컴퓨터 바이러스 분류를 위한 퍼지 클러스터 기반 진단시스템

이 현 숙†

요 약

중요한 정보를 저장하고 있는 컴퓨터를 위협하는 바이러스는 점점 현실적인 문제로 대두되고 있다. 이를 위하여 바이러스 침입 발견을 위한 소프트웨어 기술 또한 계속 발전되고 있으나, 현재까지의 표준 기술은 알려진 바이러스의 시그내쳐 패턴을 저장하여 이를 매치 검색하면서 바이러스를 찾아내는 방식을 채택하고 있다. 이는 알려진 바이러스에 대해서는 효과적이지만 새로운 바이러스를 찾아내지 못하고 손실을 당한 후에야 찾을 수 있는 단점을 가지고 있다. 이를 위하여 바이러스 정보 구축과 탐색에 학습기능을 도입함으로 새로 발생하는 바이러스를 찾아내어 대처할 수 있는 방법이 필요하다.

본 논문에서는 컴퓨터 바이러스를 위한 퍼지 진단 시스템 FDS를 제안한다. FDS에서는 FCM 알고리즘을 사용하여 알려진 정보의 클러스터를 형성하고 대표정보를 추출하고 여기에 전문가의 지식을 포함하는 지식베이스를 구축한다. 진단을 위한 컴퓨터 파일에 대하여 그 파일의 결정 상태를 확인하고 이미 저장된 지식베이스를 바탕으로 바이러스 침입에 대한 정보를 보고하도록 설계되어있다. 이 시스템은 이미 알려진 테스트 데이터와 이전에 알려지지 않은 새로운 테스트 데이터를 실험데이터로 준비하여 널리 알려진 분류 알고리즘-KNN, RF, SVM-과 함께 성능을 비교하였다. 제안된 시스템이 알려지지 않은 컴퓨터 바이러스를 효과적으로 진단할 수 있는 타당성을 보이고 있다.

키워드 : 퍼지 클러스터 분석, 악성코드, 지식획득, 결정 상태 척도

Fuzzy Cluster Based Diagnosis System for Classifying Computer Viruses

Rhee, Hyunsook†

ABSTRACT

In these days, malicious codes have become reality and evolved significantly to become one of the greatest threats to the modern society where important information is stored, processed, and accessed through the internet and the computers. Computer virus is a common type of malicious codes. The standard techniques in anti-virus industry is still based on signatures matching. The detection mechanism searches for a signature pattern that identifies a particular virus or stain of viruses. Though more accurate in detecting known viruses, the technique falls short for detecting new or unknown viruses for which no identifying patterns present. To cope with this problem, anti-virus software has to incorporate the learning mechanism and heuristic.

In this paper, we propose a fuzzy diagnosis system(FDS) using fuzzy c-means algorithm(FCM) for the cluster analysis and a decision status measure for giving a diagnosis. We compare proposed system FDS to three well known classifiers-KNN, RF, SVM. Experimental results show that the proposed approach can detect unknown viruses effectively.

Key Words :Fuzzy Cluster Analysis, Malicious Code, Knowledge Acquisition, Decision Status Measure

1. 서 론

마이크로소프트 사의 최근통계에 의하면 전체 컴퓨터의 약 0.28%가 하나 이상의 악성코드에 감염 되어 있는 것으로 보고되고 있다[1]. 악성코드의 명확한 정의에 대하여 토론 중에 있으나 소프트웨어 시스템에 첨가되거나 변경되어 시스템의 기능을 손상시키는 코드로서 보통 컴퓨터 바이러스라고 한다[2]. 최근 알려진 바이러스를 탐지하는 기술이 발전되어 McAfee Virus Scan 이나 Norton Anti-Virus와 같은 성공적인 소프트웨어들이 활용되고 있다. 최근 연구는 데이터로서 제공되는 실행파일의 분석에 집중되어 텍스트분류나 음성인식 등의 언어처리에 사용되던 n-gram 분석모델을 도입하게 되었다[3]. n-gram 분석방법은 주어진 파일로부터 n개의 연속적인 부분 문자열을 추출하여 확률적으로 전체 파일의 구조를 표현하려는 시도이다[4,5]. 바이러스 탐지 알고리즘은 바이러스 파일과 정상파일의 구조를 분석하여 서로 다른 점에 대한 정보가 중요한 역할을 하므로 n-gram 분석방법은 의미 있는 결과를 얻었고 분석하는 과정에 데이터 마이닝 기법과 정보이론이 접목되어 그 성능을 증가시키기도 하였다[6]. 이러한 연구를 바탕으로 추출된 부

† 정 회 원 : 동양공업전문대학 전산정보학부 부교수
논문접수 : 2006년 11월 1일, 심사완료 : 2007년 1월 16일

분 문자열은 시그내처로 이용되어 매칭을 기반으로 하는 분류 알고리즘에 의하여 정상파일인지 바이러스 파일인지 판정 하게 된다. 또 다른 접근 방법으로 주어진 파일을 의미망으로 표현하고 이를 그래프 탐색하는 상위레벨 접근방법을 사용하고 있다. 이는 주어진 파일이 항상 파싱 가능하지도 않으며 파싱이 되는 경우도 코드를 이해하는 어렵고 복잡한 과정을 포함하고 있으므로 현실적으로 구현하기 어렵다. 이와 같은 방법은 알려진 바이러스 파일의 분석과 단순 매칭에 기반을 두고 있다. 그러므로 알려지지 않은 바이러스를 탐지하지 못하고 그 바이러스에 의하여 시스템이 공격을 받고 난 후 바이러스 유형과 파일상태를 분석한 후에야 탐지 알고리즘에 반영될 수 있다. 그러는 사이 시스템은 손상되고 또 다른 형태의 바이러스가 만들어질 것이다. 이를 위하여 바이러스 전문가의 휴리스틱한 규칙이 적용되기도 했으나 간단하고 유연성이 없어 쉽게 노출되며 실세계에 적용하기 어려운 단점을 가지고 있다. 또한 K-nearest neighbor(KNN), Random Forests(RF)과 Support Vector Machine(SVM) 등 기존의 기계학습 연구 결과 발표된 분류 알고리즘[7]이 적용되기도 하였으나 일반화시키기 어려웠다.

이에 바이러스정보구축과 탐지과정에 학습기능과 전문가의 경험지식을 체계적으로 통합하여 새로운 바이러스 파일도 탐지하고 대처할 수 있는 방법이 필요 하게 되었다. 본 논문에서는 학습기능과 경험지식을 통합할 수 있는 컴퓨터 바이러스 탐지를 위한 퍼지 진단 시스템(Fuzzy Diagnosis System, FDS)을 설계한다. FDS에서는 주어진 파일로부터 바이러스 관련 정보를 구축하기 위한 사전정보(a priori information)로 사용하기 위하여 데이터 클러스터 분석기법을 사용한다. 클러스터 분석기법은 속성이 비슷한 것끼리 묶어 나누는 것으로, 분석하고자하는 데이터가 너무 많아 전체를 파악하기 어려울 때, 전체의 윤곽을 잡게 해 주는 기법으로 영상처리, 음성인식 등 분류가 필요한 모든 영역에 사용되는 기초단계이다. 분석하고자 하는 데이터의 경계가 명확하지 않은 대부분의 실세계 응용에서 클러스터를 퍼지 집합으로 표현하여 처리 하는 퍼지 클러스터 분석 방법을 통하여 보다 정확한 정보를 얻을 수 있다. 이에 본 논문에서는 널리 알려진 퍼지 클러스터 분석 알고리즘, fuzzy c-menas(FCM)을 적용하여[8] 얻어진 c개의 각 클러스터에 파일 수집에 참여한 바이러스 전문가의 지식을 부착하여 지식베이스를 구축한다. 다음은 FDS의 진단과정으로서 바이러스를 탐지하고자하는 주어진 파일데이터에 대하여 각 클러스터와의 퍼지 소속 값을 구한 후 그 값을 이용하여 결정상태(decision status)를 확인하고 지식베이스의 정보를 바탕으로 진단 결과를 출력해 준다. FDS의 성능은 잘 알려진 분류알고리즘-KNN, RF, SVM-과 함께 비교하여 제안된 방법의 타당성을 입증한다.

본 논문은 2장에서는 본 연구와 관련된 컴퓨터 바이러스 탐지 방법과 퍼지클러스터 분석방법인 FCM 알고리즘을 기술한다. 3장은 논문의 중심 부분으로 제안된 FDS의 지식획득 모듈과 진단모듈의 구성과 이에 사용된 퍼지 측정함수를 기술한다. 4장은 설계된 FDS의 타당성을 확인하기 위하여 준비된 데이터를 가지고 실험하고 널리 알려진 다른 방법과 비교한 결과를 소개한다. 5장은 결론으로서 제안한 시스템 FDS를 요약하고 앞으로의 발전방향을 기술한다.

2. 관련연구

2장에서는 본 연구와 관련된 컴퓨터 바이러스 탐지 방법과 퍼지 클러스터 분석방법인 FCM 알고리즘을 관련연구로서 고찰해 보고자한다.

2.1 컴퓨터 바이러스 탐지기법

1986년 컴퓨터 바이러스가 처음 출현한 이래 매년 수많은 새로운 바이러스가 중요한 정보를 저장하고 있는 컴퓨터를 위협하고 있으므로 이는 점점 현실적인 문제로 대두되고 있다. 이에 바이러스 파일을 분석하여 시스템으로부터 이를 탐지하여 대처하기 위한 연구들이 계속되고 있다. 이러한 연구의 대부분은 탐지를 위한 중요한 정보로서 코드의 특정부분 문자열을 시그내처로 추출하기 위한 방법론에 초점을 맞추고 있다. 최근에 그동안 문서분류와 음성인식 등의 자연언어처리 분야에 활용되어 온 n-gram 분석 기법이 컴퓨터바이러스 탐지를 위한 시그내처 추출에 활용되고[9], "Computer immune system" 의 설계를 위하여 제안되기도 하였다[10]. 이러한 연구는 알려진 바이러스의 시그내처를 추출하여 저장하고 이를 검색하여 처리하는 시그내처 기반 탐지기법을 바탕으로 하고 있다. 새로운 바이러스에 대하여 이의 시그내처를 추출하고 이를 탐지 알고리즘에 반영하는 동안 시스템은 이미 손실을 입고 또 다른 바이러스가 등장하게 된다.

Abou-Assaleh 등[4]은 Commom N-Gram(CNG) 방법을 제안하여 알려지지 않은 새로운 파일을 진단하는데 활용하였다. 악성코드와 정상코드로 구성된 데이터베이스로부터 자주 출현하는 n-gram을 시그내처 로서 추출하여 저장한다. 이렇게 추출된 n-gram은 특정파일의 구조를 반영하는 정보를 함축하고 있으며 바이러스 침입자가 쉽게 예측하기 어려운 것으로 알려져 있다. 분석하고자하는 코드에 대하여 이미 저장된 시스내처로부터 k-nearest 알고리즘을 적용하여 정상코드인지 악성코드인지 분류 하게 된다. 이 방법은 파라메타 n과 추출된 시스내처의 수 L에 따라 민감하게 그 성능이 좌우되는 것으로 보고 되었으나 알려지지 않은 새로운 파일을 대상으로 하는 초기연구로서 가치가 있다.

Kolter 등[6]도 비슷한 방법으로 접근하고 있으나 정보공학의 기법을 적용하였다. 특징으로 추출된 n-gram 들이 준비된 각 파일에 존재여부를 지시하는 이진데이터를 모아 평균상호 정보(average mutual information)를 계산하여 그 값이 큰 500개의 데이터를 선택하여 WEKA[7]에서 구현한 학습방법-Instance-based Learner, TFIDF, Naive Bayes, a support vector machines, a decision tree and a booted classifier-에 적용하였다. 준비된 데이터의 분류정확도에 의하여 분류하지 않고 ROC(receiver operating characteristics)에 의하여 평가하였다.

이와 같이 n-gram 기법들은 적용 가능 하기는 하나 부분 문자열의 크기 n과 특징패턴의 개수 L과 같은 파라메타에 종속적인 결과를 가져오므로 일반적인 접근방법으로 발전시키기는 어렵다. n-gram 분석방법이 분류에 공헌하는 점을 관찰하여 이진실행파일을 역어셈블 하여 명령어를 구성하는 연산코드로부터 instruction sequence를 특징패턴으로 추출하는 기법을 제안하였다[11]. 본 논문에서는 이를 바탕으로 하는 다음의 데이터 준비과정을 사용하여 특징을 추출하여 FDS의 입력데이터로 활용하였다. 우선 수집한 이진실행파일(binary executables)을 IDA Pro[12]를 사용하여 역어셈블한다. 이 과정에서 대부분의 실행파일은 완전하게 역어셈블되지 않기 때문에 경우에 따라 휴리스틱을 사용하여 추출하기도 한다. 이제 역어셈블 된 코드를 블록으로 나누고 각 블록에 대하여 블록이름과 그 안에 있는 명령어의 연산 코드(instruction operation code)로 구성된 중간 파일을 만든다. 이 중간 파일로부터 각 명령어의 출현빈도수를 구하고 이를 바탕으로 특징 연산코드를 추출하게 된다. 이러한 처리과정은 악성코드의 탐지는 정상코드와 구별되는 특징패턴으로부터 알 수 있고 그런 정보는 파일을 구성하는 명령어로부터 얻어질 수 있다는 자연스러운 아이디어에서 출발하고 있다. 특히 명령어는 연산코드와 피연산자 부분으로 되어 있는데 연산코드만으로도 파일의 내용을 모두 표현할 수 있고 특징패턴을 추출할 수 있다. 피연산자 부분은 고려하지 않으므로 중간파일의 크기를 상당히 줄일 수 있고 입력 데이터의 준비를 단순화 시킬 수 있다.

이와 같이 지금까지의 바이러스 탐지 방법은 특징패턴 추출에 집중하고 있으며 이를 통해 준비된 데이터를 가지고 알려진 분류알고리즘을 적용하여 처리하고 있다. 물론 특징추출은 영상처리나 음성인식등과 마찬가지로 성능에 직접적인 영향을 미치는 중요한 요소이다. 그러나 추출된 정보를 제대로 활용하기 위하여 기존의 인공지능이나 소프트 컴퓨팅의 연구결과와 접목시키는 노력 또한 필요하다. 본 논문에서는 여기에 초점을 두고 추출된 데이터를 가지고 퍼지 클러스터 분석과 전문가의 의견을 통하여 지식베이스를 구축하고 퍼지이론과 전문가시스템의 기능을 활용하여 진단결과를 제시하는 기법을 제안 한다

2.2 FCM 알고리즘

퍼지이론은 Zadeh에 의하여 소개되어 이분법적인 데이터 표현방법을 확장시켜 데이터를 보다 정확하게 기술하여 처리할 수 있는 이론적인 바탕을 마련하였다. 이는 주어진 데이터의 구조를 파악하기 위한 클러스터 분석방법에도 적용되어 Bezdek에 의하여 fuzzy c-means (FCM) 알고리즘이 개발되어 최적 분할, 패턴분류 및 영상 처리 등의 여러 분야에 활용되었다[8]. FCM 알고리즘은 식(1)과 같은 클러스터안의 데이터사이의 거리와 소속 값에 기반을 둔, 최소자승오류함수, J_m 을 목적함수로하여 이를 반복최적화 하도록 구성되어있다.

$$J_m = \sum_{j=1}^{n}\sum_{i=1}^{c}(u_{ij})^m (d_{ij})^2 \tag{1}$$

여기서 u_{ij}는 주어진 입력 데이터 집합 X = $\{\boldsymbol{x}_1,...,\boldsymbol{x}_n\}$에 대한 퍼지 c 분할을 $n \times c$ 의 벡터 U로 나타낼 때 그의 한 요소로 데이터 $\boldsymbol{x}_j$ 의 클러스터 i에 속하는 소속정도를 표현한다. 또한 $(d_{ij})^2 = \| \boldsymbol{x}_j - \boldsymbol{v}_i \|^2$ 이고 $\| \cdot \|$ 은 유클리드 놈을, $\boldsymbol{v}_i$ 는 클러스터 i 의 중심점을 나타내며 $m \in [1, \infty)$ 은 퍼지 정도를 표시하는 파라메타를 나타낸다. 이때 Bezdek은 $m > 1$인 경우 J_m의 국소적 최소점이 되기 위한 필요조건(충분조건은 아니지만)을 유도하여 다음과 같은 FCM 알고리즘을 구성하였다.

[단계 1] 클러스터의 수, weighting exponent m 을 정한다. 퍼지 c 분할 $U^{(0)}$의 초기치를 설정한다. 반복횟수 b=0으로 한다.

[단계 2] 다음 식(2)를 이용하여 클러스터 중심점 $\{\boldsymbol{v}_i^{(b)}\}$를 계산한다.

$$\boldsymbol{v}_i = \frac{\sum_{j=1}^{n}(u_{ij})^m \boldsymbol{x}_j}{\sum_{j=1}^{n}(u_{ij})^m}, \quad 1 \le i \le c \tag{2}$$

[단계 3] 퍼지 c 분할 $U^{(b)}$으로부터 다음 단계의 퍼지 c 분할 $U^{(b+1)}$ 을 구한다.

a) 각 데이터 $j = 1...n$ 에 대하여 다음을 계산한다.

$$I_j = \{i | 1 \le i \le c, d_{ij} = \| \boldsymbol{x}_j - \boldsymbol{v}_i \| = 0\}$$
$$\bar{I}_j = \{1, 2, ..., c\} - I_j$$

b) 각 데이터 $j = 1...n$ 에 대하여 새로운 소속 값 u_{ij}를 계산한다.

i) If $I_j = \varnothing$, $u_{ij} = \dfrac{1}{\sum_{t=1}^{c}\left(\frac{d_{ij}}{d_{tj}}\right)^{2/(m-1)}}$ (3a)

ii) Else

$u_{ij} = 0 \quad \text{if} \; d_{ij}^2 \neq 0 \; \text{and} \sum_{i \in I_j} u_{ij} = 1$ (3b)

[단계 4] $U^{(b)}$ 과 $U^{(b+1)}$ 을 비교하여 $\| U^{(b)} - U^{(b+1)} \| < \epsilon$ 이면 알고리즘을 종료하고; 그렇지 않으면, b = b+1로 정하고 단계 2 로 가서 반복한다.

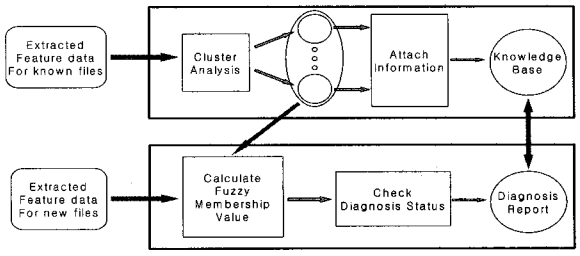
이와 같이 FCM 알고리즘은 단지 식(2)와 식(3)의 반복에 의하여 수렴점을 찾아가는 과정이다[8]. 위의 유도된 식 (3)을 이용하여 입력 데이터와 중심점 사이의 거리를 통한 퍼지 소속 값(fuzzy membership value)을 결정하게 된다. 식(3b)를 통하여 알 수 있는 바와 같이 임의의 한 클래스의

중심점과의 거리가 0인 입력 데이터에 대한 그 클래스안의 퍼지 소속 함수 값은 1이 될 것이다. 그리고 식(3)을 고찰하여 알 수 있는 바와 같이 그를 통하여 결정된 퍼지 소속 함수 값은 형성된 각 클래스에 대하여 상대적인 값을 가지며 확률적인 제약을 준수하여 그 클래스에 대한 소속의 확률치나 공유의 정도로 해석된다. 그러나 퍼지 이론에서 이용하는 소속 함수는 그 클래스에 대한 일치도나 전형성의 정도로서 해석되는 절대적인 값이므로 이를 보완하기 위한 연구도 진행되고 있다. 물론 실세계의 데이터는 그 응용에 종속적이므로 모든 경우에 적합한 알고리즘으로 일반화하기는 어렵다. 그러므로 이 방법의 수렴성과 최적화, 일반화에 대한 고찰은 계속 진행되어 발전되고 있다[13].

3. 퍼지 진단 시스템

컴퓨터 바이러스탐지과정은 일반적으로 처리를 위한 기본 요소인 특징추출과정과 주어진 파일이 악성코드인지 정상코드인지 분류하는 과정으로 나누어져 있다. 지금까지의 많은 바이러스 탐지기법은 활용가능하고 유용한 특징 추출 방법과 시그내처 기반 매칭 기법에 집중해왔다[3,4,5,6]. 일반적으로 특징추출은 같은 범주에 속하는 데이터들의 공통적인 성질(intraset features)과 서로 다른 범주에 속하는 데이터들을 구별하는 성질(interset features)로 나누어 고려되어 왔으며 분류가 필요한 모든 시스템 설계 시 포함되는 중요한 요소이다. 마찬가지로 컴퓨터 바이러스 탐지시스템을 위하여 악성코드와 정상코드가 가지는 공통적인 구조와 구별되는 특성에 해당하는 패턴을 추출해야하고 이는 파일자체 코드로부터 얻을 수 있다. 보통 추출한 특징의 종류와 개수는 시스템의 성능에 큰 영향을 미치므로 여러 단계의 실험과정제과정을 거친다. 본 논문에서는 이러한 특징추출과정을 데이터 준비과정으로 고려하여 이를 활용하여 결과를 이끌어내는 방법론에 초점을 맞추고 있다. 또한 지금까지의 바이러스 탐지를 위한 접근방법과 상용시스템은 이미 알려진 바이러스를 찾는 것을 목표로 하고 있다. 특히 기존의 시스내처 기반 매칭 방법은 데이터베이스에 시스내처를 가지고 있지 않은 새로운 바이러스를 찾지 못한다. 새로운 바이러스에 대하여 바이러스 전문가가 시그내처를 추출하여 데이터베이스를 갱신한 후에야 그 바이러스를 인식할 수 있으므로 이미 시스템은 손상되고 또 다른 바이러스가 생길 수도 있게 된다.

이러한 고찰을 통하여 가지고 있는 파일의 분석을 통한 특징추출 과정 뿐 아니라 이를 해석하여 판단하는 분류과정에



(그림 1) 퍼지 진단 시스템의 전체 구성도

기계학습과 신경망 기법이 가지는 학습기능과 전문가시스템이 가지는 지식구축 및 설명기능이 필요함을 알 수 있다. 그러므로 본 논문에서는 컴퓨터 바이러스를 찾고 이에 대한 정보를 제공해 주는 퍼지 진단 시스템 FDS를 제안한다. FDS는 준비된 특징 데이터정보를 퍼지 클러스터 분석에 의하여 학습한 후 그 결과 학습된 데이터 정보를 가지고 있는 클러스터에 전문가의 지식을 부착하여 지식베이스를 구축한다. 이때 시스템의 목표에 따라 알려진 데이터가 악성 코드로부터 온 것인지, 정상 코드로부터 온 것인지에 대한 단순한 정보로부터 검출된 악성코드의 종류와 대처 방법 등의 경험적 지식까지 다양한 작업이 이루어질 수 있다. 다음은 바이러스 판정을 원하는 파일에 대하여 지식베이스안의 각 클러스터에 대한 퍼지 소속 값을 구한 후 이 값을 이용한 결정상태에 따라 현재 지식베이스에서 제공해 줄 수 없는 상이한 파일인 경우로 판정되면 전문가에게 이를 보고 한다. 그렇지 않으면 지식베이스안의 데이터를 바탕으로 진단결과를 보고해 준다. 이와 같이 바이러스 탐지를 위한 완전한 단독시스템을 설계하는 것은 어려운 일이지만 FDS와 같은 시도는 시스템의 구조를 체계화시키고 전문가의 판단을 도와줄 수 있을 것이다.

3.1 퍼지 클러스터 기반 지식 획득 모듈

지식획득 모듈은 이미 준비된 알려진 파일들의 특징데이터를 시스템에 학습시켜 사전지식을 얻도록 해 준다. 이러한 학습을 위하여 기계학습이론, 신경망 분야의 많은 이론이 있으나 퍼지이론과 통계적인 접근방법을 기초로 학습하는 퍼지 클러스터 분석 알고리즘 FCM을 사용한다. 이는 0과 1의 이분법적 접근으로 처리하는 기존의 승자독점(winner take all) 정보처리 방식의 단점을 보안하여 처리과정에서 손실되는 정보를 고려하여 소속 정보 데이터를 다루므로 주어진 정보를 보다 정확하게 활용할 수 있도록 도와준다. 여기 사용된 FCM 알고리즘은 2.2장 관련 연구에 기술되어 있다.

클러스터분석 결과 학습되어 형성된 클러스터들은 그 대표정보를 가지고 있으며 파일을 준비하고 처리하는 전문가는 파일유형이나 바이러스 유형 등의 관련정보를 부착시킨다. 이때 이러한 전문가의 정보는 시스템의 구축 목적과 밀접한 관련이 있으며 다른 전문가시스템과 마찬가지로 경험적 지식이 포함될 수 있다. 그러나 실세계에서 얻은 데이터를 클러스터 분석에 의하여 처리한 후 형성된 클러스터를 바탕으로 단순화 시킨 후 지식을 표현하므로 지식획득이 용이하다. 단지 정상 파일인지 바이러스 파일인지 만을 판정하려면 이를 지시하는 정보만 부착시킬 수도 있고 진단 후 대처할 수 있는 solution 함수를 호출 할 수도 있다. 이러한 정보는 단순하게 지식베이스 안에 규칙(Rule)의 형태나 객체(Object)형태로 표현할 수 있으며 모듈의 대략적인 구성은 (그림 1)의 위층에 표현되어 있다.

3.2 결정 상태 기반 진단 모듈

진단 모듈에서는 진단을 원하는 파일의 추출된 특징데이터를 가지고 지식획득모듈에서 사전지식으로 학습하여 가지고 있는 클러스터 정보를 참조하여 퍼지 소속 값을 계산한

다. 퍼지 소속 값은 FCM 알고리즘 구축과정에서 얻은 결과를 가지고 다음의 식(4)을 이용하여 계산한다.

$$u_{ij} = \frac{1}{\sum_{t=1}^{c} \frac{\| x_j - v_i \|^2}{\| x_j - v_t \|^2}} \quad (4)$$

주어진 데이터 x_j 에 대하여 식(4)를 이용하여 각 클러스터 i 에 대하여 u_{ij} (i=1,2, . . , c)를 구하면 x_j 가 각 클러스터 정보에 일치하는 정도를 알 수 있다.

다음 단계로서 u_{ij} (i=1,2, . . , c)를 통하여 주어진 데이터 x_j 가 이미 학습된 사전지식을 가지고 분류할 수 있는 척도를 측정하게 된다. 이를 주어진 데이터의 결정상태라고 정의하고 [정의1]에 의하여 퍼지 상태(fuzzy status)와 분명한 상태(crisp status)로 분류 하게 된다.

[정의 1] 다음 식(5)의 조건을 만족하면 주어진 데이터 x_j 는 이미 획득된 지식에 대하여 퍼지 상태(fuzzy status)인 것으로, 아니면 분명한 상태(crisp status)로 정의된다.

$$\frac{u_{lj} - u_{sj}}{u_{lj}} \le \frac{1}{c}, \ where\, u_{lj} = \max\{u_{ij}\}, \ u_{sj} = \max\{u_{ij}\}, \ s \neq l \quad (5)$$

[정의 1]은 주어진 데이터의 가장 큰 소속 값과 두 번째로 큰 소속 값의 차이가 충분히 크면 데이터를 판정하기에 필요한 지식을 이미 가지고 있는 상태(crisp, known)임을 말해주며 그렇지 않으면 주어진 데이터는 현재 학습한 정보로는 판단하기 어려운 상태(fuzzy status)임을 말해 주고 있다.

마지막 단계로 진단 모듈은 주어진 데이터의 진단상태가 crisp status이면 가장 큰 소속 값을 가지는 클러스터 l 에 부착된 정보를 출력해 주며, fuzzy status이면 경고를 보내며 전문가에게 분석을 요청한다. 전문가에 의해 분석된 데이터는 다음 FDS 갱신을 위한 데이터로 수집될 수 있다.

보통의 컴퓨터 바이러스 진단 시스템의 경우도 파일 수집, 특징패턴 추출, 데이터표현, 분류알고리즘 적용 등의 전 과정에 바이러스 전문가가 참여하고 단독시스템을 구축하는 것은 어려운 것으로 알려져 있다. FDS는 새로운 바이러스를 처리할 수 있는 학습기능과 퍼지이론, 전문가시스템의 설명기능을 도입하여 전문가시스템의 지식획득의 어려움도 해소하고 바이러스를 탐지한 후에 처리 하고 시스템을 갱신할 수 있는 방법도 제공하고 있다.

4. 실험 및 고찰

4.1 데이터 준비

FDS의 입력을 준비하기 위하여 VX heaven[14]으로부터 200개의 바이러스 파일과 윈도우시스템 실행파일로부터 200개의 정상파일을 수집하였다. 2.1장에서 소개한 역어셈블리 코드로부터 만들어낸 중간파일을 분석하여 특징패턴을 선정한다. 명령어 열(instruction sequence)의 출현빈도수와 바이러스 파일에 자주 나오는 명령어 열은 정상파일에는 잘 나오지 않는다는 휴리스틱을 이용하여 26개의 명령어 열을 특징패턴으로 선정하였다[11, 12, 13, 15]. 이와 같은 특징패턴 추출과정은 중요한 연구과제이고 시스템의 성능을 좌우하는 중요한 요소이나 본 논문에서는 그 개념을 도입하여 데이터 준비과정으로 활용하였다. 이제 마련된 특징패턴을 가지고 중간파일을 분석하여 각 특징패턴의 파일 안에서의 정규화된(normalized) 출현횟수를 구한다. 이렇게 만들어낸 400*26의 데이터는 본 논문에서 Ldata라고 부르며 FDS의 지식획득 모듈의 입력데이터로 사용되어 시스템의 사전정보를 구축 하게된다. 또한 진단에 사용될 실험 데이터 TdataI과 TdataII를 마련한다. TdataI은 사전정보구축에 사용된 Ldata로부터 100개의 바이러스 파일에 대한 데이터와 100개의 정상파일에 대한 데이터를 뽑아 200*26의 데이터로 구성되어 있다. TdataII는 Ldata를 만들기 위해 수집한 VX heaven과 윈도우시스템 실행파일로부터 100개의 정상파일과 100개의 바이러스 파일로부터 만든 200*26의 데이터로 구성되어 있다. 즉 TdataI은 사전정보구축에 사용된 데이터로부터 선택되어 이미 알려진 파일을 진단하는 것이고 TdataII는 같은 소스로부터 수집되었으나 FDS에 알려지지 않은 임의의 파일을 진단하는데 사용된다.

4.2 실험 및 고찰

지식획득 모듈의 입력을 위하여 준비된 Ldata를 만들어낸 파일은 2종류의 바이러스 파일과 유사한 종류의 정상실행파일로부터 수집하였으므로 클러스터의 수 c를 3으로 하였다. 그러므로 LdataI을 지식획득모듈의 입력으로 주면 3개의 클러스터를 형성하고 각각에 대하여 benign data, virus dataI, virus data II와 같은 단순한 정보를 부착하고 지식베이스에 저장하였다. 실세계 응용의 경우 전문가는 바이러스 유형정보나 바이러스 대처 요령이나 solution 함수로의 호출 등의 실제적인 정보를 부착하여 그 바이러스가 탐지되면 적절한 상담을 해주거나 전문가를 돕는 시스템이 될 수 있을 것이다.

이렇게 구성된 지식베이스를 가지고 TdataI의 200*26의 데이터를 진단모듈의 입력으로 사용하여 각 데이터의 세 개의 클러스터에 대한 퍼지 소속 값을 구하여 각 데이터의 진단상태를 구한다. 같은 범주의 데이터 셋에 대하여 100번 실험한 결과 평균 96.5%는 crisp상태로 3.5% 는 퍼지 상태로 판정되었다. crisp 상태로 판정된 데이터는 평균 97.8% 정확하게 분류되었으므로 사전정보 구축에 사용된 데이터를 진단하는 경우 94.4%의 정확도를 얻었다. 제안된 FDS가 미리 알려지지 않은 데이터에 대하여 어떻게 적응하는지 알아보기 위하여 TdataII의 200*26의 데이터를 진단모듈의 입력으로 사용하여 각 데이터의 진단상태를 구한다. TdataI의 경우와 마찬가지로 100번의 실험을 한 평균을 구하면 88.2%가 crisp status로 판정되고 9.8%가 지식획득모듈에서 가지고 있는 사전지식으로는 판정하기 어려운 상태인 것으로 나타났다. 이 때 crisp status로 판정된 경우 93.7%가 정확하게 분류되었으므로 TdataII의 경우 약 82.6%의 분류정확도를 얻었다. 이를 <표1>과 같이 요약하였다.

시스템의 성능은 준비된 샘플 데이터와 준비과정의 기법에 따라 달라지므로 <표 1>에 보여준 분류정확도는 제안된 시스템의 타당성을 보여줄 뿐 큰 의미는 갖지 않는다. 그러므로 마지막으로 기존의 분류를 위한 학습 알고리즘으로 많이 사용된 Support Vector Machine(SVM), Random Forest(RF), k-nearest neighbor(KNN)와 제안된 시스템 FDS를 비교하였다. 같은 데이터에 대하여 실험한 결과 <표 2>와 같은 분류 정확도를 나타내었다. <표 2>의 결과로부터 제안된 시스템 FDS는 컴퓨터바이러스 진단을 위한 시스템으로 도입될 수 있는 가능성을 확인할 수 있다.

〈Table 1〉 FDS 실험결과

	Crisp Status	Fuzzy Status	Accuracy
TdataI	96.5%	3.5%	94.4%
	97.8%		
TdataII	88.2%	9.8%	82.6%
	93.7%		

〈Table 2〉 기존 분류 알고리즘과의 비교

	KNN	RF	SVM	FDS
TdataI	89.7%	93.1%	93.9%	94.4%
TdataII	77.9%	82.1%	84.4%	82.6%

5. 결 론

1986년 컴퓨터 바이러스가 등장한 이래 매년 새롭게 발생하는 바이러스는 중요한 데이터와 프로그램을 가지고 있는 컴퓨터 기능을 위협하고 있다. 이에 미국 주요대학 컴퓨터 공학부 위원회에서는 컴퓨터 바이러스 감염을 정부가 가장 관심을 두고 해결해야할 문제점으로 지적하기도 했다. 컴퓨터 바이러스를 탐지하는 그동안의 연구는 바이러스 파일 특성과 시그내처 추출 및 매칭에 중점을 두었다. 이러한 고찰을 바탕으로 계속 변화하는 바이러스 유형에 대처하기 위하여 기존의 학습이론과 전문가 시스템의 연구와 접목시키는 시스템을 구상하게 되었다. 제안된 시스템 FDS는 지식획득 모듈에서 퍼지 클러스터 분석과정을 통해 실세계 데이터를 분석하여 클러스터를 형성하므로 손쉽게 지식베이스를 구축할 수 있으며 진단과정에서는 이미 구축된 사전정보를 활용할 수 있는 방법을 제공하며 결정 상태를 분류하여 처리하므로 효율적으로 처리할 수 있음을 확인하였다. 사전정보구축에 사용된 데이터와 새로운 데이터를 테스트 데이터로 준비하여 실험한 결과 각각 94.4%, 82.6%의 분류정확도를 얻었으며 널리 사용된 분류 알고리즘과도 비교하여 제안된 시스템 FDS의 타당성을 확인하였다.

이와 같은 시스템의 성능은 준비된 샘플데이터와 준비과정에서 사용한 파라메터 등에 따라 다른 결과를 나타내므로 다양한 실험데이터에 대한 체계적인 분석과정을 통해 수정보안되어야한다. 구축된 지식베이스에서 진단할 수 없는 새로운 파일의 경우 전문가에 의하여 판정되고 시스템에 추가적으로 학습될 수 있는 점증적 학습기법(incremental training method)이 도입되어야 할 것이다. 또한 실세계에서 다루는 대용량의 파일과 다양한 형태의 바이러스에 학습하고 적응할 수 있는 시스템으로 발전하기 위한 노력이 필요하다.

참 고 문 헌

[1] Mathew Braverman, "Windows Malicious Software Removal Tool : Progress Made, Trends Observed", Microsoft Antimalware Team, 2006.

[2] G. McGraw and G. Morisett, "Attacking malicious code: A report to the Infosec Research Council.", IEEE Software, pp.33-41, September/October 2000.

[3] V. Ke selj, F. Peng, N. Cercone, and C. Thomas, "N-gram-based Author Profiles for Authorship Attribution.", Proceedings of the Conference Pacific Association or Computational Linguistics, (PACLING'03), 2003.

[4] Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan, "Detection of New Malicious Code Using N-grams Signatures, Proceedings of the Second Annual Conference on Privacy, Security and Trust (PST'04), pp. 193-196, 2004.

[5] Abou-Assaleh, Nick Cercone, Vlado Keselj, and Ray Sweidan, "N-Gram based Detection of New Malicious Code", Proceeding of the 28th Annual International Computer Software and Applications Conference(COMPSAC'04), 2004.

[6] Kolter, J.Z., and Maloof, M. A., "Learning to detect malicious executables in the wild", In Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 470-478. New York, NY, 2004.

[7] I. Witten and E. Frank, "Data mining: Practical machine learning tools and techniques with java implementations", Morgan Kaufmann, San francisco, CA, 2000.

[8] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum press, New York, 1981.

[9] J. O. Kephart and W.C. Arnold, "Automatic Extraction of Computer Virus Signatures.", Proceedings of the 4th Virus Bulletin International Conference, R. Ford, ed., Virus Bulletin Ltd., Abingdon, England, pp. 178-184, 1994.

[10] J. O. Kephart, "A Biologically Inspired Immune System for Computers.", Proceedings of the 4th Workshop on Synthesis and Simulation of Living Systems, pp.130-139, 1994.

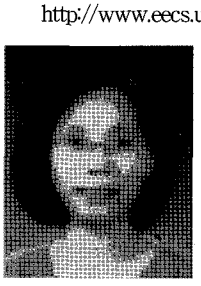
[11] Jianyong Dai, Joohan Lee and Morgan C. Wang, "Detecting Unknown Computer Virus Using Data Mining Techniques", Business Intelligent Symposium, poster presentation, April, 2006.

[12] http://www.datarescue.com

[13] Jian Yu and Miin-Shen Yang, "Optimality Test for Generalized FCM and Its Application to Parameter Selection", IEEE Transactions on Fuzzy Systems, Vol. 13, No. 1, Feb. 2005.

[14] VX Heaven : http://vx.netlux.org

[15] UCF Data Mining Research Group : http://www.eecs.ucf.edu/~jlee/dm



이 현 숙

email : hsrhee@dongyang.ac.kr

1989년 서강대학교 전자계산학과(학사)

1991년 포항공과대학교 컴퓨터공학과(석사)

1997년 서강대학교 컴퓨터학과(박사)

1991년~1997년 한국전자통신연구소(ETRI) 연구원

1997년~현재 동양공업전문대학 전산정보학부 부교수

관심분야 : 소프트컴퓨팅, 패턴인식, 데이터마이닝