퍼지 분류를 이용한 초기 위험도 예측 모델

홍 의 석† · 권 용 길††

요 약

소프트웨어 개발 초기 단계의 문제점이 개발 후반부 산물의 품질에 심각한 영향을 미치기 때문에 설계 명세를 이용하여 위험 부분을 예측하는 위험도 예측 모델은 전체 시스템 개발비용을 낮추는 데 중요한 역할을 하고 있으며, 이러한 예측 모델은 결과 산물이 매우 크고 실행 정확성이 요구되는 통신 소프트웨어 같은 실시간 시스템 설계에 더욱 필요하다. 판별분석, 인공신경망, 분류트리 등의 기법들을 이용한 모델들이 제안되었으나 이들은 결과에 대한 원인 분석의 어려움, 낮은 확장성 등의 문제점들을 지니고 있었다. 본 논문에서는 유전자 알고리즘에 의해 구축된 퍼지 규칙 베이스를 이용한 위험도 예측 모델을 제안한다. 제안 모델은 예측 결과에 대한 원인 분석이 용이하고 높은 확장성과 적용성을 지니고 규칙수에 대한 제한이 없다. 여러 내부 특성들 비교와 모의 실험을 통한 예측 정확도 비교를 통해 제안 모델이 타 모델들보다 우수함을 보였다.

Early Criticality Prediction Model Using Fuzzy Classification

Euy-Seok Hong† · Yong-Kil Kwon††

ABSTRACT

Criticality prediction models that determine whether a design entity is fault-prone or non fault-prone play an important role in reducing system development cost because the problems in early phases largely affect the quality of the late products. Real-time systems such as telecommunication systems are so large that criticality prediction is more important in real-time system design. The current models are based on the technique such as discriminant analysis, neural net and classification trees. These models have some problems with analyzing cause of the prediction results and low extendability. In this paper, we propose a criticality prediction model using fuzzy rulebase constructed by genetic algorithm. This model makes it easy to analyze the cause of the result and also provides high extendability, high applicability, and no limit on the number of rules to be found.

1. 서 론

소프트웨어 생산에서 분석이나 설계 단계와 같은 초기 단계의 중요성은 제작 기술이 발달함에 따라 더욱 중요해지고 있으며, 설계 단계를 정량화 하는 설계 메트릭들도 전체 시스템 개발 비용을 낮추는 데 중요한 역할을 하고 있다. 또한 설계 단계 산물들을 정량화하여 최종 개발물의 품질 인자들을 예측하는 예측 모델의 중요성도 매우 높다. 왜냐하면 초기 단계의 문제점이 개발 후반부 산물의 품질에 매우 심각한 영향을 미치고 대부분의 소프트웨어 결점들이 매우 적은 수의 모듈들로부터 발생한다고 알려져 있기 때문이다[1]. 이러한 설계 정량화와 예측 모델의 필요성은 결과 산물이 매우 크고 실행 정확성이 요구되는 통신 소프트웨어 같은 실시간 시스템 설계에 더욱 필요하다.

설계 단계 산물들의 복잡도 메트릭들을 이용하여 최

※ 본 연구는 1999년도 안양대학교 학술연구비 지원에 의해 수행되었음.
† 정 회 원 : 안양대학교 영상처리학과 전임강사
†† 정 회 원 : (주)오렌지씨씨 부설 인터넷연구소 전임연구원
논문접수 : 2000년 1월 28일, 심사완료 : 2000년 5월 15일

종 개발물의 품질 인자들을 예측하는 예측 모델들에 대한 연구들이 많이 행해졌다. 한 설계 개체의 위험도(criticality)란 개체가 구현되었을 때 갖는 결함경향성(fault-proneness)을 의미한다. 위험도 예측 모델이란 설계 개체를 입력으로 받아 그것이 결함경향 개체인지 아닌지를 판단하는 모델이다. 예측 모델의 입력은 설계 개체들을 정량화한 메트릭 벡터 형태가 되며 사용하는 메트릭들은 잘 알려진 복잡도 메트릭들이 된다.

기존에 제안된 위험도 예측 모델들은 많이 알려진 McCabe의 Cyclomatic Complexity[2]나 Halstead의 Software Science[3] 등의 복잡도 메트릭들로 구성된 메트릭 벡터들로 설계 개체들을 정량화 한 후 이들을 위험 그룹과 비위험 그룹으로 분류하는 분류 모델(Classification model)들이 대부분이었다. 분류 모델 기법으로 사용된 것은 판별분석법, 역전파 신경망, 분류트리 등이 있으며, 이들 중 역전파 신경망 이용 모델이 비교적 좋은 예측 결과를 보인다고 알려져 있다[4]. 하지만 이들 기법들은 분류 트리 기법을 제외하고는 모델 내부에서 행해진 수행 부분을 역추적하기가 거의 불가능한 블랙박스적인 모델이므로 위험도 결과의 원인을 분석하여 재설계 등의 조치를 취하기가 매우 어려운 문제점이 있다.

본 연구의 목적은 결과에 대한 원인 분석이 용이하고 분류 트리 기법이나 신경망 기법보다 좋은 성능을 보이는 분류 모델을 제안하는 것이다. 제안 모델은 퍼지 규칙 베이스를 사용하였으며 많은 규칙들을 유전자 알고리즘을 사용하여 최적화 하였다. 또한 퍼지 규칙에 정량적인 변수값들을 사용하지 않고 정성적인 값들을 사용함으로써 위험도 판정에 대한 결과분석이 용이하게 하였다.

위험도 예측 모델은 수천, 수만 LOC 정도의 시스템이 아니라 수십만 LOC 이상의 대형 시스템을 개발하는 경우에 유용하다. 수십개의 설계 개체로 구성된 시스템의 경우에는 설계자의 의미적인 판단으로 위험도가 높은 개체들을 선정할 수 있지만 수백개 이상의 설계 개체로 구성된 시스템의 경우에는 위험 부분을 찾는 자동화된 방법이 필요하기 때문이다. 그러므로 위험도 예측 모델에 관한 연구는 주로 큰 규모의 실시간 통신 시스템 등을 개발하는 개발 집단을 중심으로 행해져왔다. 본 논문에서 제안하는 모델의 입력 대상은 ITU-T의 표준안으로 널리 사용되고 있는 객체지향 실시간 시스템 명세 언어인 SDL(Specification and Description Language)[5]로 작성한 설계 명세이며 이를 정량화 하는 메트릭들은 본 연구의 선행 연구에서[6] 제안한 SDL 메트릭 집합을 사용한다.

2장에서는 기존에 제안된 위험도 예측 모델들을 살펴보고 그들의 장단점을 논의하며 3장에서는 본 연구의 목적을 만족시키는 새로운 예측 모델을 제안한다. 4장에서는 모의 실험을 통하여 분류 모델 중 가장 좋은 성능을 보인다는 역전파 신경망 모델과 제안 모델의 예측 정확도를 비교하고 5장에는 결론과 향후 연구 과제에 대해 기술한다.

2. 위험도 예측 모델

시스템 개발 초기 단계에서 위험도를 예측하기 위한 관련 연구들은 크게 두 가지로 구분할 수 있다. 첫 번째는 과거 프로젝트의 오류 자료들과 복잡도 메트릭 값들의 연관성을 파악하여 현재 수행중인 프로젝트의 위험도 예측을 정확하고 쉽게 해석할 수 있도록 하는 예측 모델을 만드는 것이다. 여기서 사용되는 모델들은 주로 입력 데이터들을 여러 개의 패턴으로 나누는 패턴 분류(pattern classification) 기법들을 사용한다. 이 모델들은 개체를 정량화하기 위해 각 개체를 기본 메트릭들의 벡터 형태로 만들고 실제 개발 과정을 통해 얻은 위험도가 높은 개체들과의 상관성을 모델에 학습시켜 새로운 프로젝트에 적용시키는 방법을 취한다. 대부분의 위험도 예측 모델에 대한 연구들은 이러한 형태를 취하지만 과거 프로젝트 자료를 보유하고 있는 개발 집단이 거의 없다는 것과 보유하고 있다하더라도 과거 프로젝트와 현재 프로젝트의 개발 환경, 개발 시스템의 특성이 매우 유사하여야 한다는 문제점이 있다[1].

두 번째는 프로그램의 위험도를 예측할 수 있는 메트릭들을 정의하고 그 타당성을 입증하여 정의한 메트릭들을 바탕으로 시스템의 위험도를 예측하는 것이다. 즉 이는 단순히 어떤 설계 개체가 위험한가 아닌가의 여부 결정이 아니라 실제 위험도 값을 정량화 한다. 그러므로 각 개체의 위험도를 서로 비교할 수 있다는 장점이 있다. 그러나 기존 데이터들을 통한 경험적인 지식이 아니라 하나의 복잡도 메트릭 값에 의존한다는 단점이 있다. 위험도 메트릭 제작은 위험도에 가장 관련이 많은 기본 메트릭을 하나 선정하여 예측에 사용할 수도 있고 위험도와 관련이 있는 기본 메트릭들을

조합하여 하나의 조합 메트릭 형태를 사용할 수도 있다. 후자가 위험도에 관련된 여러 요인들을 고려할 수 있을 것 같지만 기본 메트릭들을 조합하는 것은 매우 주의를 요한다. 여러 메트릭들을 조합함으로써 각 구성 요소들의 특성을 잃어버릴 가능성이 있기 때문이다[7].

두 가지 방향의 관련 연구들 모두 소프트웨어의 복잡도 메트릭이 프로그램의 오류의 분포와 관련이 있음을, 즉 복잡도가 높은 모듈일수록 오류가 발생할 가능성이 높다는 점을 가정하고 있다.

2.1 패턴 분류에 기초한 기법들

패턴 분류 기법에 기초한 예측 모델들의 예로는 판별분석법(Discriminant Analysis), 분류트리법(Classification Tree), 인공신경망(Neural Net) 이용 기법 등이 있다.

판별 분석은 다변량 통계분석의 한 방법으로서 Khoshgoftarr 등은 기존의 소프트웨어로부터 메트릭과 오류 자료 등의 데이터를 수집하여 이를 검증 데이터 집합(test data set)과 적합 데이터 집합(fit data set)으로 나눈 후, 각 메트릭들간의 상관 관계를 없애는 주성분분석을 거쳐서 적합 데이터 집합으로부터 판별 분류 함수를 구하였다[8, 9]. 그리고 이를 검증 데이터 집합에 적용하여 판별 분석이 신뢰도 예측에 유용함을 보였다.

최근의 지식 습득 도구나 기계 학습 시스템은 데이터로부터 판단트리(decision tree)를 구성하는 기능을 가지고 있다. 분류트리법은 높은 위험을 가지는 모듈들을 미리 정의된 메트릭에 기반하여 판단트리를 통하여 분류한다. 트리는 루트 노드에서 시작하여 레벨이 증가됨에 따라 정의된 메트릭에 의해 분기되며 말단 노드는 0, 1 또는 복수의 값을 지니고 있다. 말단 노드의 값을 가지고 어느 컴포넌트가 목표 클래스에 속하는지 여부를 과거 데이터에 기반하여 판단할 수 있다[10]. 분류트리법은 간단하고 설명 변수들의 상호 의존성을 어느 정도 고려할 수 있는 장점이 있지만 트리의 생성 과정에서 각 메트릭을 적용시키는 순서에 따라 동일한 질문이 반복될 가능성이 있고 트리의 형성에 있어서 상관이 없거나 덜 중요한 정보를 포함하는 경향이 있다[11].

인공신경망 프로그래밍 도구와 시스템의 발달로 신경망 모델이 신뢰도 예측에 적용 가능함을 보여주는 연구들이 수행되었으며 Khoshgoftaar 등은 신경망이 결함경향 모듈을 식별할 수 있도록 학습하는데 역전파(backpropagation) 알고리즘을 사용하였다[12]. 과거 프로젝트 데이터로 학습된 신경망 모델의 입력은 각 설계 개체를 정량화한 벡터의 구성 메트릭 뉴런들로 이루어지며 출력 뉴런은 입력 개체의 위험도를 결정한다.

위의 세가지 신뢰도 예측 모델 기법은 과거 프로젝트 데이터 즉 모델을 훈련시킬 수 있는 훈련 데이터 집합을 반드시 필요로 하며 모델이 만들어진 환경과 다른 개발 환경에서는 그 모델을 적용하여 위험도를 예측하기 어렵다. 대부분의 개발 집단은 훈련 데이터 집합을 보유하고 있지 않으므로, 이러한 문제점을 해결하기 위해 선행 연구로서 훈련 데이터 집합이 필요없는 모델을 클러스터링 기법을 이용하여 제안하였으나 모델의 유용성 측면에서 많은 문제가 있었다[13]. 훈련 데이터 집합이 반드시 필요하다는 문제 외에도 기존의 방법들은 위험 결과에 대한 원인 분석이 거의 불가능하다는 문제점을 안고 있다. 이런 문제점은 기본적으로 결과 위주(result-driven)의 접근 방법을 사용하는 신경망 모델이 가장 심각한데 이를 해결하는 것이 본 연구의 중요한 목적이다.

2.2 위험도 메트릭 기법들

Zage 등은 프로그램 모듈의 복잡도를 외부 복잡도와 내부 복잡도로 나누고 이들의 합을 모듈의 복잡도로 정의하였으며 계산된 복잡도를 가지고 극단 이상점을 제외한 복잡도의 평균보다 1 표준 편차가 큰 곳을 경계로 하여 결함경향 여부를 판단하였다[14].

Agresti 등은 기존의 관련 연구들이 만들어진 생산물의 특성에만 의존함을 지적하고 개발 프로세스의 특성을 고려하여 코드의 재사용성, 개발 경험, 검증 방법의 효율성 등을 포함한 메트릭들을 제시하고 그것이 신뢰도와 높은 연관성이 있음을 보였다[15].

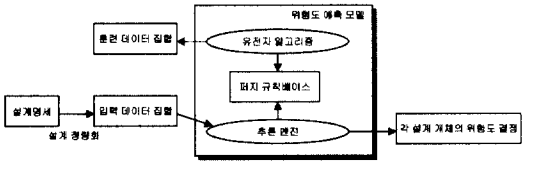
데이터 바인딩 기법은 시스템의 구성 요소들간의 상호 작용의 성질을 파악하기 위하여 데이터 바인딩이란 관점에서 시스템의 결합도과 연결 강도를 측정하는 것이다. Selby 등은 데이터 바인딩 분석을 통하여 오류가 많은 모듈들과 적은 모듈을 구분함으로써 데이터 바인딩에 의한 분석이 시스템의 결함 분석에 효과적임을 보여주었다[16].

선행 연구에서는 SDL 설계 개체의 기본 메트릭들로부터 위험도를 측정하는 두가지 형태의 혼성 복잡도 메트릭들을 정의하고, 이들이 패턴 분류 방법들 중 가

장 좋은 결과를 보인다고 알려진 역전파 신경망 모델의 성능에 근접한 성능을 나타냄을 보였다[13].

3. 제안 모델

본 논문에서 제안하는 위험도 예측 모델의 개략적인 형태는 (그림 1)과 같다. 훈련 데이터 집합이란 과거의 매우 유사한 개발 프로젝트에서 얻은 오류 데이터(입력 데이터와 해당 오류 정보에 대한 쌍)들의 집합이며 입력 데이터 집합은 현재 개발 프로젝트에서 설계 결과를 메트릭 벡터 형태들로 정량화한 데이터 집합이다.



(그림 1) 제안 모델 형태

훈련 데이터 집합에 기초하여 가장 적합한 퍼지 규칙들을 찾아내기 위해 유전자 알고리즘을 사용한다. 퍼지 규칙들은 설계 전문가들의 경험 지식으로 작성할 수 있지만 규칙이 매우 많은 경우에는 이를 자동화하여 최적화된 규칙들을 구하는 것이 훨씬 바람직하므로 유전자 알고리즘을 사용하였다. 제작된 퍼지 규칙베이스를 이용하여 추론 엔진은 입력 설계 개체의 위험도를 결정한다. 또한 퍼지 규칙에 사용되는 변수들은 실수 형태의 메트릭 값의 형태가 아니라 여러 단계를 나타내는 정성적인 값들이므로 결과에 대한 원인 해석이 용이하다.

3.1 퍼지 규칙 베이스

퍼지 전문가 시스템의 가장 중요한 구성 요소인 규칙들은 사람이 직접 찾아낼 수 있다. 이 경우 얻어진 규칙이 주관적이고 항상 동일한 결과를 내지 못할 뿐만 아니라 방대한 시간을 필요로 한다. 따라서 규칙베이스 구축의 자동화가 필요하다. 규칙베이스 자동화에 관해 많은 연구가 집중되어 왔다. Baisch 등은 소프트웨어 품질을 예측하는데 인공 신경망을 사용해서 규칙베이스를 구축하였다[17].

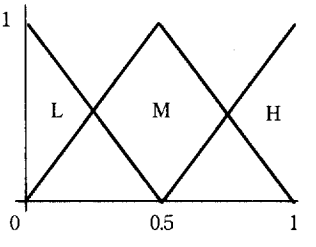
본 연구는 초기 위험도를 예측하는 퍼지 전문가 시스템을 구축하고자 한다. 이 시스템은 소프트웨어 부품의 설계 정보에서 뽑아낸 메트릭 측정치를 입력으로 하고 부품의 위험도를 출력으로 한다. 규칙 베이스의 한 예는 다음과 같다.

> 만약 M_1이 크고 M_2가 작고 M_3이 보통이고 M_4가 작고 M_5가 작고 M_6이 크면 부품 C_1의 위험도는 크다.
> (if M_1 is High and M_2 is Low and M_3 is Medium and M_4 is Low and M_5 is Low and M_6 is High, then the Criticality of the component C_1 is High.)

(그림 2) 규칙 베이스의 예

C_1은 하나의 소프트웨어 설계 개체를 나타내며 이는 기본 메트릭 M_i로 구성된 메트릭 벡터 (M_1, M_2, M_3, M_4, M_5, M_6)로 정량화 된다. 위의 규칙베이스는 메트릭 벡터 (M_1, M_2, M_3, M_4, M_5, M_6)로부터 위험도로의 대응을 추론할 수 있는 규칙들로 구성되어 있다.

본 연구에서는 초기 위험도를 예측하는 퍼지 전문가 시스템의 규칙베이스를 구축하는 작업을 유전자 알고리즘을 사용하여 자동화하였다. 입력 변수인 메트릭 벡터값을 0과 1사이의 값으로 표준정규화한 후, 다음과 같은 삼각형 소속함수를 사용하여 퍼지값으로 변환하였다.



(그림 3) 소속 함수

추론과정에서는 다음 식을 이용하는 무게 중심 비퍼지화 기법과 MAX-MIN 추론법을 사용하였다.

$$C = \frac{\sum_{i=1}^{n} p_i \cdot y_i}{\sum_{i=1}^{n} y_i} \quad \begin{cases} n : \text{규칙의 수} \\ p_i : \text{소속함수의 중앙값} \\ y_i : \text{규칙 } i\text{의 소속값} \end{cases}$$

3.2 유전자 알고리즘

3.2.1 염색체 기호화

메트릭 측정치를 Low(L), Medium(M), High(H)로 정성화하고 위험도를 Low(L), High(H)로 정성화 한다면 6차원 메트릭 측정 벡터에서 위험도로의 함수는 다

음과 같이 표현된다.

```
LLLLLL     -> L
LLLLLM     -> H
LLLLLH     -> L
LLLLML     -> L
LLLLMM     -> H
LLLLMH     -> L
LLLLHL     -> L
LLLLHM     -> H
LLLLHH     -> L
...
HHHHHH     -> L
```

(그림 4) 메트릭 측정 벡터에서 위험도로의 대응의 한 예

(그림 4)의 각 문장은 규칙베이스를 구성하는 한 개의 규칙을 의미하며 첫 번째 문장은 개체의 모든 메트릭 값이 L일 때 그 개체의 위험도가 낮다는 것을 의미한다. 6개 벡터 각각이 3개의 값(L, M, H)을 가질 수 있으므로 729개의 규칙 문장들이 모여 한 개의 대응 함수 F를 구성한다. 이 대응 함수 F는 메트릭 벡터로부터 위험도로의 대응을 나타낸다.

$$F : (M_1, M_2, M_3, M_4, M_5, M_6) \rightarrow C$$
$$\text{s.t. } M_i \text{ in } \{H, M, L\},\ C \text{ in } \{H, L\}$$

다음은 위의 대응 함수를 위치기반(Locus-based) 방법으로 염색체로 기호화한 것이다.

0	1	0	0	1	0	0	1	0	...	0
1										729

(그림 5) 염색체의 구조

729개의 염색체의 각 유전자는 (그림 5)에서 각 문장의 위험도를 순서대로 나열한 것이다. 위험도는 이진화해서 높음(High)을 1로 낮음(Low)을 0으로 표현했다.

3.2.2 유전자 알고리즘 인자

룰렛 바퀴 선택(roulette wheel selection)법과 균일 교배(uniform crossover)법을 사용했으며 교배 확률과 변이 확률은 각각 0.6, 0.015로 하였다. 모집단의 크기는 100이며 4십만 세대까지 실행시켰다. 염색체의 우열을 가리는 적합도(fitness)는 다음과 같이 계산하였다.

$$F = \frac{1}{(E_1 + kE_2)} \quad (\text{단, } k > 1)$$

E_1 : 위험하다고 판단된 안전한 모듈의 수
E_2 : 안전하다고 판단된 위험한 모듈의 수

여기서 1보다 큰 상수 k를 주어 E_2의 수가 많은 염색체를 보다 불리하게 간주한 이유는 안전하다고 판단된 위험한 모듈을 처리하는 것이 위험하다고 판단된 안전한 모듈을 처리하는 것보다 훨씬 더 많은 비용이 들기 때문이다. 본 모델을 검증하는 모의 실험에서는 k를 2로 하여 실험하였다.

3.3 제안 모델의 정성적 평가

<표 1>은 기존의 예측 모델들을 서로 비교한 것이다. 모델 구축 비용면에서 역전파 신경망 모델은 해를 얻기까지 수렴 기간을 거쳐야 하는데 이 훈련 과정은 많은 시간을 필요로 한다. 훈련 자료의 필요성 측면에서 퍼지 분류를 제외한 나머지 모델은 모두 훈련 자료를 필요로 한다. 퍼지 분류의 경우 전문가의 사전 지식을 규칙베이스에 추가할 수 있기 때문에 세모로 표시했다. 결과 분석의 가능성 측면에서 역전파 신경망은 내부를 들여다 볼 수 없는 블랙박스 구조이기 때문에 원인 분석이 어렵다. 반면 퍼지 분류나 분류 트리의 경우, 그 내부가 사람이 이해하기 쉬운 형태로 되어 있기 때문에 원인 분석이 용이하다. 다른 프로젝트 자료를 적용했을 때의 예측률을 뜻하는 이식성은 퍼지 분류가 우수한 것으로 알려져 있다[18].

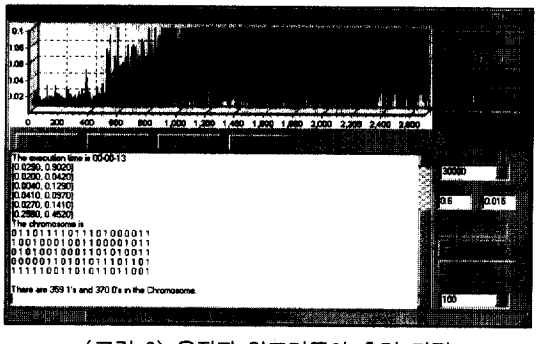
<표 1> 예측 모델들의 정성적 비교

특 징	판별 분석	분류 트리	퍼지 분류	역전파 신경망*
모델 구축비용	낮음	보통	보통	높음
훈련 자료의 필요성	O	O	△	O
결과 분석 가능 정도	보통	높음	높음	낮음
모델 이해의 용이도	보통	보통	높음	낮음
다른 프로젝트 자료로의 이식성	보통	보통	높음	낮음

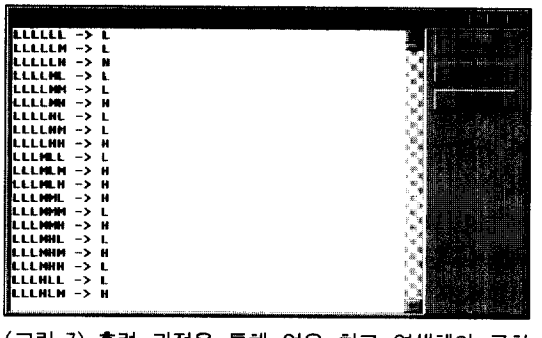
4. 모의 실험

제안 모델의 유용성을 검증하기 위해 기존 분류 모델 중 가장 우수하다고 알려진 역전파 신경망 모델과 예측 정확도를 비교하는 실험을 행하였다. 설계 개체 유형으로는 SDL 시스템 분석 단계에서의 블록을 사용하였으

며 하나의 블록을 정량화하는 메트릭 벡터는 (BRS, EBS, EBC, BP, BS, BR)[1]이다[6]. 선행 연구[13]와 유사한 제약 조건을 가진 데이터 집합을 제작하였으며 각 블록의 위험도는 SDL을 이용한 통신 소프트웨어 시스템의 설계 경험이 있는 두명의 소프트웨어 공학자에 의해 결정되었다.



(그림 6) 유전자 알고리즘의 훈련 과정



(그림 7) 훈련 과정을 통해 얻은 최고 염색체의 규칙 베이스

역전파 신경망 모델은 학습률로 0.05, 모멘텀으로 0.2를 사용하여 학습시켰다. 총 7개의 검증 자료 집합을 사용하였으며 집합의 크기는 각각 10, 50, 100, 300, 500, 1000, 3000이다. 그중 크기가 300인 집합을 비교하려는 두 모델의 훈련 데이터 집합으로 사용하였다. 훈련 과정의 효율성과 모델의 정확도 향상을 위해 훈련 데이터 집합의 블록들 중 위험도 결정이 애매한 경계 부분에 있는 블록들의 메트릭 값들을 수정하여 확실한 위험도를 가지게 하였다. 따라서 훈련 데이터 집합은 모델 훈련 과정에 의해 정확한 규칙 베이스로 구축되었으며 구축된 규칙 베이스와 훈련 데이터 집합은 거의 유사하였다. 또한 역전파 신경망 모델 역시 빠른 시간 내에 훈련되었다. 이 실험은 분류력을 비교하기 위한 것으로 제안 모델이 다른 모델과 비교하여 얼마나 잘 정확한 정보를 이끌어 내는가를 확인하는 것이다.

1) 개체 정량화에 사용된 메트릭들에 대한 설명은 [6]을 참조하기 바란다.

〈표 2〉 제안 모델의 예측 정확도 검증

검증자료 (E1, E2)	10 (10,0)	50 (47,3)	100 (279,29)	300 (279,29)	500 (457,43)	1000 (917,83)	3000 (2725,275)
역전파 모델	1,0	0,0	1,2	1,4	10,11	18,27	60,102
제안 모델	2,0	4,1	2,0	0,0	1,0	0,2	0,4

E1은 안전한 블록의 수이고 E2는 위험한 블록의 수이다. 예를 들어 크기가 10인 집합은 모두 안전한 블록으로 구성되어 있으며 크기가 50인 집합은 3개의 위험한 블록을 포함하고 있다.

쉼표(,)로 분리되어 표시된 수는 모두 모델이 오판한 블록의 수를 의미한다. 그 중 쉼표 앞의 수는 안전한 블록을 잘못 판단한 경우를 의미하며 쉼표 뒤의 수는 위험한 블록을 잘못 판단한 경우를 의미한다. 예를 들어 역전파 모델은 크기 500인 집합에 대해 457개의 안전한 블록 중 10개는 위험하다고 판단했으며 43개의 위험한 블록 중 11개는 안전하다고 판단했다.

300개 이하의 집합에 대해서는 두 모델이 비슷한 예측률을 보이고 있다. 하지만, 500개 이상의 집합에 대해서는 제안 모델의 예측 정확도가 현저하게 높은 것을 알 수 있다. 500, 1000, 3000 등으로 집합의 크기가 커질수록 역전파 모델은 선형적인 판단 오류 증가를 보이고 있는 반면 제안 모델은 판단 오류 증가가 거의 없다. 그 이유는 훈련 집합으로부터 위험도를 보다 정확하게 예측할 수 있는 규칙을 얻어내었기 때문이라고 생각된다.

5. 결 론

소프트웨어 산업이 점차 발전하고 대형 소프트웨어 개발 프로젝트가 진행됨에 따라 초기 위험도 예측 모델은 효율적인 자원 할당과 재설계 부분의 자동 결정에 사용되므로 시스템 개발 비용을 낮추는 데 큰 몫을 하고 있다.

기존의 예측 모델들은 판별 분석, 분류 트리, 역전파 신경망 등을 이용한 분류 모델들이었다. 모델 훈련 비용과 결과에 대한 원인 분석의 어려움 등의 문제점들이 존재하였다. 본 논문에서 제안한 모델은 유전자 알

고리즘에 기초한 퍼지 분류 모델로 이러한 문제점들을 해결하였다. 모델의 평가 항목들을 선정하여 기존 모델들과 정성적인 평가를 행하였으며 기존 모델들 중 예측 정확도 면에서 가장 우수하다는 역전파 신경망 모델과 예측 정확도 비교를 통하여 우수함을 보였다.

본 논문에서 제안한 예측 모델은 다른 감독형 모델에 비해 상당한 수준의 예측율을 보이고 있으며 찾으려는 규칙의 수에 제한이 없다. 본 모델은 퍼지 분류에 속하기 때문에 결과에 대한 원인 분석이 용이하며 내부 구조가 if-then 구조의 퍼지 규칙으로 구성되어 있기 때문에 이해하기 쉽고 규칙의 추가가 간단하므로 확장성이 용이하다.

향후 과제로는 좀더 예측률이 높은 규칙베이스를 얻을 수 있도록 유전자 알고리즘에서 사용된 염색체 기호화를 개선시키고, 퍼지 규칙 베이스의 다양한 추론 방법들을 연구해야 할 것이다.

참 고 문 헌

[1] N. Ohlsson and H. Alberg, "Prediction Fault-Prone Software Modules in Telephone Switches," *IEEE Trans. Software Eng.*, Vol.22, No.12, pp.886-894, Dec. 1996.

[2] T. J. McCabe, "A Complexity Measure," *IEEE Trans. Software Eng.*, Vol.2, No.6, pp.308-320, Dec. 1976.

[3] M. H. Halstead, *'Elements of Software Science'*, Elsevier North-Holland, New York, 1977.

[4] T. M. Khosgoftaar, D. L. Lanning, and A. S. Pandya, "A Comparative Study of Pattern Recognition Techniques for Quality Evaluation of Telecommunications Software," *IEEE J. Selected Areas in Commun.*, Vol.12, No.2, pp.279-291, Feb. 1994.

[5] J. Ellsberger, D. Hogrefe, and A. Sarma, *'SDL - formal object-oriented language for communicating systems'*, Prentice Hall, 1997.

[6] 홍의석, 홍성백, 김갑수, 우치수, "SDL 설계 복잡도 메트릭 집합", 정보과학회논문지(B), 제24권 제10호, pp.1053-1062, 1997.

[7] N. Fenton, "Software Measurement : A Necessary Scientific Basis," *IEEE Trans. Software Eng.*, Vol. 20, No.3, pp.199-206, March 1994.

[8] T. M. Khosgoftaar and E. B. Allen, "Early Quality Prediction : A Case Study in Telecommunications," *IEEE Software*, Vol.13, No.1, pp.65-71, Jan. 1996.

[9] J.C. Munson and T.M. Khoshgoftarr, "The Detection of Fault-Prone Program," *IEEE Trans. Software Eng.*, Vol.18, No.5, pp.423-433, May 1992.

[10] A. A. Porter and R. W. Selby, "Empirically Guided Software Development Using Metric Based Classification Trees," *IEEE Software*, Vol.7, No.3, pp.46-54, March 1990.

[11] L.C. Briand, V.R. Basili and C.J. Hetmanski, "Developing Interpretable Models with Optimized Set Reduction for Identifying High-Risk Software Components," *IEEE Trans. Software Eng.*, Vol.19, No.11, pp.1028-1044, Nov. 1993.

[12] T. M. Khoshgoftaar and D. L. Lanning, "A Neural Network Approach for Early Detection of Program Modules Having High Risk in the Maintenance Phase," *J. Systems Software*, Vol.29, pp.85-91, 1995.

[13] EuySeok Hong and ChiSu Wu, "Criticality Prediction Models using SDL Metrics Set," pp.23-30, *Proc. APSEC'97/ICSC'97*, 1997.

[14] W. M. Zage and D. M. Zage, "Evaluating Design Metrics on Large-Scale Software," *IEEE Software*, pp.75-80, July 1993.

[15] W. W. Agresti and W. M. Evanco, "Projecting Software Defects From Analyzing Ada Designs," *IEEE Trans. Software Eng.*, Vol.18, No.11, Nov. 1992.

[16] R. W. Selby and V. R. Basili, "Analyzing error-prone system structure," *IEEE Trans. Software Eng.*, Vol.17, No.2, pp.141-152, Feb. 1991.

[17] E. Baisch and C. Ebert, "Intelligent Prediction Techniques for Software Quality Models," *the 1996 ACM Symp. on Applied Computing*, 1996.

[18] C. Ebert, "Evaluation and Application of Complexity-Based Criticality Models," *Proc. of METRICS '96*, pp.174-185, 1996.

홍 의 석

e-mail : hes@aycc.anyang.ac.kr

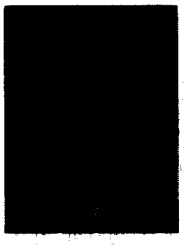
1992년 서울대학교 계산통계학과 졸업(학사)

1994년 서울대학교 대학원 계산통계학과(이학석사)

1999년 서울대학교 대학원 전산과학과(이학박사)

1999년~현재 안양대학교 영상처리학과 전임강사

관심분야 : 메트릭 기반 소프트웨어 품질 예측 모델, 웹 기반 멀티미디어 응용 기술 등



권 용 길

e-mail : mail@kwonyongkil.pe.kr

1997년 서울대학교 계산통계학과 졸업(학사)

1999년 서울대학교 대학원 전산과학과(이학석사)

1999년~2000년 삼성전자 미디어 컨텐트 사업팀 연구원

2000년~현재 (주)오렌지씨씨 부설 인터넷연구소 전임 연구원

관심분야 : 분산 객체 시스템, 진화 연산 등