous and the values of the other 9 fields are discrete. Specifically, the continuous values of 32 fields show a wide range of values. In order to encode this various and broad range of values in the genotypes that are described above, a discretisation algorithm is needed. For the negative selection, we simply defined each discretizsed cluster having contains the same number of records. Even though the simple discretisation algorithm provides the clusters, it is not certain whether each cluster is formed without serious information loss. As the result, the clonal selection in this work used an entropy-based discretisation algorithm (Witten and Frank, 2000). The basic idea of this algorithm is that splitting a numerical attribute value recursively until the new cluster intervals does not minimize its entropy. In other words, this algorithm seeks for the splitting points where makes the information gain largest. Here, the information gain is defined as the difference between the information value without the split and that with the split.

However, this kind of naive entropy-based discretisation algorithm is also hindered by the overfitting problem. Since the entropy measures the purity of data points, this value has the minimum when the data points of one split belong to the same class. However, this again leads the generated clusters reflecting only a training data set. Fayyad and Irani (1993) proposed the advanced entropy-based discretisation algorithm to alleviate the overfitting problem. Instead of searching for the largest information gain, it adopts the minimum description length(MDL) principle to stop recursive splitting. This principle suggests that the best solution will lie on the point which minimize the overall cost of the learned system. The cost of the system is defined as the degree of system complexity plus the amount of information required to specify the errors when the current system is applied. In other words, when the system complexity increases during a training period, it reflects a training data set more closely and thus the prediction error on the training data decreases. By stopping the system evolution earlier, the complexity of the system decreases, but it should pay more for informing incorrectly predicted cases. Thus, MDL priniciple proposes to seek for the balanced point between the high cost caused by bulding up the system reflecting a training set perfectly and the high cost brought from sending extra information to correct wrong predictions when the system is not complicated enough.

Fayyad and Irani's new stop criteria of recursive splitting is

$$Gain(A, T; S) < \frac{\log_2(N-1)}{N} + \frac{\Delta(A, T; S)}{N} \quad ---- \ (1),$$

where N *is* the number of examples in the set S, a feature A and partition boundary T and

$$Gain(A, T; S) = Ent(S) - E(A, T; S),$$

the class information entropy of the partition made by T is

$E(A, T; S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2)$, and *Ent(S)* is the entropy of S.

$\Delta(A, T; S) = \log_2(3^k - 2) - [k \cdot Ent(S) - k_1 \cdot Ent(S_1) - k_2 \cdot Ent(S_2)]$, and k_1 is the number of class labels represented in the set S_i. Here, the first component of (1) is the information needed to specify the splitting point and the second is a correction cost required to transmit which classes corresponding to the upper and the lower subintervals. Since this recursive partition of each numerical attribute value is completed by evaluating each splitting point independently using above criteria, some numerical attributes are clustered very finely whereas others will be partitioned coarsely.

나. Fitness Functions

While the generation of detectors by applying genetic operators is performed at the genotype level, the fitness value measurement among competing detector candidates is operated at the phenotype level. For effective network intrusion detection, two important objectives of AIS are regarded: accuracy, and generality. The *accuracy* of generated detector rules can be measured by considering false negative errors and false positive errors.. The human immune system uses the unique method to reduce false positive errors via negative selection algorithm. The clonal selection algorithm developed in this work also embed this negative selection in its fitness evaluation stage. (See section 4.3). The *generality* of is also achieved by its unique fitness evaluation method using antibody sampling. The intelligibility which was considered as one of significant golas to be achieved by the financial fraud detection is excluded from the objective function of clonal selection. Even though this feature will be the important feature, our system currently concentrates on equipping its accuracy and generality first. Thus, the intelligibility of this system still remains as the problem to be tackled and we can apply the similar multiobjective fitness functions which were used in the financial fraud detector. As a result, the clonal selection of the AIS developed in this work has a single fitness function which indicates the accuracy of generated detector rules.

Many other evolutionary rule learning algorithms usually employ fitness functions measuring the accuracy of generated detector rules by counting the number of examples correctly classified. and its modality is maintained by a nested search. However, the clonal selection algorithm used in this work maintains its generality via the emergent fitness sharing completed by the antigen sampling at the fitness evaluation stage. This emergent fitness sharing was achieved because it lets only the selected antibodies in a sample to compete with each other per generation. In this case, the antibodies within a hypersphere of a specific radius around a given antigen form a local competing group and the fittest one in this group becomes the generalist one to describe this local search area. Thus, the key point of emergent fitness sharing of clonal selection is the natural formation of competing groups based on the hypersphere with the given radius around the selected antigen. In other words, the multi-modals in an antigen search space are searched by grouping candidate antibodies, which are similar enough to evolve towards a same target antigen, into one competing group. Therefore, the clonal selection algorithm used in this work adopts the similarity measurement in order to boost its accuracy feature instead of counting correctly classified items. Furthermore, the distance measurement has the advantage which accelerate the evolution speed at an early stage. This is because the detectors are initially generated with random genes and many of them are still far from any of antigens. Thus, if the system uses the correctly detected antigens as its fitness function values, it can have zero values or very tiny values. This can result in making the evolution proceed very slowly at the early stage. However, since the similarity measurement informs to the system how far each detector resides from the current antigens, the evolution can progress towards the current antigens a lot faster even before any of existing detectors detect any antigen.

The initial approach to measure the similarity between a selected antigen and antibodies is applied on their genotype level. Thus, the antigen phenotype value should be converted into an appropriate cluster number which was used to define detector genotypes. Then, if the corresponding nucleotide of detector genotype gene is turned on(the bit value is 1), this pair matches. Whenever a given antigen gene and detector gene match, that detector gene is assigned a match distance 0. However, if the corresponding nucleotide of detector genotype gene is not turned on(the bit value is 0), it searches for the closest nucleotide turned on. Then, the number of bits between the closest detector nucleotide, which is turned on, corresponding to the antigen cluster number is calculated as the match distance of these two genes. The second step is counting the matched fields and measuring the distance between unmatched fields. This finally defines the final match score, M_{gb}, between an antigen, A_g, and a detector, A_d. That is

$$M_{gb}(A_g,A_d)=\sum_g\left(\frac{d_{mg}}{N_{cg}}\cdot\frac{1}{N_g}\right),$$

where d_{mg} is the match distance between the gth antigen gene and the gth detector gene, N_{cg} is the number of total clusters comprising the gth gene and N_g is the number of total genes. Here, the gene match distance for each gene pair is divided by the total number nucleotide of the given gene in order to gain the gene match distance relative to the defined gene length. In addition, the gene match distance is scaled in order to make its maximum score 1 by dividin it by the total number of genes.

This match score represents the similarity between a given antigen and a detector. When this score is zero, they match completely and have the highest similarity. In the contrast, when this score shows 1, their distance is the largest and so their similarity becomes the least. The match score of the detector is kept until the detector/antigen sampling and evaluation procedure finishes after the certain number of repeated work. Then, the match score stored in the detectors are compared and only the detector which has the highest similarity (= the least match score) is selected for updating its fitness values. If the selected detectors are more than one, the clonal selection chooses only one detector to break the ties. The chosen detector update its fitness score by adding (1-match score) and represents one with the highest fitness score as the more desirable one.

다. Genetic Operators

Child rules are generated using two genetic operators: crossover and mutation.

○ **Crossover**

The crossover used in this work is the classic single-point crossover of genetic algorithm. It finds two random points in the parent genotypes and splicing the genotypes at that point. These spcliced genotypes are crossed over, and that the binary numbers within the genes are also crossed. This operator is used to generate all new offspring detectors, i.e., crossover is applied with a probability of 100%.

○ **Mutation**

The clonal selection algorithm uses a various types of mutation operators. As the first one, it uses the conventional mutation operator to modify randomly the binary numbers in each gene by a single bit. The second mutation operator is the generalisation mutation. This mutation is designed to generate a more generalised offspring. To the new detector to be more general, this operator truns on one bit and results in having one more condition in a phenotype detector. For example, after applying the generalisation operator, a detector

(A1 = small) and (A2 = good) might create a new offspring

(A1 = small or medium) and (A2 = good).

To make the effect of this mutation the gradual change on the offspring, the clonal selection algorithm selects one nucleotied randomly and it is turned on only any of its adjacent bits is turned on. For the case when all nucleotides are turned off, any nucleotide can be truned on.

The next mutation operator applied is the specialisation mutation. On the contrast to the generalisation mtation, this mutation simply drops one bit and generates an offspring with one less condition. For instance, a detector

(A1 = small or medium) and (A2 = good) can generate a new offspring

(A1 = small) and (A2 = good) after being applied by the sepecialisation mutation.

This mutation is applied in a similar way to the generalization mutation's. That is that any randomly selected nucleotide is turned off only if any of its adjacent bits is turned off. In the case when all nucleotide are turned on, any nucleotide can be tuned off.

The forth mutation operator is the shift mutation. It shifts all the bits of a chosen gene to the right direction or the left direction. The shift direction is determined at random. Thus, a detector

(A1 = small) and (A2 = good) might have its offspring

(A1 = medium) and (A2 = good) because of the application of the shift mutation.

Finally, the delete mutation is also introduced in this system. It aims to delete an entire gene from a given detector. For example, a detector

(A1 = small) and (A2 = good) can have its offspring

(A1 = small) after being applied by the delete mutation.

This mutation works by turning on all the nucleotide and results in excluding the gene when a detector is mapped into phenotypes. This mutation is designed in order to operate as a feature selection operator. Since the candidate features included in a training set often include irrelevant ones and they can disrupt the search method immensely, the clonal selection algorithm uses the deletion mutation to filter out those genes.

○ **Others**

The clonal selection employs population overlapping, where the worst *n%* of the population are replaced by the new offspring generated from the best *m%*. Typically values of n = 80 and m = 40 seem to provide good results. The population size was normally between 100 and 200 individuals.

6. Conclusion

This research has investigated network-based IDSs and provided a set of general requirements for them by a careful examination of the literature. Based on these requirements, three principal design goals were identified. After sketching the simplified human immune system, their salient features that can contribute to build a competent network-based intrusion detection system were analysed. This analysis show that the human immune system is equipped with a number of sophisticated mechanisms, which satisfy the three identified design goals. Consequently, the design of a novel network-based IDS based on the human immune system is promising for future network-based IDSs. his research also investigated the existing network-based IDSs. They were categorised into three different approaches: monolithic, hierarchical and co-operative and problems were identified for each approach. In order to resolve these problems, a novel artificial immune model was presented. This model combines the three evolutionary stages: gene library evolution, negative selection and clonal selection into a single methodology. These three processes are co-ordinated across a network to satisfy the three goals for designing effective IDS's: being distributed, self-organizing and lightweight. Analysis of the characteristics of this unified evolutionary approach show that, unlike existing approaches, the proposed artificial immune model does satisfy the requirements of network-based IDSs. Consequently, algorithms based on this model show considerable promise for future IDSs.

A network-based IDS utilizing the artificial immune model is being implemented in order to prove the validity of this approach. Current work is focusing on building initial self profiles and detectors from normal and abnormal TCP/IP packets, which were collected from a real network environment. As the first attempt of this effort, the negative selection stage was implemented and experiments showed its infeasibility for its application to the essential profiling fields of real network data. This result directs this research to re-define the role of negative selection algorithm within the overall artificial immune system framework. Finally, the intrusion detection mechanism of clonal selection stage were investigated and the clear understanding of task of clonal selection stage helped us to comprehend the distinct job of negative selection stage.

The contributions of this work will provide an applicable methodology for designing an artificial immune system to be able to perform network intrusion in a truly distributed, self-organizing and lightweight way.

제6장 에이전트 기반의 Intruder Detection System 개발

본 장에서는 에이전트를 사용한 외부침입탐지 시스템에 대해 서술한다. 개발된 시스템에서는 침입 탐지의 기반 정보가 되는 메시지 처리상의 메시지 유실 및 메시지 내 정보에 대한 처리를 보다 안정적으로 제공하기 위해 단위 침입 행동별로 학습된 모니터링 프로세스들로부터 전송되어지는 메시지에 대한 처리를 담당하는 조정자 에이전트 시스템을 제안한다. 제안된 조정자는 안정화된 메시지 처리 문제 뿐 아니라, 기존 모델의 에이전트간 협력 작업에 의해 처리되었던 침입 판단 기능 및 모니터링 프로세스들의 관리 기능 또한 수행하도록 하여 시스템의 유연성 및 확장성 향상을 도모하도록 하였다.

제1절 시스템 접근방법

네트워크에서의 침입은 크게 오용침입(Misuse Intrusion)과 비정상적인 침입(Anomaly Intrusion)의 두 가지로 분류한다. 오용 침입(Misuse Intrusion)은 시스템이나 응용프로그램 소프트웨어의 알려진 약점이나 버그를 이용한 침입으로 예를 들면, 인터넷에서 사용되는 send mail이나 fingered의 버그를 예로 들 수 있다. 비정상적 침입(Anomaly Intrusion)은 정상적인 시스템의 사용 패턴들과의 차이(Deviation) 관찰에 의해 침입이 결정되는 것으로, 모니터 되어야 할 시스템의 프로파일(profile)을 구축하여 이로부터 중요한 차이를 찾아냄으로써 침입이 탐지된다. 오용 침입은 잘 알려진 패턴을 따르므로 Audit trail 정보와의 패턴 매치를 통해 탐지할 수 있다. 그러나 이에 비해 비정상적 침입은 탐지하기가 쉽지 않다.

그리고 정상적인 사용자의 행동에 대해 이를 침입으로 분류해 버리는 경우, 이를 무시할 수 있는 능력 또한 갖추어야 하는데, 이에 대한 판단은 보안 담당자에 의해 사전에 규정해 놓는다던가, 오랜 시간 시스템을 사용해본 경험에 근거한다던가 하는 Heuristic적인 면에 의존하게 되는 것이다.

침입 탐지 시스템에 있어 행위 판별(Behavior Classification)과 자료 축소(Data Reduction) 문제를 중심으로 수행한 연구들이 있다. 행위 판별은 주어진 일련의 행위들에 대해 이것이 침입인지 침입이 아닌지를 판단할 수 있는지를 결정하는 문제이고, 자료 축소는 수 메가바이트에 이르는 방대한 양의 분석되어져야 할 데이터에 대해 이의 양을 줄여 나가는 것을 말한다. 대체적으로 이러한 문제의 해결에 있어

도입된 인공지능의 기법들은 규칙 기반 시스템(Rule-based System)과 신경망[5] 또는 통계적 분류 시스템의 방법을 사용하고 있다. 이러한 접근법의 한 계점들은 이들이 많은 양의 초기 학습을 필요로 하며, 시스템의 수명동안 지속적으로 시스템의 유지보수를 위해 많은 노력이 필요하다는 것이다.

규칙 기반 침입 탐지 시스템의 대표적인 예로는 IDES 시스템이 있다. 이 시스템은 대상 시스템의 취약성 및 보안 정책, 그리고 과거의 침입들에 대한 지식을 담고 있는 규칙 데이터베이스를 가지고 있다. 현재의 시스템의 상태로부터 침입이 발생할 경우, 탐지 시스템은 침입에 대한 해당 규칙에 의해 현 시스템이 침입 당했는지 여부를 분류하게 된다. 그러나 IDES시스템은 통계적인 기존의 규칙 기반 접근법에 비해 과거의 침입에 대해 이를 기억하고 있다는 중요한 특징이 있다. 이것은 침입 탐지 시스템의 성능 향상에 있어 매우 중요한 특징으로 대부분의 새로운 침입 형태들은 기존 형태의 부분적인 변형이기 때문이다.

Handy, Luger등에 의해 제안된 새로운 솔루션으로는 분류 시스템(Classifier System)을 이용하여 네트워크 시스템의 현재 상태를 분류해 내는 방법이 있다.[8] 이것은 네트워크 패킷 정보에 관한 매트릭스들을 구한 후 이들로부터 네트워크에 관한 분류를 어떻게 할 수 있는지를 추론하게 된다. 그러나 이러한 접근법은 크게 두 가지의 한계점을 지니고 있다. ATM이나 FDDI 백본과 같이 많은 컴퓨터들이 빠른 속도의 네트워크로 연결되어 있는 경우 성능의 저하가 현격하다는 것과 네트워크의 상태를 판단하는데 사용되는 정보가 패킷의 헤더데이터에 국한되어 있다는 점이다. 헤더로부터 추출한 정보로부터는 행위의 특징을 처리할 수 없기 때문에 헤더의 정보만으로는 분류에 필요한 유용한 정보를 추출하기에는 역부족이다. 이에 대한 예로는 합법한 메일 포트를 통하여 침입이 이루어질 경우 이 접근법에 의한 시스템은 침입을 구분해 낼 수 없게 된다.

Kumar와 Spaffod에 의해 제안된 패턴 매칭 기법에 기반한 접근법은 시스템상에 요구되는 유연성의 향상에 대한 문제로 초점이 맞추어 졌지만, 학습능력을 갖추지 못하였다는 단점을 가지고 있다. 이들은 시스템 상에 나타나는 현상들에 근거하여 침입을 어떻게 분류하는 지를 보여주었다. 여기서의 각 패턴들은 시스템 상태들간의 의존도를 인코딩하고 있는 것이다. 이러한 접근법은 침입을 탐지하는 강력한 방법이나 사전에 만들어진 패턴들에 의존적이라는 단점 또한 가지고 있다. 즉, 패턴 자체가 완전하지 못할 경우 시스템의 방어에 커다란 허점이 나타나게 되는 것이다. 그리고 보안 정책이나 시스템 운영상에 변화가 있을 경우 패턴들을 다시 만들어야 한다.

제2절 시스템 설계

기존에 제안되어진 대부분의 침입 탐지 시스템들은 하나의 통합된 단일 시스템으로 커널과 같은 대상 시스템의 운영체제 위에 놓여져 커널로 들어오는 모든 처리 요구에 대해 모니터링을 하도록 되어있다. 그러나 이러한 시스템 모델들은 전체 시스템에 걸리는 부하문제 및 탐지 모듈의 파괴에 따른 안정성의 문제, 시스템의 확장에 따른 성능 보장의 문제 등과 같은 많은 문제점을 지니게 되었다.[2] 이러한 성능상의 문제들의 해결을 위해 대상 시스템을 지역적, 또는 기능적으로 분할한 후, 가벼운 형태의 여러 개의 프로세스들로 하여금 각각 독립적인 동작으로 분할된 시스템 자원을 모니터링하고, 전체 시스템에 대한 침입 발생시 이들 프로세스간의 협력을 통해 이를 탐지하도록 침입 탐지 시스템을 구성할 필요가 있다. 이러한 구조는 기존 시스템에 대한 부하문제 및 탐지 모듈의 파괴에 따른 전체 기능의 마비, 시스템의 확장에 따른 탐지 시스템의 확장성 보장 등의 문제점들을 해결할 수 있다.

1. 독립 에이전트 기반 침입탐지 시스템

본 연구에서 제안하고 있는 시스템에서는 각 모듈 프로세스들이 시스템 자원들에 대한 행동패턴을 관찰하여, 비정상적인 행동이라 여겨질 경우 이를 알릴 수 있도록 학습되어진 에이전트의 형태를 가진다. 학습을 위해서는 유전자 알고리즘을 사용하여 의심스러운 행위 패턴에 대해 이를 분별할 수 있도록 학습시킨 후 각 모니터링 대상이 되는 자원들 위에 올려놓는다.

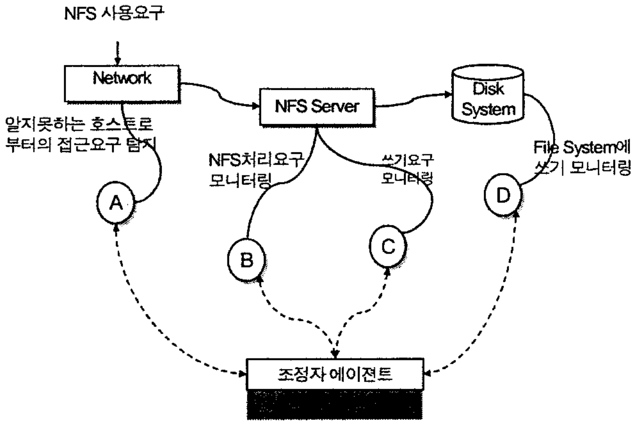


그림 25 :독립 에이전트를 이용한 침입 탐지 시스템의 구성

시스템 사용자들에 의한 행동은 일련의 시스템 자원에 대한 접근 및 서비스 요구들로 이루어짐으로 각각 대상 시스템 자원에서 사용자의 행태가 모니터링된다. 상호 독립적으로 동작하는 에이전트들은 모니터링 중 의심스런 행위가 발생하였을 경우, 이를 주변의 에이전트들에게 알리고(Broadcasting) 의심스러운 특정 사용자의 행위에 대해서는 계속적으로 모니터링을 하게 된다. 만약 그 행동이 시스템 전체에서 허용하는 특정 수위의 한계치(Threshold)를 넘을 경우 이를 비정상적인 행동으로 여겨 이에 대한 대응 및 보고를 수행하게 된다. 이러한 침입에 대한 탐지 메카니즘은 탐지 시스템 상에서 발생할 수 있는 오류중 False-positive(긍정적 오류) 오류의 발생가능성을 최소화시키고, 각각의 에이전트들을 해당 리소스에 대한 모니터링을 위해 최적화해 구성할 수 있음으로 인해, 궁극적으로 침입 탐지 시스템의 성능 향상을 도모할 수 있다. 다음 그림은 독립에이전트 시스템의 작업 예를 보인 것이다.

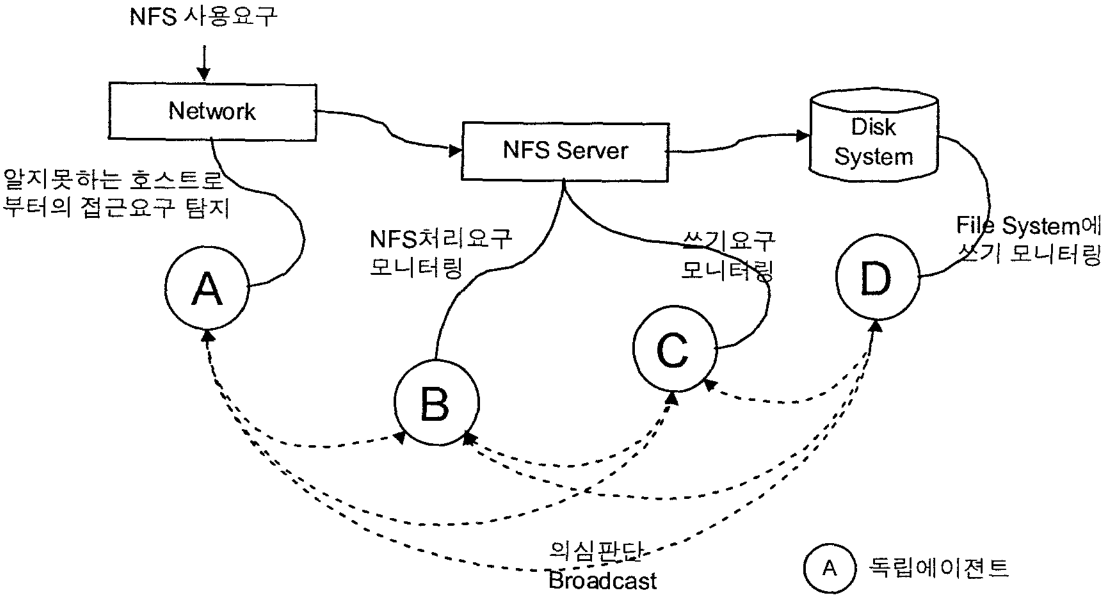


그림 26 : 독립에이전트의 활동예

2. 에이전트의 학습을 위한 Genetic Programming의 사용

본 연구에서 제안되는 탐지시스템에서 에이전트는 audit 정보를 모니터하기 위하여 침입프로파일에 대하여 학습을 하여야 한다. 이런 필요에서 도입된 Genetic Programming(GP)(Koza 1992)은 에이전트 프로그램이 새로운 침입유형과 알려지지 않은 침입유형에 따라 계속적으로 진화할 수 있으며 그리고 나서 실제 환경에서 사용될 수 있다. Genetic Programming의 최종적인 결과는 프로그램의 집합이며 이 프로그램은 실제 시스템에 탑재되어 연속적으로 동작한다. Genetic Programming은 audit data field에 접근할 수 있으며 audit data를 잘 조정할 수 있는 성격을 가지는 단순한 언어로 코드화된다. stand-alone solution에서 이 Genetic Programming은 audit 정보와 함께 이것을 제공하는 evaluator에 의해 해석된다. 아래그림은 에이전트의 내부구조를 보여주고 있다.

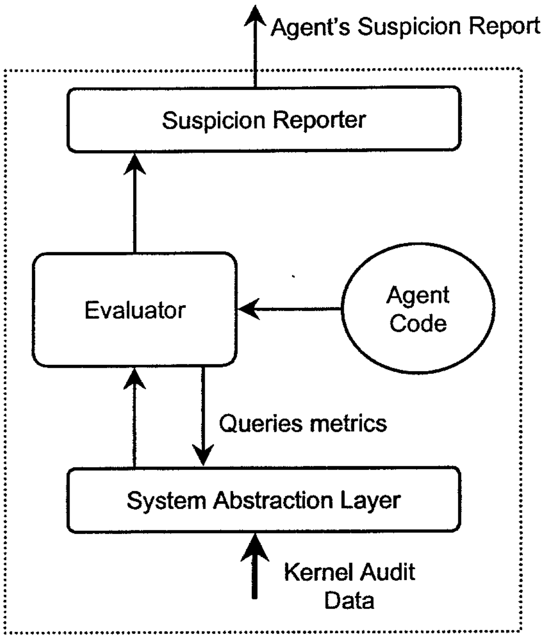


그림 27 : 독립 에이전트의 내부구조

에이전트가 가지고 있는 코드는 진화과정으로부터 얻게 되고 evaluator에 의해 에이전트 안에 놓여진다. evaluator는 SAL(System Abstraction Layer)로부터 audit 정보를 획득하게 된다. SAL은 audit record에 대하여 다양한 통계를 계산하고 agent에게 그 정보를 제공하며 audit record를 해독하고 필요한 영역으로 추출한다. SAL은 모든 시스템의 기본적인 요소(평균 CPU 사용률, 평균 login 횟수등)를 연속적으로 제공하는 목적을 가지고 있다.

가. Learning module

에이전트를 학습시키는 모듈로서 한번 학습된 에이전트들은 학습모듈과 독립적으로 동작할 수 있고, 또한 에이전트들이 동작하는 동안의 모든 기록은 다음 학습의 입력으로 사용된다.

나. Agents

에이전트는 제안 침입탐지모델에서 중요한 역할을 하는 부분으로서 하위 네트워크 layer에서 올라오는 패킷을 탐지모듈로 검사하여 시스템의 침입여부를 판단한다. 각 에이전트들은 자신이 맡은 모듈에 입력되는 메시지를 검사하여 침입으로 간주되면 침입 추정 값을 증가시킨다. 이 침입추정값이 특정임계치에 이르면 이를 확실한 침입으로 간주하게 된다. 다음은 UDP bomb공격을 탐지하는 에이전트의 코드의 예이다.

```
for-each-packet do
  if (get-subnet-part(ip-dest-addr-of-packet)
    is-not-equal-to my-subnet-address)
  then
    generate-a-suspicion-broadcast

    if ( (packet-protocol equals UDP) and
        (udp-dest-port equlas 520)) then
          generate-a-suspicion-broadcast
    endif
  endif
endfor
```

다. DLP1

Sun DLP1 인터페이스로 응용프로그램이 실제 데이터 링크 계층의 패킷을 전송하고 받을수 있게 해주는 인터페이스이다.

라. Network Primitive Abstraction

DLP1 인터페이스로부터 실제 네트워크 패킷을 받아서 에이전트들이 그 패킷을 다룰수 있도록 패킷의 구조를 바꾸어 주는 레벨이다. 에이전트들은 비정상적이라고 생각되는 일련의 행동에 의심에 대한 표시를 하고 시스템 행위를 관찰하도록 훈련받는다. 이 원형에서는 에이전트는 시스템상의 네트워크 트래픽을 감시한다. 에이전트를 이용한 시스템 감시는 먼저 시스템 관리자와 같이 시스템 보안에 관련된 사람의 수작업에 의한 지식데이터의 수립이 되어야 하는데 이것이 에이전트가 효과적으로 임무를 수행할 수 있는 선결 과제이다. 또한 이 지식데이터를 가지고 에이전트를 학습시키고 시스템에 적응시키는데 필요한 시간이 경우에 따라 많이 소모될 수 있으며 또한 너무 낮은 레벨의 네트워크에서는 시스템 감시를 위한 정보의 수집이 어려워지게 되어 에이전트 본래의 목적과 상응하지 않게 된다. 또한 에이전트가 얼마나 빠르게 침입의 유형을 학습하고 정확하게 침입을 시스템 운영자에게 알릴 수 있는지에 대한 객관적인 근거는 없다. 만약 내부 사용자들 중 비정상적인 사용자 행위나 오용에 대한 패턴이나 경로가 빠르게 바뀔 경우 실제적으로 에이전트의 데이터추출, 학습 및 협동을 통한 감시는 상당한 한계를 가질 수 밖에 없다. 그러나

이러한 단점에도 불구하고 에이전트는 어느 정도 시스템에 대한 적응과 훈련을 위한 시간이 소모되고 실제로 에이전트가 시스템의 감시자로서 역할을 하게 되면 과거의 시스템 침입 탐지를 위한 방법들보다 상당한 도움을 줄수 있다.

3. 조정자 에이전트 중심의 협력에이전트 시스템의 설계

본 연구에서 제시하는 새로운 침입탐지 모델은 audit data를 통한 해석기를 통하여 침입판단 및 탐지모듈과 시스템의 전반적인 조정을 할수 있는 조정자 에이전트와 그 하부에 활동적인(부분적인 제어를 단위행동 에이전트)에이전트를 기반으로 두 단계의 에이전트를 두어 상부의 에이전트는 genetic을 이용한 내부사용자의 비정상적인 사용과 오용 행위 패턴을 지식데이터베이스로 구축하고 coordinator agent와 autonomous agent간에는 black board 시스템을 구성하여 실제적으로는 시스템상의 다양한 여러 에이전트들이 협조하여 침입을 감시하고 새로운 형태의 침입에 대한 반응 속도를 높여 실시간적인 침입 탐지 시스템을 구축하는 기본적인 형태이다. 제안 모델 상에서 개별 작업을 수행하는 독립 모니터링 프로세스들과 일대 다로 연결되어 안정적 메시지 처리 지원, 침입 탐지 및 보고, 그리고 이들 모니터링 프로세스들로 구성되어진 전체 시스템 구조의 유지 및 운용을 위한 관리자를 본 연구에서는 조정자(Coordinator)로 명명한다. 본 연구를 통해 제안하는 조정자는 독립 에이전트 기반의 침입 탐지 시스템 상에서 에이전트들에 의해 수행되어야 했던 많은 역할을 이양 받게 되고, 또 이들을 보완시킨다. 그러므로 제안 모델에서의 모니터링 프로세스와 FGA에서의 에이전트간에는 그 성격 및 역할에 있어 몇 가지 중요한 차이를 갖는다.

-

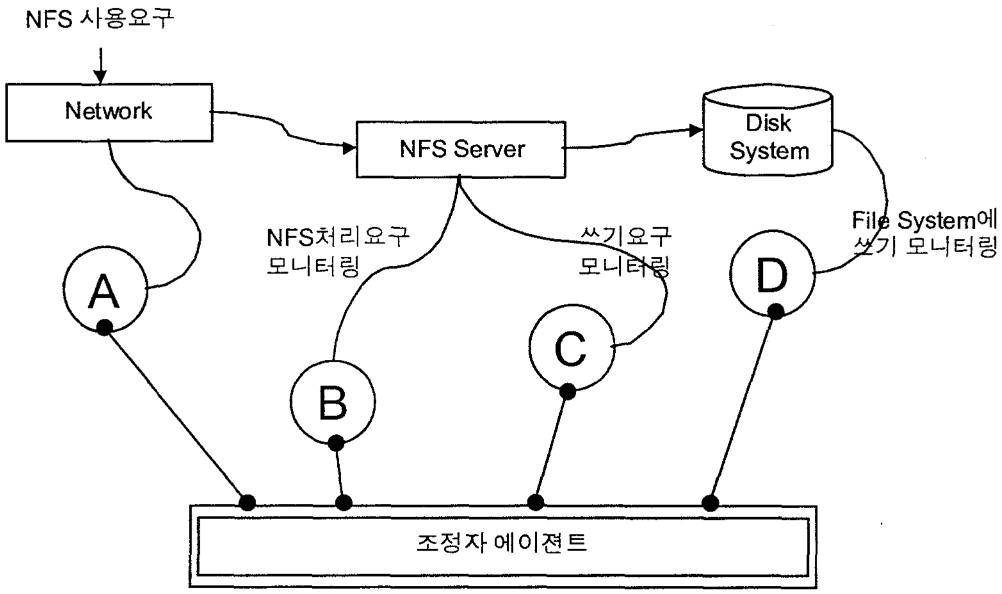


그림 28 :조정자 에이이젼트와 독립 에이젼트간의 메시지교환

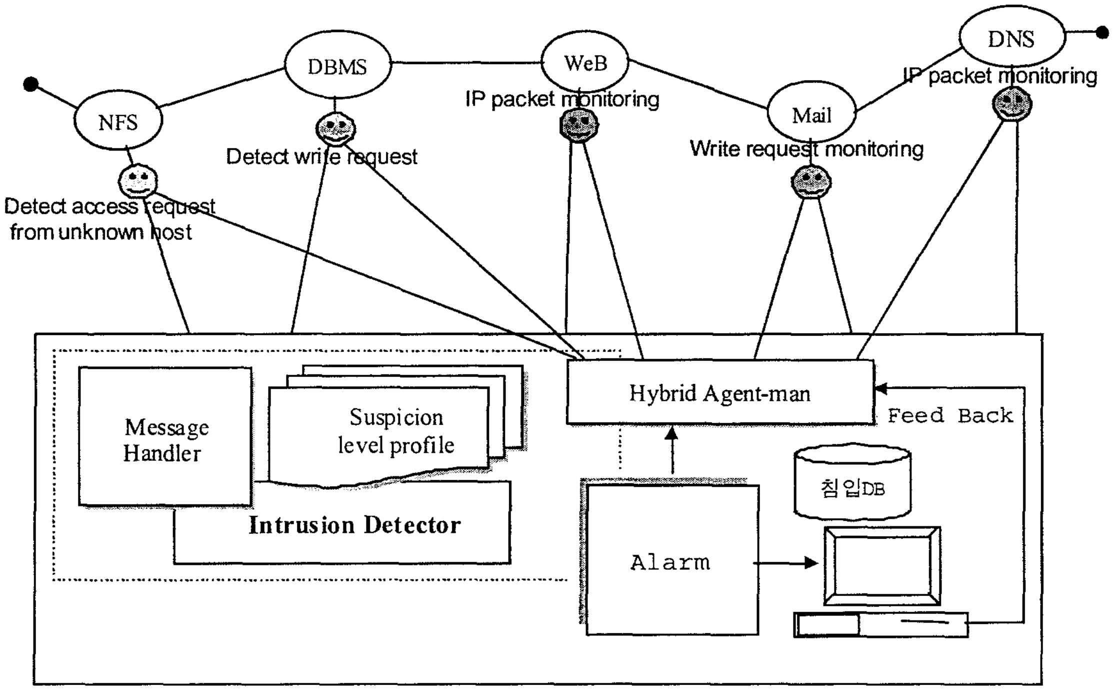


그림 29 : 침입탐지시스템의 전체구조

우선 제안 모델에서의 모니터링 프로세스는 FGA에서의 에이전트와 같이 위협행동을 수행한 수많은 사용자들에 대해 주위의 에이전트로부터 회송되어진 메시지에 대한 처리 부하가 필요하지 않다. 또한 회송 메시지를 전달되어지는 메시지 정보에 대해서도 제안 모델의 경우 이를 중앙의 조정자에서 관리하게 됨으로 모니터링 프로세스 상에 이 정보에 대한 별도의 저장 및 처리 모듈이 필요하지 않게 된다. 결

국 FGA에서의 에이전트들에 비해 제안 모델에서의 프로세스들은 훨씬 가벼운 형태로 존재하게 되며, 그 역할은 학습에 의해 판단할 수 있는 개별 시스템 자원에 대한 사용자의 위협 행동을 찾아내 그 사실을 중앙의 조정자에 전달하기만 하면 되는 것이다. 이는 모니터링 프로세스 기반의 침입 탐지 시스템의 물리적, 또는 논리적 확장성을 제공할 뿐 아니라, 시스템의 유연성 등 FGA 모델이 기존의 중앙집중식 침입 탐지 시스템들에 대해 갖는 장점들을 강화할 뿐 아니라, 침입 탐지의 기반이 되는 메시지 처리를 중앙 조정자내 메시지 처리자가 존재하여 안정적으로 제공하게 된다. 조정자 에이전트의 입장에서 시스템을 바라보면 조정자 에이전트는 시스템의 자체 침입 판별 알고리즘을 지원하기 위해 모니터링 프로세스로부터의 메시지 가공, 사용자별 행위 데이터의 작성 및 이를 기반한 침입 판단, 그리고 이에 대한 보고 기능 등과 함께 전체 침입 탐지 시스템의 유지 및 운용을 위해 프로세스들의 관리 기능까지 수행하게 된다. 이렇듯 모니터링 프로세스들과 상호작용하며, 침입에 대한 탐지 및 이에 대한 보고, 그리고 모니터링 프로세스들에 대한 관리를 위한 조정자는 기존의 중앙집중식 시스템 구조를 가진 지식 기반의 침입 탐지 시스템의 우수한 기능을 갖고 있으며, 그 구조에 있어 유사한 면이 존재한다.

제3절 침입탐지 검사모듈의 구성

침입 탐지 검사모듈은 임의의 패턴을 source address와 protocol port&signature, 그리고 traffic분석을 통하여 침입행위인지를 판별한다. 다음그림은 침입탐지 모듈의 구성도이다.

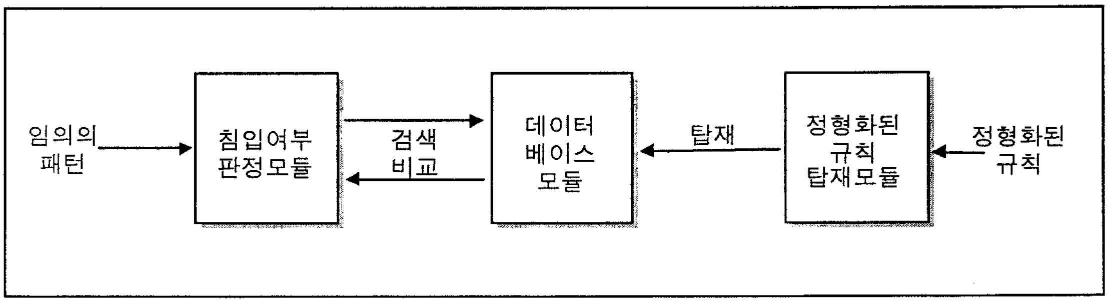
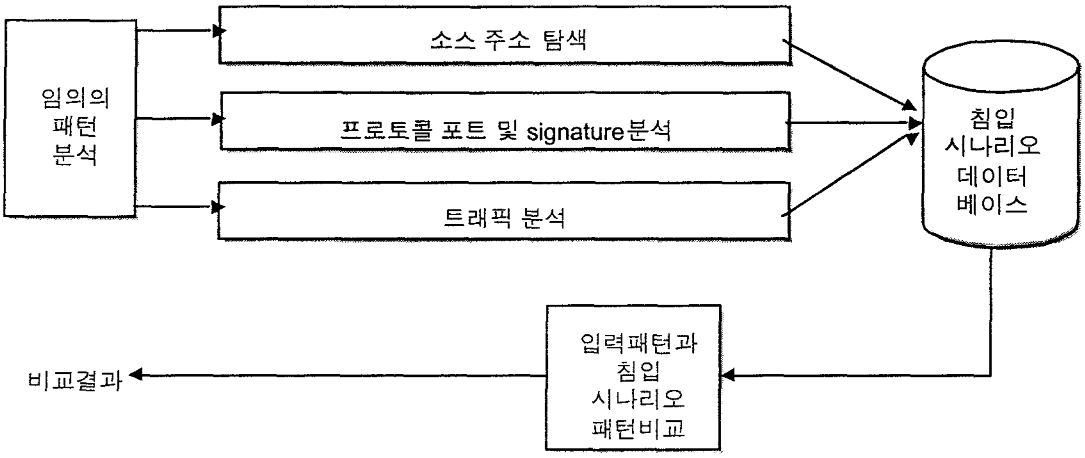


그림 30 : 침입여부 판정엔진

임의의 사용자 행위는 침입여부를 판정하기 위해 위의 3가지 기준으로 '침입여부판정모듈'에서 분석된다. 분석된 결과는 침입패턴들이 모아져있는 '데이터베이스모듈'과 비교하여 침입인지 아닌지를 가려낸다. 또한 새로운 침입패턴들이 발생한 경우에는 '정형화된 규칙 탑재모듈'에서 데이터베이스 모듈로 새로운 패턴이 탑재된다.

1. 침입여부 판정모듈

침입여부판정모듈은 임의의 패턴을 source address와 protocol port&signature, 그리고 traffic 분석후에 침입관련DB에 검사하여 침입인지를 판정하는 모듈이다. 다음 그림은 침입여부 판정모듈의 구성도이다.



임의의 패턴 분석
소스 주소 탐색
프로토콜 포트 및 signature분석
트래픽 분석
침입 시나리오 데이터 베이스
입력패턴과 침입 시나리오 패턴비교
비교결과

그림 31 : 침입여부 판정모듈

2. 침입관련 데이터베이스 모듈

침입관련 데이터베이스 모듈은 여러가지 침입패턴들을 모아놓은 DB라고 할수 있는데, 이 DB에 얼마나 많은 양의 침입패턴들이 있느냐가 전체적으로 중요한 문제가 된다. 침입여부 판정모듈에서 들어온 임의의 패턴을 분석하여 보내진 query들은 세가지 기준을 모두 거쳐 침입인지 여부를 판정하게 된다. 다음 그림은 데이터베이스 모듈의 구성도이다.

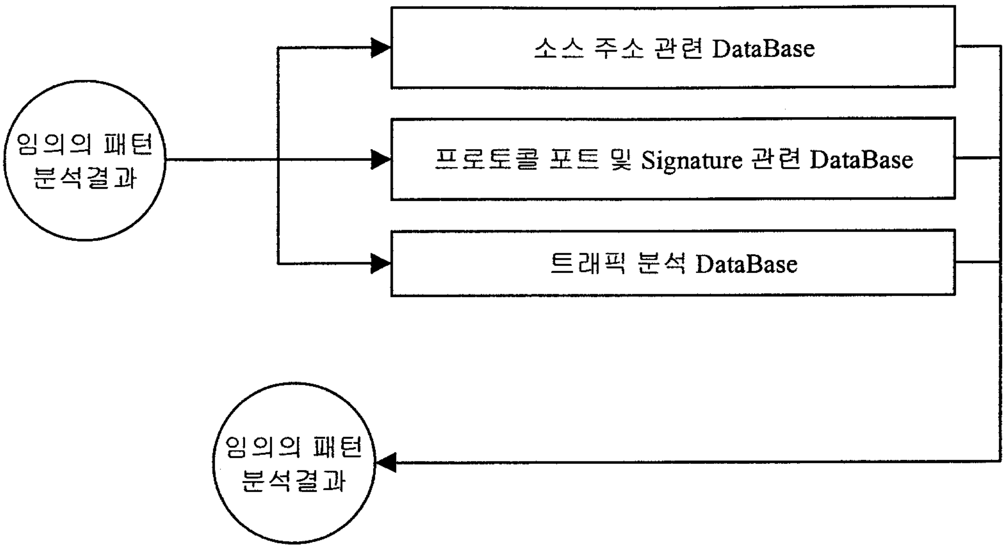


그림 32 : 침입관련 데이터베이스 모듈의 설계

위의 그림 구성도에서 세가지 DataBase source address관련 DataBase, protocol port&signature관련 DataBase,Traffic 분석 관련 DataBase 대해서 각각 field구성과 실제 패턴예를 살펴보도록 하자.

표는 source address관련 DataBase의 Field구성으로 Field는 rule name과 source address, source port, destination address, destination port, 그리고 허용여부를 결정하는 action으로 구성된다. 이에 대한 패턴 예를 다음 표에서 보여주고 있다.

Field Name	*Field Type*	*Field Length*	*Description*
Rule Name	String	10	규칙의 이름
Source Address	String	15	근원지 이름
Source Port	Integer	2	근원지 프로토콜
Dest. Address	String	15	목적지 주소
Dest. Address	Integer	2	목적지 포트번호
Action	String	1	D(Deny) or P(Permit)허용여부

표 45 : source address 관련 Database Fields

Field Name	Field Type	Field Length	Description
Rule Name	String	10	규칙의 이름
Command	String	10	사용한 명령어
Arguement#1	String	10	인자 1
Arguement#2	String	10	인자 2

표 46 : Protocol port & Signature 관련 Database의 Field

Rule	*Src.Addr*	*Src.Port*	*Dest.Addr*	*Dest Port*	*Action*
A	203.237.174.182	8	172.16.16.0	24	Permit
B	203.253.64.1	15	202.34.89.1	16	Deny
C	203.237.168.1	4	198.34.6.1	119	Deny

표 47 : Source Address 관련 Database패턴

3. 침입탐지모듈의 상세설계

지금까지 침입탐지 검사모듈의 세가지 구성성분의 각각에 대한 구성도와 패턴예를 살펴보았다. 다음그림은 침입여부 판정모듈의 상세설계이다.

임의의 패턴은 source address, source port가 존재하는지를 조사한다. 존재할 경우 탐색해서 source address관련 DataBase에 Query 를 보낸다. 존재하지 않을 경우에는 다음 단계로 내려간다. 다음 단계에서는 ftp, telnet과 같은 protocol port등이 존재하는지를 검사한다. 이것도 마찬가지로 존재할 경우는 protocol port관련DB로 Query를 보내고 없으면 다음단계로 내려간다. 3번째 단계에서는 traffic과 관련된 e-mail, finger등이 존재하게 될 것이다. 여기서의 결과는 traffic관련 Database로 Query를 보내게 된다. 이렇게 보내진 Query들은 각각 matching이 되는지 검사되어지며, 한가지라도 matching이 되면 침입으로 결정되어진다.

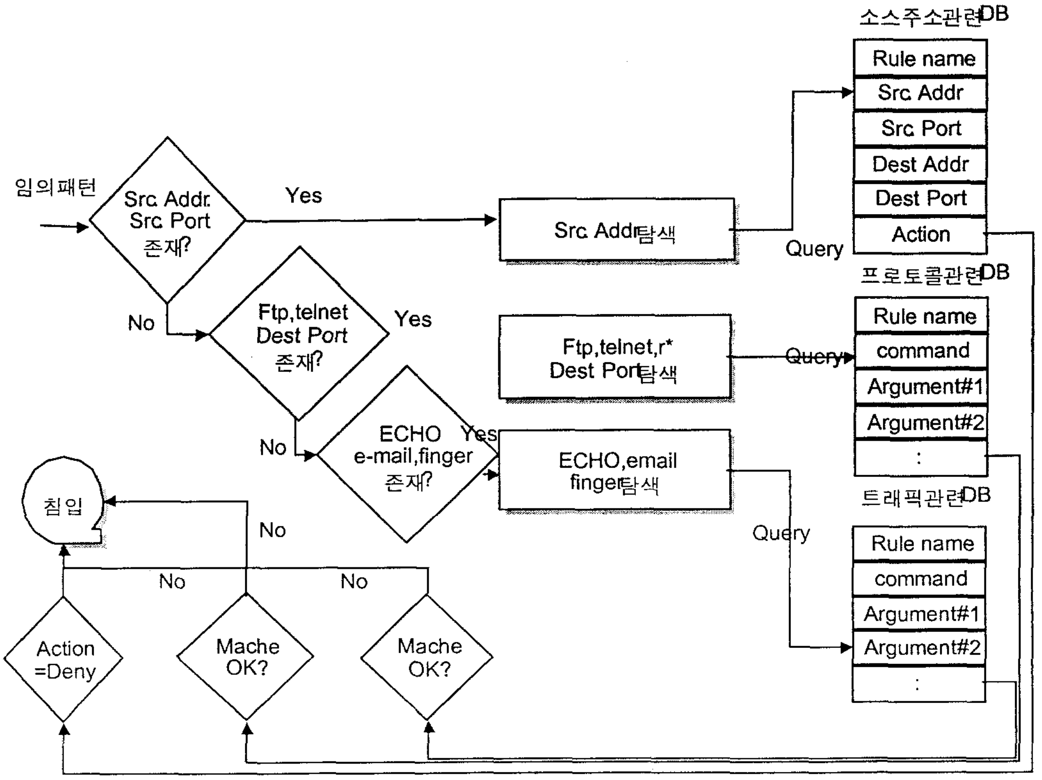
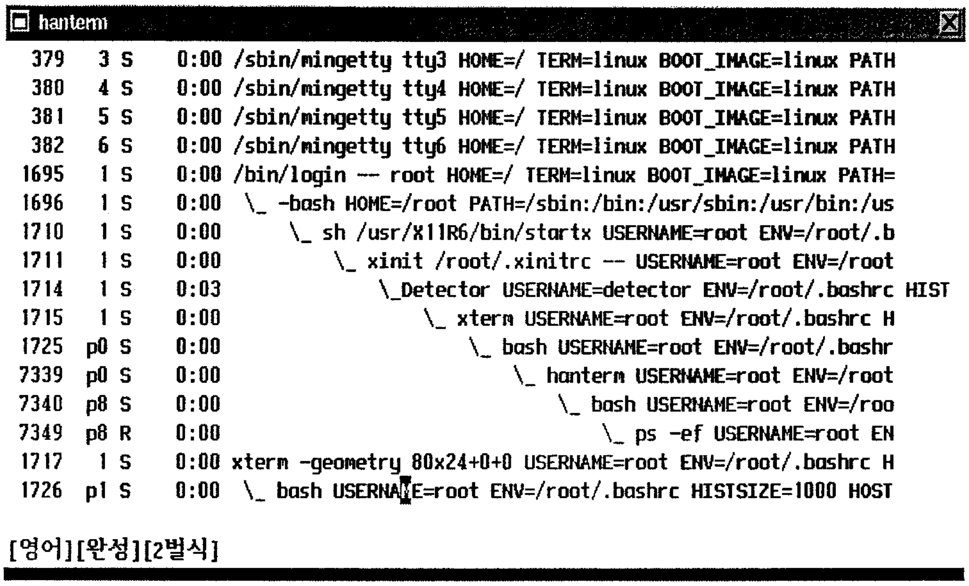


그림 33 :침입여부 판정모듈의 상세설계

제4절 연구결과 및 Prototype

침입탐지 에이전트 프로그램은 Unix 시스템상에서 데몬프로세스로 동작하며 시스템에 대한 이상 작용이 즉각 화면상에 이상유무를 경고메시지와 함께 출력하며 그 사용자를 지속적으로 감시하게 된다. 다음 그림은 침입탐지 데몬인 Detector가 데몬프로세스로 동작중인 것을 프로세스 ID로 확인해본 그림이다.



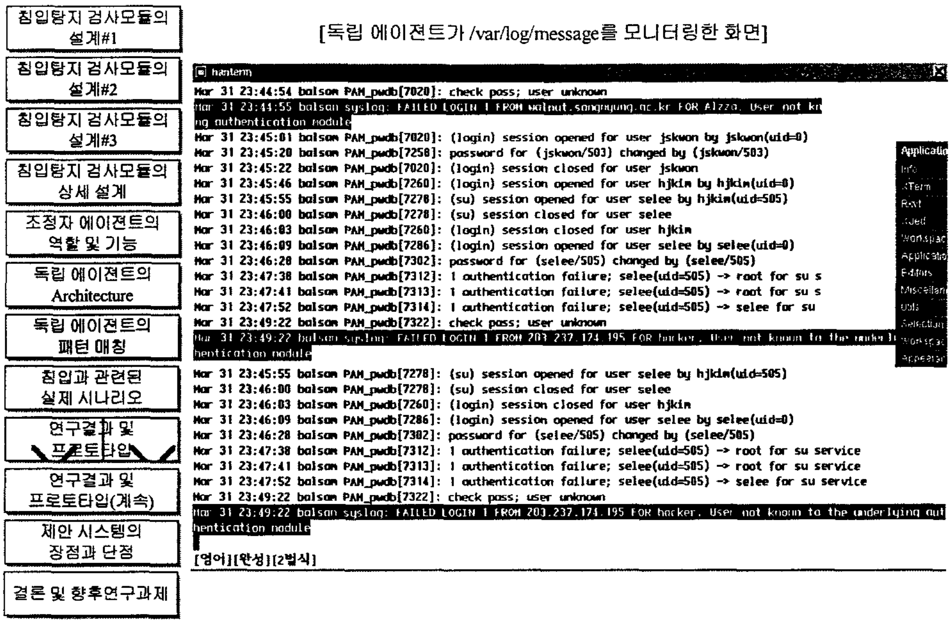
hanterm

```
  379   3 S    0:00 /sbin/mingetty tty3 HOME=/ TERM=linux BOOT_IMAGE=linux PATH
  380   4 S    0:00 /sbin/mingetty tty4 HOME=/ TERM=linux BOOT_IMAGE=linux PATH
  381   5 S    0:00 /sbin/mingetty tty5 HOME=/ TERM=linux BOOT_IMAGE=linux PATH
  382   6 S    0:00 /sbin/mingetty tty6 HOME=/ TERM=linux BOOT_IMAGE=linux PATH
 1695   1 S    0:00 /bin/login -- root HOME=/ TERM=linux BOOT_IMAGE=linux PATH=
 1696   1 S    0:00  \_ -bash HOME=/root PATH=/sbin:/bin:/usr/sbin:/usr/bin:/us
 1710   1 S    0:00      \_ sh /usr/X11R6/bin/startx USERNAME=root ENV=/root/.b
 1711   1 S    0:00          \_ xinit /root/.xinitrc -- USERNAME=root ENV=/root
 1714   1 S    0:03              \_Detector USERNAME=detector ENV=/root/.bashrc HIST
 1715   1 S    0:00                  \_ xterm USERNAME=root ENV=/root/.bashrc H
 1725  p0 S    0:00                      \_ bash USERNAME=root ENV=/root/.bashr
 7339  p0 S    0:00                          \_ hanterm USERNAME=root ENV=/root
 7340  p8 S    0:00                              \_ bash USERNAME=root ENV=/roo
 7349  p8 R    0:00                                  \_ ps -ef USERNAME=root EN
 1717   1 S    0:00 xterm -geometry 80x24+0+0 USERNAME=root ENV=/root/.bashrc H
 1726  p1 S    0:00  \_ bash USERNAME=root ENV=/root/.bashrc HISTSIZE=1000 HOST

[영어][완성][2벌식]
```

그림 34 : 탐지에이전트 데몬프로세스

아래 그림에서와 같이 여러 경로에서 login을 시도한 외부 사용자들이 단 한번의 login실패에도 콘솔상의 모니터 상에서 login실패가 어느 경로로부터 일어났는가를 알수 있다. 그리고 내부 사용자에 의한 루트권한으로의 시도 역시 경고메시지와 함께 관리자의 모니터상에 출력되어 내부사용자에 의한 권한 남용을 사전에 인지할 수 있다.



침입탐지 검사모듈의 설계#1

침입탐지 검사모듈의 설계#2

침입탐지 검사모듈의 설계#3

침입탐지 검사모듈의 상세 설계

조정자 에이전트의 역할 및 기능

독립 에이전트의 Architecture

독립 에이전트의 패턴 매칭

침입과 관련된 실제 시나리오

연구결과 및 프로토타입

연구결과 및 프로토타입(계속)

제안 시스템의 장점과 단점

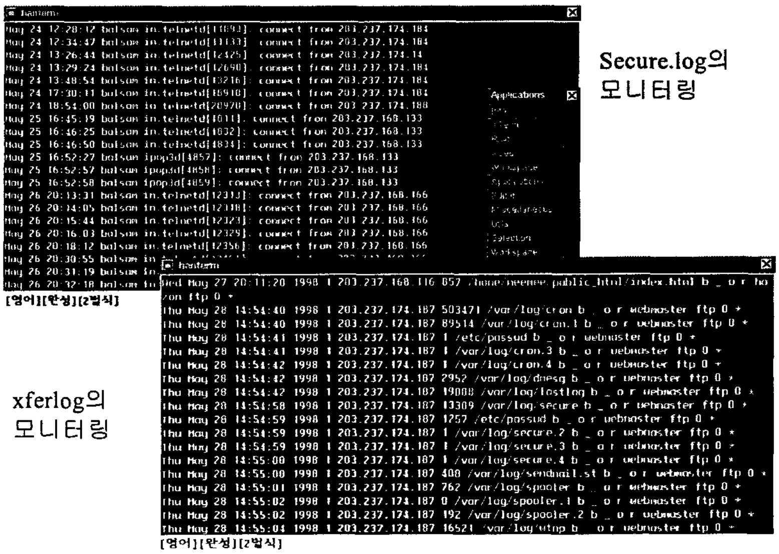
결론 및 향후연구과제

[독립 에이전트가 /var/log/message를 모니터링한 화면]

```
hanterm
Mar 31 23:44:54 balsom PAM_pwdb[7020]: check pass; user unknown
Mar 31 23:44:55 balsan syslog: FAILED LOGIN 1 FROM walnut.sangmyung.ac.kr FOR Alzza, User not kn
ng authentication module
Mar 31 23:45:01 balsom PAM_pwdb[7020]: (login) session opened for user jskwon by jskwon(uid=0)
Mar 31 23:45:20 balsom PAM_pwdb[7258]: password for (jskwon/503) changed by (jskwon/503)
Mar 31 23:45:22 balsom PAM_pwdb[7020]: (login) session closed for user jskwon
Mar 31 23:45:46 balsom PAM_pwdb[7260]: (login) session opened for user hjkim by hjkim(uid=0)
Mar 31 23:45:55 balsom PAM_pwdb[7278]: (su) session opened for user selee by hjkim(uid=505)
Mar 31 23:46:00 balsom PAM_pwdb[7278]: (su) session closed for user selee
Mar 31 23:46:03 balsom PAM_pwdb[7260]: (login) session closed for user hjkim
Mar 31 23:46:09 balsom PAM_pwdb[7286]: (login) session opened for user selee by selee(uid=0)
Mar 31 23:46:28 balsom PAM_pwdb[7302]: password for (selee/505) changed by (selee/505)
Mar 31 23:47:38 balsom PAM_pwdb[7312]: 1 authentication failure; selee(uid=505) -> root for su s
Mar 31 23:47:41 balsom PAM_pwdb[7313]: 1 authentication failure; selee(uid=505) -> root for su s
Mar 31 23:47:52 balsom PAM_pwdb[7314]: 1 authentication failure; selee(uid=505) -> selee for su
Mar 31 23:49:22 balsom PAM_pwdb[7322]: check pass; user unknown
Mar 31 23:49:22 balsan syslog: FAILED LOGIN 1 FROM 203.237.174.195 FOR hacker, User not known to the underlu
hentication module
Mar 31 23:45:55 balsom PAM_pwdb[7278]: (su) session opened for user selee by hjkim(uid=505)
Mar 31 23:46:00 balsom PAM_pwdb[7278]: (su) session closed for user selee
Mar 31 23:46:03 balsom PAM_pwdb[7260]: (login) session closed for user hjkim
Mar 31 23:46:09 balsom PAM_pwdb[7286]: (login) session opened for user selee by selee(uid=0)
Mar 31 23:46:28 balsom PAM_pwdb[7302]: password for (selee/505) changed by (selee/505)
Mar 31 23:47:38 balsom PAM_pwdb[7312]: 1 authentication failure; selee(uid=505) -> root for su service
Mar 31 23:47:41 balsom PAM_pwdb[7313]: 1 authentication failure; selee(uid=505) -> root for su service
Mar 31 23:47:52 balsom PAM_pwdb[7314]: 1 authentication failure; selee(uid=505) -> selee for su service
Mar 31 23:49:22 balsom PAM_pwdb[7322]: check pass; user unknown
Mar 31 23:49:22 balsan syslog: FAILED LOGIN 1 FROM 203.237.174.195 FOR hacker, User not known to the underlying aut
hentication module

[영어][완성][2벌식]
```

그림 35 :연구 결과 프로토 타잎 화면



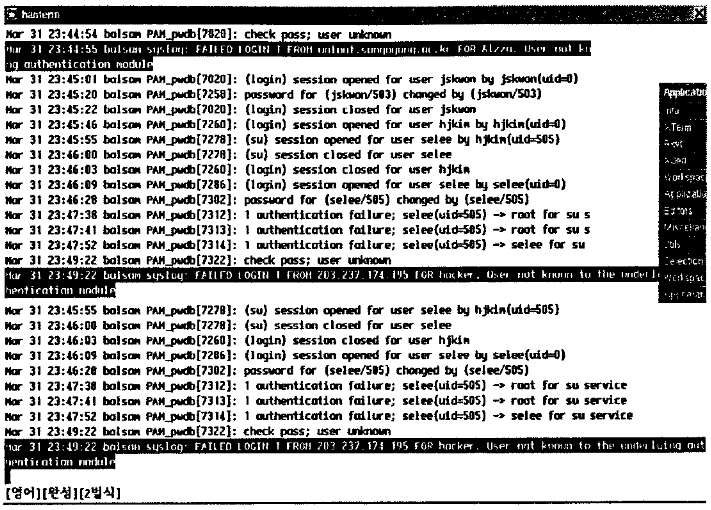
Secure.log의 모니터링

```
hanterm
May 24 12:28:12 balsom in.telnetd[11693]: connect from 203.237.174.184
May 24 12:34:47 balsom in.telnetd[11133]  connect from 203.237.174.184
May 24 13:26:44 balsom in.telnetd[12425]  connect from 203.237.174.14
May 24 13:29:24 balsom in.telnetd[12690]. connect from 203.237.174.184
May 24 13:48:54 balsom in.telnetd[13216]: connect from 203.237.174.184
May 24 17:30:11 balsom in.telnetd[16910]. connect from 203.237.174.184
May 24 18:54:00 balsom in.telnetd[20970]: connect from 203.237.174.188
May 25 16:45:19 balsom in.telnetd[1011]. connect from 203.237.168.133
May 25 16:46:25 balsom in.telnetd[1032]: connect from 203.237.168.133
May 25 16:46:50 balsom in.telnetd[1034]: connect from 203.237.168.133
May 25 16:52:27 balsom ipop3d[4857]: connect from 203.237.168.133
May 25 16:52:57 balsom ipop3d[4858]: connect from 203.237.168.133
May 25 16:52:58 balsom ipop3d[4859]: connect from 203.237.168.133
May 26 20:13:31 balsom in.telnetd[12313]: connect from 203.237.168.166
May 26 20:14:05 balsom in.telnetd[12318]: connect from 203.237.168.166
May 26 20:15:44 balsom in.telnetd[12323]: connect from 203.237.168.166
May 26 20:16:03 balsom in.telnetd[12329]. connect from 203.237.168.166
May 26 20:18:12 balsom in.telnetd[12356]: connect from 203.237.168.166
May 26 20:30:55 balsom in
May 26 20:31:19 balsom in
May 26 20:32:18 balsom in
[영어][완성][2벌식]
```

xferlog의 모니터링

```
hanterm
Wed May 27 20:11:20 1998 1 203.237.168.116 857 /home/neenee/public_html/index.html b _ o r ho
zon ftp 0 *
Thu May 28 14:54:40 1998 1 203.237.174.187 503471 /var/log/cron b _ o r webmaster ftp 0 *
Thu May 28 14:54:40 1998 1 203.237.174.187 89514 /var/log/cron.1 b _ o r webmaster ftp 0 *
Thu May 28 14:54:41 1998 1 203.237.174.187 1 /etc/passwd b _ o r webmaster ftp 0 *
Thu May 28 14:54:41 1998 1 203.237.174.187 1 /var/log/cron.3 b _ o r webmaster ftp 0 *
Thu May 28 14:54:42 1998 1 203.237.174.187 1 /var/log/cron.4 b _ o r webmaster ftp 0 *
Thu May 28 14:54:42 1998 1 203.237.174.187 2952 /var/log/dmesg b _ o r webmaster ftp 0 *
Thu May 28 14:54:42 1998 1 203.237.174.187 19008 /var/log/lastlog b _ o r webmaster ftp 0 *
Thu May 28 14:54:58 1998 1 203.237.174.187 13309 /var/log/secure b _ o r webmaster ftp 0 *
Thu May 28 14:54:59 1998 1 203.237.174.187 1257 /etc/passwd b _ o r webmaster ftp 0 *
Thu May 28 14:54:59 1998 1 203.237.174.187 1 /var/log/secure.2 b _ o r webmaster ftp 0 *
Thu May 28 14:54:59 1998 1 203.237.174.187 1 /var/log/secure.3 b _ o r webmaster ftp 0 *
Thu May 28 14:55:00 1998 1 203.237.174.187 1 /var/log/secure.4 b _ o r webmaster ftp 0 *
Thu May 28 14:55:00 1998 1 203.237.174.187 408 /var/log/sendmail.st b _ o r webmaster ftp 0 *
Thu May 28 14:55:01 1998 1 203.237.174.187 762 /var/log/spooler b _ o r webmaster ftp 0 *
Thu May 28 14:55:02 1998 1 203.237.174.187 0 /var/log/spooler.1 b _ o r webmaster ftp 0 *
Thu May 28 14:55:02 1998 1 203.237.174.187 192 /var/log/spooler.2 b _ o r webmaster ftp 0 *
Thu May 28 14:55:04 1998 1 203.237.174.187 16521 /var/log/wtmp b _ o r webmaster ftp 0 *
[영어][완성][2벌식]
```

그림 36 :연구결과 프로토 타잎 화면



```
Mar 31 23:44:54 balsam PAM_pwdb[7020]: check pass; user unknown
Mar 31 23:44:55 balsam syslog: FAILED LOGIN 1 FROM unlout.sungyung.ac.kr FOR Alzza. User not kn
ng authentication module
Mar 31 23:45:01 balsam PAM_pwdb[7020]: (login) session opened for user jskwon by jskwon(uid=0)
Mar 31 23:45:20 balsam PAM_pwdb[7258]: password for (jskwon/503) changed by (jskwon/503)
Mar 31 23:45:22 balsam PAM_pwdb[7020]: (login) session closed for user jskwon
Mar 31 23:45:46 balsam PAM_pwdb[7260]: (login) session opened for user hjkim by hjkim(uid=0)
Mar 31 23:45:55 balsam PAM_pwdb[7278]: (su) session opened for user selee by hjkim(uid=505)
Mar 31 23:46:00 balsam PAM_pwdb[7278]: (su) session closed for user selee
Mar 31 23:46:03 balsam PAM_pwdb[7260]: (login) session closed for user hjkim
Mar 31 23:46:09 balsam PAM_pwdb[7286]: (login) session opened for user selee by selee(uid=0)
Mar 31 23:46:28 balsam PAM_pwdb[7302]: password for (selee/505) changed by (selee/505)
Mar 31 23:47:38 balsam PAM_pwdb[7312]: 1 authentication failure; selee(uid=505) -> root for su s
Mar 31 23:47:41 balsam PAM_pwdb[7313]: 1 authentication failure; selee(uid=505) -> root for su s
Mar 31 23:47:52 balsam PAM_pwdb[7314]: 1 authentication failure; selee(uid=505) -> selee for su
Mar 31 23:49:22 balsam PAM_pwdb[7322]: check pass; user unknown
Mar 31 23:49:22 balsam syslog: FAILED LOGIN 1 FROM 203.237.174.195 FOR hacker. User not known to the underl
hentication module
Mar 31 23:45:55 balsam PAM_pwdb[7278]: (su) session opened for user selee by hjkim(uid=505)
Mar 31 23:46:00 balsam PAM_pwdb[7278]: (su) session closed for user selee
Mar 31 23:46:03 balsam PAM_pwdb[7260]: (login) session closed for user hjkim
Mar 31 23:46:09 balsam PAM_pwdb[7286]: (login) session opened for user selee by selee(uid=0)
Mar 31 23:46:28 balsam PAM_pwdb[7302]: password for (selee/505) changed by (selee/505)
Mar 31 23:47:38 balsam PAM_pwdb[7312]: 1 authentication failure; selee(uid=505) -> root for su service
Mar 31 23:47:41 balsam PAM_pwdb[7313]: 1 authentication failure; selee(uid=505) -> root for su service
Mar 31 23:47:52 balsam PAM_pwdb[7314]: 1 authentication failure; selee(uid=505) -> selee for su service
Mar 31 23:49:22 balsam PAM_pwdb[7322]: check pass; user unknown
Mar 31 23:49:22 balsam syslog: FAILED LOGIN 1 FROM 203.237.174.195 FOR hacker. User not known to the underlying aut
hentication module
```

그림 37 : 독립에이전트의 침입탐지과정

다음 perl script 코드는 위와 같은 시스템의 기본적인 침해행위시 보안관리자의 모니터상에 메시지와 경고음을 출력하는 perl script program 의 configuration file이다.

```
# Detector 프로그램의 configuration file
#

/INVALID|REPEATED|INCOMPLETE/                    echo=inverse, bell=3
# login 이나 패스워드변경과 같은 행위시 실행
/LOGIN/                echo=inverse, bell=3
/passwd/               echo=bold, bell=3
/ruserok/              echo=bold, bell=3

# Ignore this stuff
/sendmail/,/nntp/,/xntp|ntpd/,/faxspooler/ignore

# Report unusual tftp info
/tftpd.*(ncd|kfps|normal exit)/ignore
/tftpd/          echo,bell=3
```

```
# Kernel 과 관련된 행위시 메시지 출력 및 경고음
/(panic|halt|SunOS Release)/   echo=bold, bell
/file system full/         echo=bold, bell=3
/vmunix.*(at|on)/ignore
/vmunix/                   echo, bell

/fingerd.*(root|[Tt]ip|guest)/   echo,bell=3
/atkins/          echo=inverse,bell=3

/su:/             echo=bold,bell=4
/.*/              echo
```

제5절 향후 연구 과제

현재 국내에서는 침입탐지기술에 대한 연구가 활발하게 이루어지고 있다. 본 연구에서는 단일모듈의 침입탐지 시스템의 단점을 보안하고 기존의 중앙 집중식 구조의 침입 탐지 시스템의 장점을 수용하기 위해 모니터링 프로세스와 조정자를 가진 시스템 구조의 하이브리드한 침입 탐지 시스템을 제안하였다. 그리고 모델로만 제안되었던 단위행동 에이전트를 실제로 구현하여 보았다. 침입탐지모듈의 설계부분에서는 대체로 발생하는 침입패턴들이 원치 않는 근원지에서 오거나 특정서비스를 이용하거나 혹은 시스템에 과부하를 걸고 다른 일을 하는 패턴이 많은데 주안점을 둔 것으로 DB와 DB에 새로운 침입패턴들을 탑재시키는 부분과 들어오는 패턴들을 DB와 비교하여 침입인지를 판정하는 부분으로 구성된다. 그러므로 얼마나 많은 침입시나리오들이 데이터베이스에 구축되느냐가 중요한 문제이다.

제7장 연구개발목표 달성도 및 대외기여도

제1절 연구개발 내용

년 도	목 표	개 발 내 용
1차년도 (1997-1998)	개인 신용평가 모델(Financial Fraud Detection) 초기 알고리즘 개발 침입 탐지 시스템(Intrusion Detection System) 현장조사 및 데이터 수집	1. 개인 신용평가 모델 초기 알고리즘 개발 - 클러스터링 알고리즘을 이용한 개인 신용카드 데이터 수집 및 전처리 - 퍼지 전문가 시스템 구축 - 유전자 알고리즘을 이용한 퍼지 규칙 학습 알고리즘 구현 2. 침입 탐지 시스템(Intrusion Detection System) 현장조사 및 데이터 수집 - 현존 침입 탐지 시스템 분석 - 네트워크 데이터 수집 및 전처리 - Automated Profiler 개발
2차년도 (1998-1999)	개인 신용평가 모델(Financial Fraud Detection) 알고리즘 개발 새로운 네트워크 침입 탐지 시스템(Network-Based Intrusion Detection System) 프레임 웍 개발	1. 개인 신용평가 모델(Financial Fraud Detection) 알고리즘 개발 - 다양한 새 퍼지 멤버쉽 함수 개발 - 노이즈 데이터 처리를 위한 multi-objective fitness function 개발 - 다양한 변수의 자동 조절을 위한 committee members 개발 2. 새로운 침입 탐지 시스템(Network-Based Intrusion Detection System) 프레임 웍 개발 - 인간 면역시스템의 연구 - 새로운 침입 탐지 시스템 모델인 인공 면역 시스템 개발 - Negative Selection Algorithm 개발
3차년도 (1999-2000)	시스템 통합, 데스트	1. 개인 신용평가 모델(Financial Fraud Detection) 평가 - 실 데이터를 이용한 시스템 평가 2. 인공 면역 시스템 평가 및 추가 개발 - Negative Selection Algorithm 평가 - Clonal Selection Algorithm 개발

제2절 연구수행 내용, 결과 및 연구 기여도

3년에 걸친 본 연구의 연구수행 내용 및 결과는 다음과 같다. 우선 본 연구는 개인 신용 평가 시스템의 개발과 새로운 지능형 네트워크 침입 탐지기를 개발하는 것을 주요 목표로 연구가 수행되었다.

1. 지능형 개인 신용 평가 시스템의 개발

그동안의 많은 개인 신용 평가 시스템이 신용 스코어링 테이블, 통계적인 방법, 신경망 알고리즘 등을 이용해 왔고, 신용 평가 예측력을 높이기 위해 일반화에 관한 연구가 진행되어 왔다. 본 연구에서는 이러한 기존 방법들과는 다른 퍼지 규칙의 진화를 이용하여 지능형 개인 신용 평가 시스템을 개발하였다. 본 연구에서 개발된 시스템은 기존 신용 카드 고객들의 카드 트랜잭션 데이터를 분석하는 것으로, 새로운 신용 카드 고객들의 신용도를 예측한다. 본 시스템은 크게 퍼지 전문가 시스탬과 유전자 알고리즘을 이용한 퍼지 규칙 진화 시스템으로 이루어져 있다.

○ 퍼지 전문가 시스템에서는

- 다양한 클러스터링 알고리즘을 이용한 퍼지 규칙의 정의하고, 각기 다른 클러스터링 알고리즘에 따른 신용 평가 예측도와 신용 평가 퍼지 규칙의 이해도를 평가
- 다양하고 새롭게 정의된 퍼지 멤버십 함수를 이용하여 퍼지 규칙을 정의하고, 각기 다른 멤버쉽 함수에 따른 신용 평가 예측도와 신용 평가 퍼지 규칙의 이해도를 평가

이상의 연구 내용들이 수행되었고,

○ 유전자 알고리즘을 이용한 퍼지 규칙 진화 시스템에서는

- 유전자 프로그래밍의 트리 구조를 이용하여 좀더 다양한 퍼지 규칙이 진화되도록하나, 유전자 프로그래밍의 취약점인 교차(Crossover)오퍼레이터에 의한 진화 과정의 분열(disruption)을 막기위한 문법적으로 유사한 교차점을 찾아 교차하는 새로운 교차(Crossover)오퍼레이터의 개발,
- 개발된 신용 평가 시스템이 하나 이상의 다중 목표를 만족하도록 하는 적합함수(fitness function)의 구현: 이에 구현된 다중 목표 적합 함수는
- 실 데이터가 갖고있는 노이즈를 효과적으로 처리하도록하는 적합함수
- 시스템의 판단 결과를 시스템 사용자가 이해 하기 쉽도록 짧은 퍼지 규칙

으로 진화하도록하는 적합함수

- 진화된 퍼지 규칙이 불량 신용을 갖는 고객을 탐지 예측함과 동시에, 우량 신용 고객을 불량 신용고객으로 잘못 판단하지 않도록 예측하도록 하는 적합함수로 구성되어, 시스템이 서로 상충될 수 있는 이상의 목표들을 모두 동시에 적합한 선에서 만족하도록 개발
- 유전자 프로그래밍의 진화 과정이 불량 신용 고객의 패턴을 구성하는 여러개의 퍼지 규칙을 모두 찾을 수 있도록 유도하는 nested genetic search 의 구현
- 많은 변수들의 서로 다른 조합에 따라 다양한 시스템의 신용도판별능력과 진화된 퍼지 규칙의 이해도가 달라지고, 이들 다양한 결과중 가장 이상적인 시스템의 판단결과를 선택하기 위한 위원회 결정(committe-decision) 의 구현

이상의 연구 내용들이 수행되었다.

개발된 시스템은 Lloys/TSB가 제공한 보험 고객 데이터와 국내 카드 회사에서 제공한 개인 신용 카드 고객 데이터에 적용하여 시스템의 성능이 평가되었으며, 노이즈가 많은 실데이타에도 불구하고, 높은 신용평가 예측력과 이해도를 보였다.

2. 새로운 네트워크 침입 탐지 시스템

본 연구에서 제안한 네트워크 침입 탐지 시스템은 불법 네트웍 침입자의 네트웍 침입시 정상적인 네트웍 트래픽에서는 관찰되지 않는 다른 패턴의 네트웍 트래픽을 탐지 하여 네트웍 관리자에게 불법 침입자의 침입 가능성을 자동으로 알리는 시스템이다. 이러한 지능적 네트워크 침입 탐지 시스템의 연구 개발은 비교적 새로운 연구 분야로서, 대부분의 시도되고 있는 연구들은 중앙 집중적인 통계적 방법들을 이용하고 있다. 본 연구에서는 수행된 연구로는

○ 전혀 새로운 네트워크 침입 탐지 시스템인 인공 면역 시스템을 제안

- 현존하고 있는 네트워크 침입 탐지 시스템들의 광범위한 문헌 조사를 통해, 현존하고 있는 대부분의 중앙 집중적인 통계적 방법들들의 한계들을지적
- 이러한 한계들을 극복하기 위한 전혀 새로운 접근법의 제안을 위한 인간 면역 시스템을 연구
- 인간 면역 시스템의 연구를 통해 인간 면역 시스템만이 갖고 있는 특징

들을 네트워크 침입 탐지 시스템에 적용할 수 있는지의 여부와 그러한 적용이 이미 본 연구에서 지적된 종전의 중앙 집중적인 네트워크 침입 탐지 시스템들의 한계를 극복할 수 있는지를 연구
- 이상의 연구를 통해 전혀 새로운 네트워크 침입 탐지 시스템인 인공 면역 시스템을 제안

○ 인공 면역 시스템을 구성하는 주 네트워크 침입 탐지 시스템의 구현
- 정상적인 네트워크 트래픽과 불법 네트워크 침입자의 네트워크 침입시의 네트워크 트래픽의 차이를 구분할 수 있는 네트워크 트래픽 특징들을 선별
- 선별된 네트워크 트래픽 특징들을 모아진 네트워크 패킷에서 추출하여 트래픽의 profile을 만드는 자동네트워크 트래픽 profiling 모듈 구현
- 정상적인 네트워크 트래픽 패턴을 불법 네트워크 침입 패턴으로 잘못 인식하는 것을 막기위한 부정적 선택(Negative Selection) 알고리즘 구현
- 구현된 부정적 선택(Negative Selection) 알고리즘을 실네트워크 트래픽 데이터에 적용해보고 그의 한계점 연구

○ 인공 면역 시스템을 구성하는 부 네트워크 침입 탐지 시스템의 구현
- 불법 네트워크 침입이 이미 알려진 네트워크 트래픽에서 그들의 패턴을 추출하여, 미래에 동일하거나 유사한 네트워크 침입 발생시 그의 탐지를 실시간내에 할 수 있도록 하는 복제적 선택(clonal selection) 알고리즘의 구현
- 복제적 선택(clonal selection) 알고리즘의 수행중 생성되는 탐지기(detector)의 일반화(generalisation)를 높이기에 적합한 탐지기 표현에 관한 연구
- 부정적 선택(Negative Selection) 알고리즘을 복제적 선택(clonal selection) 알고리즘의 부분으로 통합시키는 것으로 부정적 선택(Negative Selection) 알고리즘의 본래의 목적울 달성하면서, 긴 computaional time을 요구하는 부정적 선택(Negative Selection) 알고리즘의 한계를 극복

이상과 같이 본 연구는 기존의 네트워크 침입 탐지 시스템을 유사하게 구현하거나 확장하는 식의 현존기술의 국내화 보다는 연구과정중 인식된 현존하는 네트워크 침입 탐지 시스템들의 근본적인 문제들을 해결하는데 그 중점을 두었다. 따라서, 국내-외 최초로 통합적인 모델로서의 인공 면역 시스템을 네트워크 침입 탐지 시스템의 새로운 대안으로 제시하였으며, 그에 대한 기본적인 부분의 구현과 평가가 실시되었다.

3. 적용분야

○ 전자상거래에서의 응용

Fraud Detection System 및 IDS 기술은 기술은 전자상거래의 쇼핑몰 인프라에 매우 중요 부분에 적용되어 인터넷의 패킷 정보 뿐 아니라, 사용자의 행위, 통계적인 시스템의 변화 등을 고려하여 오용 및 침입여부를 판단하게 되고, 이미 설정된 대응 정책에 따라 적절한 대응 행위를 취하게 된다. 일반적으로 사기 및 침입의 형태는 쇼핑몰의 리소스를 위조 및 서비스 거부행위, 정보를 탈취하거나 파괴, 변조하는 행위 등으로 나타나게 되는 데, 대부분의 경우 복합적으로 이루어진다. 인터넷 상점 내부 시스템에 침입하여 개인 정보를 습득하여 네트워크를 통해 자신의 시스템으로 이동시키고 자신의 침입사실을 은닉하기 위해서 시스템 파일을 변조하거나 삭제하는 경우가 그 예이다.

Fraud Detection System의 경우, 인터넷상의 금융거래의 유형을 파악하여, 소비자의 유형을 분석할 수 있으며, 침입탐지 시스템은 침입 상황에 대한 실시간 경보, 침입 통계 작성 및 탐지된 침입에 대응하는 기능을 가지며, 이를 통해서 안전한 전자상거래를 도모할 수 있다. 인터넷은 정보의 위변조 및 탈취 등의 침입에 항상 노출되어 있으므로 사실상 전자상거래는 범죄의 위험속에서 실현되는 행위로 볼 수 있기 때문에 전자상거래의 가장 중요한 보안기술이라 할 수있다.

○ 금융 분야(은행, 카드, 보험 등)의 응용

본 연구에서 개발된 신용평가 시스템은 신용카드, 전자상거래, 멤버쉽을 이용한 기관거래에 사용될 수 있다. 이는 사용자의 고정적인 데이터, 예를 들면 소득이나 나이, 학력, 성별등과 같은 개인정보가 개인 프라버시 보호라는 명목 때문에 얻어지기 힘들어지기 때문에 각 소비자의 행동 패턴을 기초자료로 신용을 평가할 수밖에 없게 된다. 본 연구에서 개발된 신용평가 시스템은 이러한 용도로 사용될 수 있으면 국내 금융기관들, 즉, 은행, 카드, 보험, 증권, 투신등의 고객서비스 강화에 이바지 할 수 있다.

○ 기타 분야에서의 응용

금융기관의 중요 데이터 서버들을 사기감지 및 보호하기 위해 설치된 시스템은 시스템은 예금 관련 데이터베이스의 감시와 고객 정보의 유출 방지, 웹서버 등 사내

전산자원의 불법사용 여부를 판단하게 되고, 설정된 대응 정책에 따라 적절한 대응을 취하게 된다.

대학 및 기타 교육기관의 전산자원을 보호하기 위해 설치된 침입탐지 시스템은 학적 데이터베이스의 감시, 연구자료들에 대한 불법적인 유출 방지, 학내 시스템에 대한 유해 행위 등을 탐지하고, 탐지 결과에 대한 적절한 대응을 취하게 된다. 이외에도 침입탐지 기술은 서비스 거부공격, 금융사기, 바이러스 등의 수많은 정보 침해 유형에 적절히 대응하기 위해서 기타 여러 분야에서 응용될 수 있으며, 건전한 사이버 문화를 형성하는 데, 기반이 될 것이다.

제8장 연구개발결과 및 실적

제1절 국제공동연구개발 추진 실적

○ 특허 출원

특허명	출원번호	출원일	특허신청자
암호화 송신 장치 및 방법	99-10816	99/3/29	최종욱, 이원하, 멀티정보

○ 논문

(1) Kim, J. and Bentley, P. J. (1999). "Negative Selection and Niching by an Artificial Immune System for Network Intrusion Detection" *A late-breaking paper, Genetic and Evolutionary Computation Conference (GECCO '99), Orlando, Florida, July 13-17 .*

(2) Kim, J. and Bentley, P., (1999), "The Human Immune System and Network Intrusion Detection", *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT '99), Aachen, Germany, September 13- 19.*

(3) Kim, J. and Bentley, P., (1999), "The Artificial Immune Model for Network Intrusion Detection", *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT'99), Aachen, Germany, September 13- 19.*

(4) Kim, J. (1999), "The Artificial Immune System for Network Intrusion Detection", *Phd Student Workshop, Genetic and Evolutionary Computation Conference, Orlando, Florida. July 13-17 (GECCO-99).*

(5) Bentley, P. J. (2000). "Evolutionary, my dear Watson: Investigating Committee-based Evolution of Fuzzy Rules for the Detection of Suspicious Insurance Claims" the second *Genetic and Evolutionary Computation Conference* (GECCO 2000), July 8-12, Las Vegas, Nevada, USA.

(6) Bentley, P. J. (2000). "Evolving Fuzzy Detectives: An Investigation into the Evolution of Fuzzy Rules", Chapter in Suzuki, Roy, Ovasks, Furuhashi and Dote

(Eds), *Soft Computing in Industrial Applications*. Springer Verlag London Ltd. ISBN 1-85233-239-X.

(7) Seungwon Shin, Jonguk Choi, "Enhancement of Travel Time Forecasting Accuracy, Based on Wavelet Transformation and Neural Network," submitted to *Transportation Research Part C: Emergency Technology,* 1998. 11.

(8) 신우철, 최종욱, "Blackboard 기반의 침입탐지 시스템 개발," 정보보안학회지, 1999. 5 (심사중)

(9) 김회준, 최종욱, "에이전트 기반의 침입탐지 시스템 구현," 정보처리학회, 1999, 5 (심사중)

(10) 신승원, 박종진, 최종욱, "시계열 분석 방법을 이용한 암호화 알고리즘의 주기성 평가,"" 제9회 통신 정보합동학술대회 논문집, 1999. 4.22

(11) 정길호, 김정원, 최종욱, "인간 면역 체계와 네트워크 침입 탐지," 99 춘계공동학술대회-지식경영과 지식공학, 한국지능정보시스템학회, 한양대학교 1999. 6. 4.

(12) 정길호, 김정원, 최종욱, "인공면역 모델을 이용한 네트워크 침입 탐지," 99 춘계공동학술대회-지식경영과 지식공학, 한국지능정보시스템학회, 한양대학교 1999. 6. 4.

(13) 신우철, 최종욱, "Blackboard 기반의 침입탐지 시스템 개발," 한국경영정보학회 '99춘계학술대회, 6월 5일 광운대학교

(14) Bentley, P. J., Kim, J., Jung, G. and Choi, J., "Fuzzy Darwinian Detection of Credit Card Fraud", 한국정보처리학회, 대전 한국 정보 통신 대학원 대학교, 2000. 10. 13.

(15) Bentley, P. J., Kim, J., and Choi, J., "Negative Selection within an Artificial Immune System for Network Intrusion Detection", 한국정보처리학회, 대전 한국 정보 통신 대학원 대학교, 2000. 10. 13.

(16) Bentley, P. J., Kim, J., Jung, G. and Choi, J., "The Investigation of Fuzzy Darwinian Detection of Korean Credit Card Fraud", 한국 통신 정보 보호 학회지, 2000. 9. (심사중)

제9장 연구개발결과의 활용계획

제1절 추가연구의 필요성

1. 지능형 개인 신용 평가 시스템

본 연구에서 개발된 지능형 개인 신용 평가 시스템은 국내 신용 카드 회사의 고객 트랜잭션 데이터에 적용해본 결과 만족할만한 높은 예측력과 함께, 시스템의 예측 결과를 짧은 퍼지 규칙으로 표현하여 이용자로 하여금 그 결과를 이해하게 함으로써 시스템의 신용도를 높였다. 따라서, 그간의 연구 과제였던 시스템의 일반화 문제와 신뢰도를 해결하였으나, 개발된 시스템을 여러 금융기관 현장에서 바로 사용하기 위해서는 다음과 같은 추가적인 연구가 요구된다.

- 엄청나게 불어나는 신용카드 트랜잭션 데이터의 처리: 전자 상거래를 비롯한 온라인 쇼핑 형태의 증가로 인한 신용 카드 사용의 폭발적인 증가는 신용 카드 트랜잭션의 증가를 가져오고 있으며, 이에 따라 개발된 시스템이 엄청나게 불어나는 신용카드 트랜잭션 데이터를 요구되는 시간 안에 빠르게 처리할 수 있는 기능을 갖도록 보안되어야 한다. 현재 개발된 시스템은 최대 10000개의 트랜잭션 데이터를 처리하였으나, 이를 위해 요구된 시스템의 퍼지규칙진화에는 적지 않은 시간이 소요되었다. 특히, 이 시스템이 온라인 쇼핑몰에서의 실시간으로 신용카드 사기 등을 탐지하기 위해서는 현재의 중앙 집중적인 시스템의 구조를 분산형이나 병렬형으로 수정, 여러 대의 신용평가 서버가 동시에 새롭게 빠른속도로 입력되는 많은양의 데이터들에 대한 퍼지규칙의 진화를 수행하게끔 함으로써, 요구되는 실시간 탐지가 가능하도록 해야할 것이다. 본 연구에서는 이러한 문제를 인식하고, 개발 초기 단계에 분산형의 구현이 보다 자연스럽고 쉬운 유전자 알고리즘이 채택되었었다.

- 현재 사용되고 있는 전자상거래 및 온라인 쇼핑을 위한 시스템과의 통합: 본 연구에서 개발된 시스템을 포함한 기존의 개인신용평가 시스템이 독립적인 서버로 존재하여 배치처리로 그 판단결과를 통보하는 환경을 전제로 개발되었다. 따라서 개발된 시스템이 전자상거래등의 온라인 쇼핑몰에서의 사용을 위해서는 요구되므로 현존하고 있는 이들 시스템들과 연동될 수 있도록 적합한 시스템 구조의 설계가 필요하다 하겠다.

2. 네트워크 침입 탐지 시스템

본 연구에서는 기존의 중앙 집중형 네트워크 침입 탐지 시스템의 한계점들을 지적하고, 그를 보완하기 위한 분산형 네트워크 침입 탐지 시스템으로서 인공 면역 시스템을 제안하고, 제안된 시스템의 기본적인 부분들을 구현 평가해 보았다. 따라서, 개발된 시스템을 확장 보안하기 위해서는 다음과 같은 추가적인 연구가 필요하다.

- 개발된 인공면역모듈들의 통합 평가: 본 연구에서는 새로운 분산형 네트워크 침입 탐지 시스템 모델인 인공면역시스템을 제안하고 각 모듈들을 구현 평가해보았으나, 각 모듈들을 통합하여 실 네트워크 환경에서 평가까지는 이루어지지는 못했다. 인공면역시스템의 제안 동기가 분산형 네트워크 침입 탐지 시스템의 개발이었던 만큼, 개발된 기본 모듈들의 통합과 평가는 필수적이라 하겠다.

- 시간 패턴을 포함하는 네트워크 트래픽 프로화일 구축: 본 연구에서 개발된 네트워크 트래픽 프로화일링 모듈은 하나의 네트워크 커넥션동안 관찰된 네트워크 트래픽에 대한 프로화일만울 구축하였으나, 최근에는 네트워크 트래픽 패턴은 일정시간 단위에 따라 정규적으로 변화하는 것을 주목, 이러한 패턴들을 네트워크 트래픽 프로화일에 포함시키는 연구들이 활발히 진행되고 있다. 프로화일 전달해줄수 있는 정보의 질과 양에 따라 인공면역시스템의 탐지 성능이 크게 좌우되는 만큼 이에대한 향후 연구가 요구된다.

- 기존의 네트워크 침입 탐지 시스템과의 통합: 현재 상용화되어 사용되고 있는 대부분의 네트워크 침입 탐지 시스템은 이미 알려진 네트워크 침입 흔적만을 탐지하는 시스템이다. 이와는 달리 본 연구에서 제안된 인공면역시스템은 이전에 알려지지 않은 새로운 네트워크 침입 탐지와 알려진 네트워크 침입이라도 수시로 변화되는 네트워크 환경을 고려하여 함께 변화되는 네트워크 침입 흔적을 탐지하는 시스템이다. 따라서, 본 연구에서 제안된 인공면역시스템이 현존하는 시스템과 함께 연동되어 각기 다른 형태의 네트워크 침입들이 각 시스템에서 탐지된 침입 흔적들의 공유, 통합을 통해서 즉각적으로 탐지될 수 있도록 하는 연구가 필요하다.

- 현존하는 네트워크 보안 시스템들과의 통합: 네트워크 침입 탐지는 독립적인 문제가 아니라 네트워크 보안이라는 커다란 분야에서의 한 부분으로 인식하고, 그의 궁극적 문제인 네트워크 보안을 극대화할 수 있도록 현재 네트워크 보안을 위해 사용되고 있는 방화벽, 암호화 등과 효과적인 통합 방법론에 대한 연구가 요구된다.
뿐만 아니라, 개인용 신용 평가 시스템과 네트워크 침입 탐지 시스템간의 통합 역

시 중요한 향후 연구과제이다. 본 연구의 본래의 궁극적 목표는 금융안전이 완전히 보장된 온라인 경제를 구축하기 위한 방안으로서 지능형 개인용 신용 평가 시스템과 네트워크 침입 탐지 시스템의 통합적인 개발을 목표로 하였으나, 예산 감축등으로 인한 연구 범위의 축소로 이러한 목표가 달성되지는 못하였다. 그러나, 본래 제시된 온라인 경제의 보안 문제는 현재도 풀어야 할 중요한 과제로 남아있으며, 이를 위한 본 연구에서 개발된 두 시스템의 통합은 향후 온라인 경제의 보안 문제에 크게 기여할 수 있을 것으로 보여진다.

제2절 타연구에의 응용

본 연구에서 개발된 개인용 신용 평가를 위한 퍼지 규칙의 진화 알고리즘과 퍼지 규칙 전문가 시스템, 네트워크 침입 탐지를 위한 인공 면역 시스템은 다음과 같은 타 연구에도 응용될 수 있다.

- 각종 사기 범죄 탐지에의 응용: 본 연구에서 개발된 퍼지 규칙의 진화에 의한 퍼지 규칙 전문가 시스템은 불량 신용을 갖는 고객의 적발에 그 초점을 두었으나, 동일한 기술을 각종 다른 온라인 사기범죄 등의 적발에 이용할 수 있다. 예를 들어, 온라인 상에서 타인의 신용카드 정보를 가로채어 타인의 신용카드를 이용하는 자, 타인의 휴대폰 정보를 가로채어 타인의 휴대폰 번호를 이용하는 경우에서부터 크게는 각 금융기관의 고객들의 계좌 모니터링에 응용하여 범죄자들의 돈세탁등을 적발할 수 있는 시스템을 개발하는 것도 가능하다.

- 기업 트랜잭션 모니터링에의 응용: 뿐만 아니라 동일한 기술을 기업내의 전산화된 시스템의 트랜잭션 모니터링에 이용하여 새로운 비즈니스 기회를 찾아내는데 응용할 수 있다. 예를 들어, 각 고객들의 구매패턴 구분한다.

- 각종 분산 시스템 오류 탐지에의 응용: 특히 본 연구에서 제안된 인공면역시스템은 각종 분산형 시스템의 오류 탐지에도 응용될 수 있다. 예를 들어, 자동화된 공장 라인의 오류 탐지, 네트워크 오류 탐지와 유지보수 등이 응용될 수 있는 분야이다.

- 오류 모빌 에이전트 탐지에의 응용: 최근 들어 활발히 연구되고 있는 새로운 정보통신 핵심분야 중 하나인 모빌 에이전트는 이용자의 개입없이 프로그램 자체가 인터넷 상을 돌아다니면서, 이용자가 필요한 정보등을 자동으로 수집 분석하여다 주는 일등을 수행한다. 이러한 모빌 에이전트의 이용은 그 유용성과 함께 오류 모빌 에이전트가 가져올 수 있는 보안상의 문제를 심각하게 갖고 있다. 분산적인 탐

지기들에 의한 비정상적인 패턴을 찾아내는 인공면역시스템은 오류 모빌 에이전트나 허가되지 않은 모빌 에이전트의 시스템 접근의 탐지등에 응용될 수 있다.

제3절 기업화 추진방향

개발된 연구 결과는 국내의 산업현장에서 사용 가능한 것들이다. 특히, 신용평가시스템은 금융기관이나 주택업체, 회원 전용 스포츠업체, 인터넷 거래 업체 등에서 개인신용평가는 물론, 업체 신용평가에 직접 적용할 수 있을 것으로 본다. Frauds Detection 기술은 컴퓨터 보안 업체와 통신회사, 컴퓨터회사, 서비스 기업 등에서 불량 고객으로 인한 손해를 줄일 수 있는 기술이며 외부침입탐지 기술은 컴퓨터 보안 업체와 일반 관공서, 사기업체의 보안에 적용할 수 있는 기술이다.

공동으로 참여한 (주)Security Technologes International (www.stitec.com)에서는 보안 시스템의 개발에 본 연구에서 개발된 기술을 상당부분 수용하고 있어 상용화 측면에서 상당한 성과가 있었다. 현재 이 회사에서는 암호화 관련 서비스를 하고 있으나 금융 솔루션 제품과 기타 제품(J/LOCK, J/CAS, J/SSLOCK, J/SecureSession)에 개발된 기술을 채용하고 있다.

제10장 Reference

Abe, S. and Lan, M. S., "A Classifier using Fuzzy Rules Extracted Directly from Numerical Data", Proceeding of 2nd IEEE International Conference on Fuzzy Systems, San Francisco, Calif, pp.1191-1198, 1993.

Anderson J.P. "Computer security threat monitoring and surveillance", Technical Report, James P. Anderson & Co. April, 1980.

Anderson, D., "Safeguard Final Report: Detecting Unusual Program Behaviour Using the NIDES Statistical Component", Technical Report, Computer Science Laboratory, SRI International, Menlo Park, CA, December 1993.

Anderson, J. P., "Computer Security Threat Monitoring and Surveillance", Technical Report, James P. Anderson Co., Fort Washington, PA, April, 1980.

Anderson, Valdes. SAFEGUARD Final Report: Detecting Unusual Program Behavior Using the NIDES Statistical Component http://www.csl.sri.com/nides/index5.html

Balasubramaniyan, J. S. et al., "Software Agents for Intrusion Detection", Department of Computer Sciences, Purdue University, 1997.

Balasubramaniyan, J. S. et al., 1998, "An Architecture for Intrusion Detection using Autonomous Agents", Department of Computer Sciences, Purdue University, Available at http://www.cs.purdue..edu/coast/coast-library.html

Bentley, P. J. & Wakefield, J. P. "Hierarchical Crossover in Genetic Algorithms" In *Proceedings of the 1st On-line Workshop on Soft Computing (WSC1),* (pp. 37-42), Nagoya University, Japan, 1996.

Bentley, P. J., "Evolutionary, my dear Watson-Investigating Committe-based Evolution of Fuzzy Rules for the Detection of Suspicious Insurance Claims", in *Proceeding of the second Genetic and Evolutionary Computation Conference (GECCO 2000),* July 8-12, Las Vegas, Nevada, USA.

Bentley, P. J., "Evolving Fuzzy Detectives: An Investigation into the Evolution of Fuzzy Rules". Chapter in Suzuki, Roy, Ovasks, Furuhashi and Dote (Eds), *Soft Computing in Industrial Applications*. Springer Verlag London Ltd. ISBN 1-85233-239-X.

Bentley, P. J., and Wakefield, J. P, Finding acceptable solutions in the Pareto-Optimal Ranges using multiobjective genetic algorithms., Chawdhry, P. K., Roy, R., and Pant, R.K., (eds) Soft Computing in Engineering Design and Manufacturing, Springer Verlag London Limited, Part 5, pp.231-240.

Bentley, P. J., *Generic Evolutionary Desion of Solid Objects using a Genetic Algorithm*, PhD Thesis, Division of Computing and Control Systems, The University of Huddersfield, 1997.

Beranek, W. and W. Taylor, "Credit-Scoring Models and the Cut-off Point A Simplification", *Decision Science*, vol. 7, 1976.

Bezdek and Pal, "*Fuzzy Models for Pattern Recognition*", IEEE Press, New York, 1992.

Bishop, M., "A Standard Audit Trail Format", *Proceeding of the 1995 National Information Systems Security Conference*, Baltimore, Maryland, pp. 136-145, October 10-13, 1995.

Bishop, M., Wee, C., Frank, J., "Goal Oriented Auditing and Logging." Submitted to *IEEE Transactions on Computing Systems*, 1996.

Booker, L. B., Goldberg, D. E., and Holland, J. H., "Classifier Systems and Genetic Algorithms", *Artificial Intelligence*, Vol.40, No.1-3, pp.235-282, 1989.

Bunn, D., "Forecasting with more than one model." *Journal of Forecasting* v8, pp. 161-166, 1989.

Carbonell, J. G., R. S. Michalski, and T. M. Mitchell, "An Overview of Machine Learning." In R. S. Michalski, J. G., Carbonell, and T. M. Mitchell (Ed.), Machine Learning : An Artificial Intelligence Approach. Tioga, Palo Alto, Calif., 1983.

Carlberg, K.(SAIC), personal communication., Dec.1998.

Carter, C. and Catlett, J., "Assessing Credit Card Application Using Machine Learning", *IEEE Expert*, pp.71-79, Fall, 1987.

Carter, C., and J. Catlett, "Assessing Credit Card Applications using Machine Learning", *IEEE EXPERT*, pp.71-pp.79, Fall 1987

Chiu, S., "Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification", in *Fuzzy Information Engineering*, (Ed) Dubois, Prade, Yanger Wiley, 1997.

Choi, H. Y., "Application of Neural Network Technique for Developing a Credit Evaluation System", *Thesis for degree of M.S, Dept. of Applied Computer Science, Graduate School of Management and Information Science, Hankuk University of Foreign Studies*, Feb, 1995.

Chung, F. L., and Lee, T., "A Fuzzy k-nearest Neighbour Algorithm", *IEEE Transactions on System, Man, and Cybernetics*, Vol. 15, pp.580-585.

Chung, H. M. and M. S. Silver, "Rule-Based Expert Systems and Linear Models: An Empirical comparison of Learning-By-Examples Methods", *Decision Sciences*, Vol. 23, pp.687-707, 1992.

Continuous Assessment of a Unix Configuration: Integrating Intrusion Detection and Configuration Analysis. In Proceedings of the ISOC' 97 Symposium on Network and Distributed System Security.San Diego, California, 1997. http://www.info.fundp.ac.be/~amo/publications.html

Crosbie M.and Spafford G., "Defending a Computer System using Autonomous Agents", In Proceedings of the 18th NISSC Conference, October 1995.

Crosbie, M. and Spafford, E. H., "Applying Genetic Programming to Intrusion

Detection", *Proceeding of AAAI Fall Symposium on Genetic Programming*, pp.1-8, Nov 1995.

Crosbie, M., and Spafford, E. H., "Active Defence of a Computer System Using Autonomous Agents", Department of Computer Sciences, Purdue University, CSD-TR-95-008, 1995.

Crosbie, M., and Spafford, E. H., "Defending a Computer System Using Autonomous Agents", Department of Computer Sciences, Purdue University, CSD-TR-95-022, 1994.

D'haeseleer., P., "An Overview of the Immune System" Available at http://www.cs.unm.edu/~steveah/imm-html/immune-system.html

Dasarathy, B.V., "Nosing Around the Neighborhood: A New System Structure and Classification Rule for Recognition in Partially Exposed Environments." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-2, No. 1, 67-71, 1980.

Dasgupta, D., "An Overview of Artificial Immune Systems and Their Applications", In Dasgupta, D. (editor). *Artificial Immune Systems and Their Applications*, Berlin: Springer-Verlag. Pages 3-21, 1998.

Dasgupta, D.; Attoch-Okine, N., "Immunity-Based Systems: A Survey", *Proceeding of the IEEE International Conference on Systems, Man and Cybernetics*, Orlando, October, 1997.

De Jong, K. A., Spears, W. M., and Gordon, D. F., "Using Genetic Algorithms for Concept Learning", *Machine Learning*, Vol.13, No.2/3, pp.161-188, 1993

Deb, K. and Goldberg, D. E., "An investigation of niches and species formation in genetic function optimization", *Proceeding of the 3rd International Conference on Genetic Algorithms*, pp.42-50, Fairfax, VA, Morgan Kaufmann, 1989.

Deber, H., Becker, M., and Siboni, D., "A Neural Network Component for an

Intrusion Detection System", *Proceeding of IEEE Symposium of Res. Security, Privacy*, Oakland, CA, pp.240-258, May 1992.

Decker, K. M. and Focardi, S., "Technology Overview: A Report on Data Mining", Technical Report, CSCS-ETH, Swiss Scientific Computing Centre, 1995.

Denning D.E.,IDES-"an Intrusion Detection Model.", IEEE Trans. Software Engineering. February 1987

Denning, D. E., "An Intrusion-Detection Model", *IEEE Transactions on Software Engineering*, Vol.SE-13, No.2, pp.222-232, February 1987.

Dhaeseleer, P. et al, "A Distributed Approach to Anomaly Detection", *ACM Transactions on Information System Security.*, 1997. Available at http://www.cs.unm.edu/~patrik

Distributed Audit Trail Analysis.In Proceedings of the ISOC '95 Symposium on Network and Distributed Systems Security. San Diego, California, February 1995. http://www.info.fundp.ac.be/~amo/publications.html

Doak, J., "An Evaluation of Search Algorithms for Feature Selection", 1994. Available at http://www.lanl.gov/users/u112295/public_html/

Douglas, L. R., Collins, E., Scofield, C., and Ghosh, S., "Risk Assessment of Mortgage Applications with a Neural Network System:an Update as the Test Portfolio Ages", Proceeding of IJCNN, 2, 1990.

Douglas, R. L., E. Collins, C. Schfield, and S. Ghosh, "Risk Assessment of Mortgage Applications with Neural Network systems: An Update as the Test Portfolio Ages", *Proceedings of IJCNN 1990*, Vol.Ⅱ, pp.479-pp.482, 1990.

Dungan and Chandlers, "Auditor: a Microcomputer-Based Expert System to Support Auditors in the Field", Expert System, pp.210-224, 1985.

Dungan, *"A Model of an Audit Judgement in the Form of an Expert System"*, Phd Thesis, Dept of Accounting, University of Illinois at Urbana-Champaign, 1982.

Dungan, C. W. and J. S. Chandlers, “Auditor : A microcomputer-based expert system to support auditors in the field”, *Expert System(October 1985)*, pp.210-224.

Dungan, C. W., “A model of an Audit Judgement in the Form of an Expert System”, *Ph D., Dissertation, Dept. of Accounting, University of Illinois at Unbana-Champaign(December, 1982)*

Dutta, S. and S. Sheckhar, "Bond Rating : A Non-Conservative Application of Neural Networks." *Proceedings of the IEEE International Conference on Neural Networks*, San Diego, CA., pp.443-pp.450, 1988.

Eberhart, "Computational Intelligence PC Tools", Academic Press, London, 1996.

Fahlman, S. E., "Faster-Learning Variations on Back-Propagation : An Empirical Study." In D. Touretsky, G. Hinton & T. Sejnowski, eds., *Proceedings of the 1988 Connectionist Models Summer School.* San Mateo, CA : Morgan Kaufmann, pp. 38-51.

Farmer, J. D.; Packard, N. H.; Perelson, A. S., "The Immune System, Adaptation and Machine Learning", *Physica* 22D, pp.182-204, 1986.

Fausett, L., “Fundamentals of Neural Networks, *Architectures, Algorithms, and Applications"*, Prentice Hall, 1994

Fayyad, U. M., and Irani, K. B., Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *Proceeding of The Thirteenth International Joint Conference on Artificial Intelligence*, pp.1022-1027, 1993.

Flockhar, I. W., "GA-MINER: Parallel DataMining with Hierarchical Genetic

Algorithms" Final Report, EPCC-AIKMS-GA-MINER-REPORT 1.0, Edinburgh Parallel Computing Center, 1995.

Forrest S., Hofmeyr S. A., Somayaji A., and Longstaff T. A., "A Sense of Self for Unix Proccesses.", In Proceedings of the 1996 IEEE Symposium of Security and Privacy, pp120-128, 1996.

Forrest, S. et al, "Self-Nonself Discrimination in a Computer", *Proceeding of 1994 IEEE Symposium on Research in Security and Privacy,* Los Alamos, CA: IEEE Computer Society Press, 1994.

Forrest, S. et al, "Using Genetic Algorithms to Explore Pattern Recognition in the Immune System", *Evolutionary Computation*, 1(3), 191-211, 1993.

Forrest, S. et al., "A Sense of Self for Unix processes", *Proceedings of 1996 IEEE Symposium on Computer Security and Privacy*, Los Alamos, CA, pp.120-128, 1996.

Forrest, S.; Hofmeyr, S; Somayaji, A, "Computer Immunology", *Communications of the ACM*, Vol.40, No.10, pp.88-96, 1997.

Fox K.L., Henning R.R, Reed J.H. and Simonian R.P.,"A Neural Network Approach Towards Intrusion Detection.", Technical Report, Government Info. Systems Division, Harris Corp., July 1990

Frank J., "Artificial Intelligence and Intrusion Detection: Current and Future Directions", NSA URP MDA904-93-C-4085, June, 1994.

Frank, J., "Artificial Intelligence and Intrusion Detection: Current and Future Directions." *Proceedings of the National Computer Security Conference,* 1994.

Garvey, T. D. and Lunt, T. F., "Model Based Intrusion Detection", *Proceeding of the 14th National Computer Security Conference*, Washington, DC, pp.372-385, October 1991.

Giarratano, J. and Riley, G., *Expert Systems: Principles and Programming*, PWS Publishing Company, Boston, 1994.

Giodana, A. and Neri, F., "Search-intensive concept induction", *Evolutionary Computaion,* Vol.3, No.4, pp.375-416, 1996.

Goldberg D. Genetic Algorithm in Search, Optimization and Machine Learning. Addision-Wesley, 1989

Goldberg, D. E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Addison-Wesley, 1989.

Habra, N., Le Charlier, B., Mounji, A. and Mathieu, I., "ASAX: Software Architecture and Rule-base Language for Universal Audit Trail Analysis", in *Proceedings of the Second European Symposium on Research in Computer Security (ESORICS*), Toulouse, France, November 1992. Available at http://www.info.fundp.ac.be/~cri/DOCS/asax.html

Habra, N., Le Charlier, B., Mounji, A., "Preliminary Report on Advanced Security Audit Trail Analysis on uniX", Institut DInformatique, Research Report, December 1991. Available at http://www.info.fundp.ac.be/~cri/DOCS/asax.html

Halme, L. and Bauer, R. K., "AINT misbehaving - a Taxonomy of Anti-Intrusion Techniques", *Proceeding of the 18th National Information Systems Security Conference*, pp.163-172, Oct 1995.

Han, I. G., Y. S. Kwon, and H-K, Jo, “A Review of Artificial Intelligence Models in Business Classification”, *Korean Expert Systems Journal, Vol. 1*, pp.23.-pp.41, 1995.

Hartigan, J. A (1975). *Clustering algorithms.* Wiley, NY.

Heady R., Luger G., Maccabe A., Servilla M. "The architecture of a network level intrusion detection system". Technical Report, Department of Computer Science,University of New Mexico, August 1990.

Heady, R. et al, "A Prototype Implementation of a Network Level Intrusion Detection System", Technical Report CS91-11, Dept of Computer Science, University of New Mexico, Albuquerque, New Mexico 87131-1386, 1991.

Heberlein, L. T., et al., "A Network Security Monitor", *Proceeding of 1990 Symposium on Research in Security and Privacy*, Oakland, CA, pp.296-304, May, 1990.

Hekanaho, J., DOGMA: A GA-Based Relational Leaner, TUCS Technical Report No.168, Turku Centre for Computer Science, May, 1997.

Hoagland, J., Wee, C., Levitt, K. N., "Audit Log Analysis Using the Visual Audit Browser Toolkit", U.C. Davis, Computer Science Department, Technical Report CSE-95-11, 1995. Available at http://seclab.cs.ucdavis.edu/papers.html

Hofmeyr, S. A., Somayaji, A., and Forrest, S., "Intrusion Detection System Using Sequences of System Calls", *Journal of Computer Security* (In press).

Hofmeyr, S., *An Immunological Model of Distributed Detection and Its Application to Computer Security*, Phd Thesis, Dept of Computer Science, University of New Mexico, 1999.

Ilgun, K., Kemmerer, R. A., and Porras, P. A., "State Transition Analysis: Rule-Based Intrusion Detection Approach", *IEEE Transactions on Software Engineering*, Vol. 21, No. 3, pp.181-199, March 1995.

Ilgun, K., *USTAT: A Real-Time Intrusion Detection System for UNIX*, MSc Thesis, Department of Computer Science, University of California Santa Babara, 1992.

Ishibuchi, H., Murata, T., and Nakashima, T., Genetic Algorithm-Based Approaches to Classification Problems, Fuzzy Evolutionary Computation, Witold Pedrycz(Ed), pp.127-153, 1997.

Jackson, K., DuBois, D. and Stallings, C., "An Expert System Application for Detecting Network Intrusion Detection", *Proceeding of the 14th National Computer Security Conference*, pp.215-225, Oct. 1991.

Jackson, K., DuBois, D. and Stallings, C., "The NIDES Statistical Component Description and Justification", Technical Report, Computer Science Laboratory, SRI International, Menlo Park, CA, March, 1994.

Janikow, C. Z., A Knowledge-Intensive Genetic Algorithm for Supervised Learning, *Machine Learning*, Vol.12, pp.189-228, 1993.

Javitz, H. S. and Valdez, A., "The SRI IDES Statistical Anomaly Detector", *the Proceeding of IEEE Symposium on Research in Security and Privacy*, Oakland, CA, May 1991, pp.316-376. Available at http://www.csl.sri.com/nides/index.html

Jo. H., "Bankruptcy Prediction using Multivariate Discriminant Analysis, Analogical Reasoning, and Neural Network", *Master's Thesis, KAIST*, Korea, 1994.

Kephart J. O., A Biologically Inspired Immune System for Computers, High Integrity Computing Laboratory, IBM Thomas J. Watson Research Center, MIT Press, 1994

Kephart, J. O., "A Biologically Inspired Immune System for Computers", *Artificial Life IV, Proceeding of the Fourth International Workshop on the Synthesis and Simulation of Living Systems*, pp.130-139, 1994.

Kim, J, *Neural Networks for Motor Insurance Rating,* MSc thesis, Dept of Artificial Intelligence, University of Edinburgh, 1996.

Kim, J. and Bentley, P., "The Artificial Immune Model for Network Intrusion Detection", *7th European Conference on Intelligent Techniques and Soft Computing (EUFIT99), Aachen, German, September 13- 19, 1999.*

Kim, J. and Bentley, P., "The Human Immune System and Network Intrusion

Detection", *7th European Congress on Intelligent Techniques and Soft Computing (EUFIT '99), Aachen, Germany, September 13- 19, 1999.*

Kim, J. W., *A Comparative Analysis of Rule Based, Neural Network and Statistics Classification for the Bond Rating Problem*, Phd Thesis, Dept of Information Systems, Virginia Commonwealth University, 1992.

Kim, J. W., J. U. Choi, H. U. Choi, Y. Chung, and B. H. Kang, "Developing a Neural Net-Based Credit Evaluation System with Noisy Data", *Proceedings of the International Conference on Neural Information Processing*, Seoul, Oct., 1994.

Kim, J. W., "A Comparative Analysis of Rule Based, Neural Networks, and Statistical Classification Systems for the Bond Rating Problem", *Unpublished doctoral dissertation*, Virginia Commonwealth University, Richmond, Virginia, 1992.

Kim, J. W., "A Neural Net-Based Credit Evaluation System", *Thesis for degree of M.S, Dept. of Applied Computer Science, Graduate School of Management and Information Science*, Hankuk University of Foreign Studies, Feb, 1994.

Ko, C., et al, "Analysis of an Algorithm for Distributed Recognition and Accountability", *Proceeding of ACM Conference of Computer and Communication Security*, Fairfax, MD, Nov 3-5, 1993. Available at http://seclab.cs.ucdavis.edu/~stanifor

Ko, C., *Execution Monitoring of Security-Critical Programs in a Distributed System: A Specification-Based Approach*, Department of Computer Science, UC Davis, Ph.D. Thesis, August 1996.

Ko, C., Fink, G., Levitt, K., "Automated Detection of Vulnerabilities in Privileged Programs by Execution Monitoring". *Proceeding of the 10th Annual Computer Security Applications Conference*, Orlando, FL, pp.134-144, 5-9 Dec. 1994.

Koza, J., *Genetic Programming: On the programming of computers by means of* *natural selection.,* MIT press, 1992.

Kumar S. and Spafford G. "A Pattern Matching model for Misuse Intrusion Detection". In Proceedings of the 17th National Computer Secutiry Conference, October 1994

Kumar, S., and Spafford, E. H., "A Pattern Matching Model for Misuse Intrusion Detection", *Proceeding of National Computer Security Conference*, Baltimore, MD, pp.11-21, 1994. Available from http://www.cs.purdue.edu/coast/coast-library.html

Kumar, S., *Classification and Detection of Computer Intrusions*, PhD Thesis, Department of Computer Science, Purdue University, August, 1995.

Laird, P., "Machine-Learning in intrusion and misuse detection". In Proceeding of Workshop on Future Detections in Computer Misuse and Anomaly Detection, Univ. of California, Davis, March 1992

Lam, K. Y., Hui, L., and Chung, S. L., A Data Reduction Method For Intrusion Detection, *Journal Of Systems And Software*, Vol. 33, No. 1, pp.101-108, 1996.

Lane, T and Brodley, C. E., "Sequence Matching and Learning in Anomaly Detection for Computer Security", *Proceeding of AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management*, 1997.

Lane, T. and Brodley, C. E., "An Application of Machine Learning to Anomaly Detection", *Proceeding of 20th Annual National Information Systems Security Conference*, 1997. Available at http://www.cs.purdue.edu/coast/coast-library.html

Lane, T., and Brodley, C. E., "Detecting the Abnormal: Machine Learning in Computer Security", Technical Report ECE-97-1, January 1997, Department of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907. Available at http://www.cs.purdue.edu/coast/coast-library.html

Langley, P., *Elements of Machine Learning*, Morgan Kaufmann Publishers, San

Francisco, 1996.

Lawrence Livermore National Lab., Sandia National Lab.,"National Info-Sec Technical Baseline - Intrusion Detection and Response.". Oct. 1996.

Lee J. S., J. H., Han, “Usability Test of Non-financial Information in Bankruptcy Prediction using Artificial Neural Network - *The Case of Small and Medium-Size Firms", Korean Expert Systems Journal, Vol. No. 1.*, pp123.-pp134, 1995.

Lee K. C., I. G., Han, M-J, Kim, "A Study on the Credit Evaluation Model Integrating Statistical Model and Artificial Intelligence Model", *Korean Management Science*, Vol.21, No.1, pp.81-pp.100, 1996.

Lee, W., *A Data Mining Framework for Constructing Feautres and Models for Intrusion Detection Systems*, Phd Thesis, Dept of Computer Science, Columbia University, 1999.

Lee, W., Stolfo. S., and Chan, P., "Learning Patterns from Unix Process Execution Traces for Intrusion Detection", *Proceeding of AAAI 97 Workshop on AI Methods in Fraud and Risk Management*, 1997. Available at http://www.cs.columbia.edu/~sal/recent-papers.html

Leinweber, D., "Knowledge-Based Systems for Financial Applications", *IEEE Expert,* Fall, pp.18-31, 1988.

Leinweber, D., “Knowledge-Based Systems for Financial Applications”, *IEEE EXPERT*, pp.18-pp.31, Fall 1988.

Liang, T., J. Chandler, I. Han, and J. Roan, “An Empirical Investigation of Some Data Effects on the Classification Accuracy of Probit, ID3, and Neural Networks”, *Contemporary Accounting Research*, pp.306-pp.328, Fall 1992.

Liepins, G. E., and Vaccaro, H. S., "Anomaly Detection: Purpose and Framework", *Proceedings of the 12th National Computer Security Conference,* pp.495-504, 1989.

Lunt, T. F. and Jagannathan, R., "A Prototype Real-Time Intrusion Detection", *Proceeding of 1988 IEEE Symposium on Security and Privacy*, Oakland, CA, pp.59-66, April 1988.

Lunt, T. F., "Automated Audit Trail Analysis and Intrusion Detection: A Survey", *Proceeding of 11th National Computer Security Conference*, Baltimore, MD, Oct 1988. Available at http://www.csl.sri.com/nides/index5.html

Lunt, T. F., "Detecting Intruders in Computer Systems", *Proceeding of 1993 Conference on Auditing and Computer Technology*, 1993.

Lunt, T. F., et al., "A Real-time Intrusion Detection Expert System (IDES)", Technical Report SRI-CSL-92-05, Computer Science Laboratory, SRI International, Menlo Park, CA, April 1992.

Mallinson, H. and Bentley, P.J. (1999). Evolving Fuzzy Rules for Pattern Classification. In *International Conference on Computational Intelligence for Modelling, Control and Automation - CIMCA'99.*

Marmelstein, R. E., and Lamont, G. B., "Evolving Compact Decision Rule Sets", in Koza, J. (Ed.), Late Breaking Papers at the Genetic Programming 1997 Conference, University of Wisconsin, July 22-25, pp.144-150, 1998.

Mattahias Schumann, Thomas Lohrbach, “Comparing Artificial Neural Networks with Other Methods within the Field of Credit Scoring”, *Proceedings of 2nd Pacific Rim International Conference on AI*, Vol.I, pp.72-pp.78, 1992.

Me, L., "GASSATA, a Genetic Algorithms as an Alternative Tool for Security Audit Trails Analysis", Available at http://www.supelec-rennes.fr/rennes/si/equipe/lme/these/these-lm.html

Me, L., "Genetic Algorithms, as Alternative Tool for Security Audit Trail Analysis", Available at http://www.supelec-rennes.fr/rennes/si/equipe/lme/these/

these-lm.html

Messier, and Hanssen, J. V., "Expert Systems in Auditing : The States of the Art", *Auditing: A Journal of Practice and Theory*, Vol.7, No.1, pp.94-105, 1987.

Messier, W., Jr., and J. Hansen, “Inducing Rules for Expert System Development : An Example Using Default and Bankruptcy Data”, *Management Science, December 1988*, pp.1403-1415.

Mitchell, T., *Maching Learning*, McGraw-Hill Inc., 1997.

Mounji, A., Le Charlier, B., Zampunris, D., and Habra, N. "Preliminary Report on Distributed ASAX", *Institut d'Informatique*, Research Report, May 1994.

Mounji, A., Le Charlier, B., Zampunris, D., and Habra, N., "Distributed Audit Trail Analysis", *Proceeding of the ISOC '95 Symposium on Network and Distributed Systems Security*, San Diego, California, February 1995.

Mykerjee, B.; Heberlein, L. T.; Levitt, K. N., "Network Intrusion Detection", *IEEE Network,* Vol.8, No.3, pp.26-41, 1994.

Noel, C., "Credit Scoring Systems : A Critical Analysis", *Journal of Marketing, (Vol. 46)*, Spring 1982, pp.82-91.

Noel, C., "Credit Scoring Systems: A Critical Analysis", *Journal of Marketing,* Vol.48, Spring, pp.82-91, 1982.

Obaidat, M. S., and Macchiarolo, D. T., "An On-Line Neural Network System for Computer Access Security", *IEEE Transactions on Industrial Electronics*, Vol.40, No.2, April, pp.235-242, 1993.

Odom, M. D. and Sharda, R., "A Neural Network Model for Bankruptcy Prediction", *Proceeding of IJCNN,* 2, 1990.

Odom, Marcus D., Ramesh Sharda, “A NEURAL NETWORK MODEL FOR

BANKRUPTCY PREDICTION", *Proceedings of IJCNN 1990* Vol II, pp.163-pp.168, 1990

Paller, *"Selecting the Right Intrusion Detection Tools and Active Auditing Tools", course book,* Alan Paller (Ed), SNAS Institute, Oct 1998.

Paul, W. E., "The Immune System: An Introduction", *Fundamental Immunology* 3rd Ed., W. E. Paul (Ed), Raven Press Ltd.

Paxson, V., "Bro:A System for Detecting Network Intruders in Real-Time", *Proceeding of 7th USENIX Security Symposium,* San Antonio, TX, January, 1998. Available at http://www-nrg.ee.lbl.gov/nrg-papers.html

Pedrycz, W. (Ed.), *Fuzzy Evolutionary Computation,* Kluwer Academic Publishers, MA., 1997.

Playfair, J. H. L., *Immunology at a Glance*, 6th Ed, Blackwell Science, 1996

Porras, P. A., *STAT: A State Transition Analysis Tool for Intrusion Detection,* MSc Thesis, Department of Computer Science, University of California Santa Babara, 1992

Porras, P. A.; Neumann, P. G., "EMERALD: Event Monitoring Enabling Responses to Anomalous Live Disturbances", *Proceeding of 20th National Information System Security Conference.,* 1997. Abailable at http://www.csl.sri.com/emerald/downloads.html

Porras, P. A.; Valdes, A., "Live Traffic Analysis of TCP/IP Gateways", *Proceeding of ISOC Symposium of Network and Distributed System security, 1998.* Available at http://www2.csl.sri.com/emerald/downloads.html

Potter, M. A., *The design and analysis of a computational model of cooperative co-evolution.* PhD Thesis, George Mason University, Fairfax, VI., 1997.

Pukeza, N. J., et al., "A Methodology for Testing Intrusion Detection Systems", *IEEE Transactions on Software Engineering*, Vol. 22, No. 10, pp.719-729, February 1996.

Ranum, M. J. and et al, Implementing a Generalized Tool for Network Monitoring, *Proceedings of the 11th Systems Administration Conference (LISA '97)*, San Diego, California, USA, October 26-31, 1997. Available at http://www.nfr.net/forum/publications/LISA-97.htm

Raymer, M. L., Punch, W. F., Goodman, E.D., and Kuhn, L. A., "Genetic Programming for Improved Data Mining Application to the Biochemistry of Protein Interactions", *Proceeding of the First Annual Genetic Programming Conference*, pp.375-380, Stanford University, 1996.

Raymond B. Stephen R. Spence Raj Nigam, "Margin Credit Evaluation System", *IEEE*, pp.128-pp.134, 1991

Roitt, I., Brostoff, J., and Male, D., *Immunology*, Fifth Ed., Mosby International Ltd, 1998.

Ross, P. and Hallem, J., *Connectionist Computing*, Dept of AI, University of Edinburgh, Edinburgh, 1995

Rumelhart, D. E., G. E. Hinton, & R. J. Williams.(1986a). "Learning Internal Representations by Error Propagation." In D. E. Rumelhart & J. L. McCelland, eds., *Parallel Distributed Processing, vol. 1.* Reprinted in Anderson & Rosenfeld[1988], pp 675-695.

Rumelhart, D. E., G. E. Hinton, & R. J. Williams.(1986b). "Learning Internal Representations by Back-Propagating Error." Nature, 323:533-536. Reprinted in Anderson & Rosenfeld[1988], pp.675-695.

Ryan, M. D., and Rayward-Smith, V. J., "The Evolution of Decision Trees", *Proceeding of the Third Annual Genetic Programming Conference*, pp.350-358, University of Wisconsin, Madison, Wisconsin, 1998.

Ryu, T., and Eick, C. F.,. "Deriving queries from results using genetic programming",. *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining,* pages 303--306, Portland, OR, 2.-4. August 1996. AAAI Press, Menlo Park, CA, 1996.

Shortliffe, E. H and B. G. Buchnan, “A Model of Inexact Reasoning in Medicine”, *Mathematical Biosciences(1975),* pp.351-379.

Simon, H. A., "Why Should Machine Learn." in R. S. Michalski, J. G. Carbonell, and T. M. Mitchell(Ed.), Machine Learning : An Artificial Intelligence Approach. 1983.

Smith, R. E., Forrest, S., and Perelson, A. S., "Searching for Diverse, Cooperative Populations with Genetic Algorithm", *Evolutionary Computation,* 1(2), 127-149, 1993.

Smith, S. F., "Flexible learning of problem solving heuristics through adaptive search", *Proceeding of the 8th Internaional Joint Conferenece on Artificial Intelligence,* pp.422-425, Karlsruhe, Germany, 1983.

Snapp, S., et al., "DIDS(Distributed Intrusion Detection System)- Motivation, Architecture, and An Early Prototype", *Proceeding of 14th National Computer Security Conference, Washington, D. C.,* pp.167-176, Oct., 1991.

Somayaji, A.; Hofmeyr, S.; Forrest, S., 1997, "Principles of a Computer Immune System", *Proceeding of New Security Paradigms Workshop, Langdale, Cumbria,* pp.75-82, 1997.

Spafford, E. H., "Computer Viruses as Artificial Life", *Journal Of Artificial Life;* Vol. 1, No. 3, pp. 249-265, 1994.

Staniford-Chen, S. G. and Heberlein, L. T., "Holding Intruders Accountable on the Internet", *Proceedings of the 1995 IEEE Symposium on Security and Privacy,* Oakland, CA. 1995. Available at http://seclab.cs.ucdavis.edu/~stanifor

Staniford-Chen, S., et al., "GrIDS -- A Graph-Based Intrusion Detection System for Large Networks", *Proceeding of the 19th National Information Systems Security Conference.,* 1996.

Surkan, A. J., and J. C. Singleton, “Neural Networks for Bond Rating Improved by Multiple Hidden Layers”, *Proceedings of IJCNN 1990*, Vol.Ⅱ, pp.157-pp.162., 1990.

Surkan, A. J. and Singleton, J. C., "Neural Network for Bond Rating Improved by Multiple Hidden Layers", *Proceeding of IJCNN,* 2, 1990.

Tam, K. and Kiang, “Managerial Applications of Neural Networks : The Case of Bank Failure Predictions”, *Management Science*, pp.926-pp.947, July 1992.

Teng, H., Chen, K., and Lu, S., "Adaptive Real-Time Anomaly Detection Using Inductively Generated Sequential Patterns", *Proceeding of the 1990 Symposium on Security and Privacy*, Oakland, CA, May 7-9, pp.278-284, 1990.

Tizard, I. R., *Immunology: Introduction*, 4th Ed, Saunders College Publishing, 1995.

Vigna, G and Kemmerer, R. A., "NetSTAT: A network-based intrusion detection approach," *Proceedings of the 14th Annual Computer Security Applications Conference*, Scottsdale, Arizona, December 1998.

Walter H., “Model-based Financial Data Interpretation”, *IEEE*, pp.178-pp.186, 19991.
Wee, C., "LAFS: A Logging and Auditing File System", *Proceeding of the 11th Computer Security Applications Conference,* 1995. Available at http://seclab.cs.ucdavis.edu/papers.html

White, G. B.; Pooch, U; Fisch, E. A, "Cooperating Security Managers: A Peer-Based Intrusion Detection System, *IEEE Network*, Vol.10, No.1, pp.20-23, 1996.

White, G., B., and Pooch, U., "Cooperating Security Managers-Distributed Intrusion Detection Systems", *Computer & Security*, Vol.15, No.5, pp.441-450, 1996.

Witten, I.H. and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, 2000.

Wolberg, W. H., and Mangasarian, O. L. "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", In *Proceedings of the National Academy of Sciences*, 87, pp.9193--9196, 1990.

Yu, T. and Bentley, P., "Methods to Evolve Legal Phenotypes.", In *Proceedings of the Fifth Int. Conf. on Parallel Problem Solving From Nature*. Amsterdam, Sept 27-30, 1998, pp. 280-282, 1998.

Zhang, J., "Selecting typical instances in instance-based learning". In *Proceedings of the Ninth International Machine Learning Conference*, pp. 470--479. Aberdeen, Scotland: Morgan Kaufmann, 1992.

박무송, 소비자 금융과 신용, 행림출판사, 1988

송연선, 재무재표의 신뢰성에 관한 연구, 한국신용평가(주), 1988

채서일, 사회과학 조사방법론, 학연사, 1995.

Available at http://www.csl.sri.com/nides/index.html

Available at http://seclab.cs.ucdavis.edu/papers.heml

Available at http://seclab.cs.ucdavis.edu/papers.html

Available at http://seclab.cs.ucdavis.edu/papers.html

available at http://www.cs.purdue..edu/coast/coast-library.html

Available at http://www.cs.purdue.edu/coast/coast-library.html

Available at http://www.cs.ucsb.edu/~kemm/netstat.html/documents.html

Available at http://www.cs.unm.edu/~steveah/papers.html

Available at http://www.info.fundp.ac.be/~amo/publicatoins.html

Available at http://www.msci.memphis.edu:80/~dasgupta/publications.html