

신경망 기반의 소프트웨어 개발노력 추정모델 구축에 관한 연구

김병관*, 백승익**
삼성 SDS*, 한양대학교**

Software Development Effort Estimation Using Neural Network Model

Byungwan Kim*, Seung Baek**
Samsung SDS*, Hanyang University**
byungwan.kim@samsung.com, sbaek@hanyang.ac.kr

Abstract

소프트웨어 개발노력 추정에 대한 연구는 소프트웨어가 복잡해지고 범위가 크게 증가함에 따라서 그 중은 지속적으로 부각되고 있다. 관련 프로젝트를 발주하는 업체나, 이를 수주하고 개발을 진행하는 업체에게 원가를 고려하는 측면에서 매우 중요한 부분을 차지하고 있다. 이러한 개발노력 추정을 위하여 다양한 접근 방식들이 고려되어지고 있는데, 그중에서 많이 활용되어지고 있는 방식은 소프트웨어 규모에 기반을 둔 LOC(Line Of Code) 기반 COCOMO (Constructive Cost Model) 모델이나, 기능점수(Function Point)를 기반으로 한 회귀분석 모델, 인공지능(Artificial Intelligence)을 활용한 신경망(Neural Network) 모델, 사례분석기법(CBR, Case Based Reasoning) 등이 있다. 이 중에서 최근에 기능점수를 활용한 개발노력 추정에 관한 연구들이 활발히 진행되고 있으나 개발노력 추정에는 소프트웨어 규모의 척도인 기능점수 뿐만 아니라, 개발환경을 구성하는 여러 가지 측면에 대한 고려가 추가되어야 한다. 이에 본 논문은 최신의 소프트웨어 개발 사례들에 대하여 기능점수 및 추가적인 개발환경 요소들을 면밀히 분석하고, 분석한 내용에 대해서 전문가들의 설문을 통한 빈도분석 및 로지스틱 회귀분석, 데이터마이닝 기법인 신경망 분석 등을 활용하여 개발노력 추정 모델을 구축함으로써, 소프트웨어 개발의 다양한 측면의 중요성을 강조하고, 정확한 추정의 방안을 제시 하고자 노력 하였다.

1. 서론

소프트웨어 개발 시 중요하게 제기되는 문제점 중에 하나는 개발 초기 단계에 개발과 관련된 노력과 비용을 정확하게 산정할 수 있는 기업의

능력이다. 정확하지 못한 개발노력의 추정은 기업의 자원의 낭비와 프로젝트의 비효율성을 불러일으키게 되고, 결과적으로 프로젝트의 실패를 가져오게 된다. 이런 개발노력 추정에 대한 중요성 때문에 소프트웨어 개발노력 추정

분야는 과거 30여 년 동안 활발한 연구가 수행되었으나, 소프트웨어 개발 노력과 비용에 영향을 미치는 요인과 그들 간의 관계가 아직 불명확한 관계로 구체적인 노력과 비용 추정 모델이 없는 실정이다. 현재까지 개발되어져서 사용되고 있는 많은 모델들은 소프트웨어의 규모 측정을 통하여 노력과 비용을 간접적으로 추정하는 모델이며, 그 중에서 가장 널리 사용되어 왔던 모델로는 LOC (Line Of Code)를 기반으로 하는 Boehm (1981; 1984)의 COCOMO (Constructive Cost Model) 모델과 기능점수분석 (FPA, Function Point Analysis) (Albrecht, 1979; Albrecht & Gaffney, 1983)을 기반으로 하는 다양한 모델들을 들 수 있다. 그러나, 이러한 모델들은 최근의 다양한 개발환경 요소들인 프로그래밍 언어, 하드웨어, 소프트웨어 엔지니어링 방법론, 전송방식과 네트워크 기술들까지를 고려하고 있지는 못한 실정이다 (Venkatachalam, 1993). 따라서 소프트웨어 및 정보시스템의 복잡도가 급속도로 증가하고, 다양한 패키지 제품들의 사용으로 인한 개발 편의성 증가 또는 Web 시스템 등과 같은 새로운 개발환경들의 대두와 프로젝트를 둘러싸고 있는 기타 외부적 환경 등 소프트웨어 개발에 영향을 미치는 다양한 요소들을 추가적으로 고려하는 새로운 개발노력 예측기법이 필요한 상황이다. 그 동안 정보기술의 수요 증가 추세에도 불구하고, 소프트웨어 개발노력을 계산하고 합리적인 비용을 산정하는 방법이 모호했다. 이에 따라 발주기관은 정보화 예산을 편성하는데 많은 어려움을 겪고, IT 업체들도 적정 개발노력을 추정하는데 난항을 겪는 등 소프트웨어 개발노력 추정 자체가 기준이 사실상 부재했다.

따라서 본 연구에서는 새로운 소프트웨어 개발 환경에 적합한 개발노력 추정 모델을 제시하고자 노력하였다. 특히, 기존의

소프트웨어 개발노력 추정 기준인 '본수' 방식과 '기능점수' 방식에서 한 걸음 더 나아가서 데이터 마이닝 기법 중에 하나인 신경망 시스템을 활용하여 개발노력을 추정하고자 하였다. 본 연구에서는 개발노력 추정에 영향을 미치는 요인으로 기존의 기능점수에 추가적으로 소프트웨어 개발 생산성에 영향을 미치는 보정항목들을 타 연구 모델로부터 참조하였고, 또한 정보통신부의 "소프트웨어 사업대가의 기준" 내의 생산정보정계수 항목들과 개발노력과 관련되어질 수 있는 추가 항목들도 고려하였다. 이런 여러 개의 항목 중에서 개발노력 추정을 위하여 사용되어질 항목을 선정하기 위하여 국내 64명의 프로젝트 관리자 (Project Manager)의 설문과 통계적 분석을 통하여 소프트웨어 개발노력에 영향을 주는 항목들을 선별하였다.

2. 관련 연구

소프트웨어 개발에 드는 비용이 높아져서 소프트웨어 개발이 기업 정보시스템 관련 예산에서 주요한 원가 요인이 되고 있다. 가트너 그룹이 발표한 2003년도 소프트웨어 시장 규모는 7,546억불 규모이며, 그 중에서 소프트웨어 개발과 연관된 IT 서비스 영역인 컨설팅, SI (System Integration), IT 아웃소싱, BPO (Business Process Optimization), 하드웨어/소프트웨어 유지보수에 추정되는 소요 비용은 약 5,900억불에 달하는 것으로 예측되고 있다 (Gartner Group, 2002). 따라서 매년 많은 비용이 발생하는 이와 같은 소프트웨어 개발 프로젝트가 시작되기 전에 관련된 비용 및 이익에 대한 추정을 세심히 고려해야 할 필요성이 있다. 소프트웨어 개발비용 추정의 정확성은 기업이나 단체의 소프트웨어 투자관련 의사결정에 직·간접적으로 중대한 영향을 미친다. 이러한 소프트웨어의 비용 산정을

위하여 다양한 방법이 시도되어 왔다. 주요한 접근 방식은 소프트웨어 개발노력과 비용을 소프트웨어의 규모를 통하여 간접적으로 측정하는 것이다. 이러한 접근 방식에 대한 다양한 연구가 진행되어 왔는데, 주요한 모델들은 LOC 기반으로 한 Boehm (1982; 1984)의 COCOMO 모델과 기능점수 기반의 알고리즘적 추정 모델로 Albrecht & Gaffney (1983) 모델, Matson et al. (1994)의 모델 등이 있으며, 기타 새로운 추정방법으로 기존의 개발노력 추정 데이터를 근거로 하여 개발노력을 추정한 사례기반분석(Case-based Reasoning) 모델 (Mukhopadhyay, 1992) 등이 있다.

2.1 LOC 기반 모델

소프트웨어 개발노력을 추정하기 위해서는 우선 소프트웨어의 규모를 측정해야 할 것이다. 직접 소프트웨어 코드의 라인 수를 기반으로 하여 소프트웨어의 개발노력을 추정하기 위해 최초로 개발된 소프트웨어 척도 중 하나가 LOC (Line of Code) 이다. 여러 개의 LOC 기반의 개발노력 추정모델 중에서 가장 많이 알려진 모델이 Boehm (1982)의 COCOMO이다.

LOC 기반의 개발노력 추정 모델은 측정하기가 쉽고, 이해하기가 쉬운 장점이 있는 반면에 개발에 사용되어질 언어에 따라서 LOC가 다르게 추정되어질 가능성이 있어서 일관적인 개발노력 추정이 힘들다는 단점이 있다 (이상운, 2002; 이상운 외 2인, 2002). 이 밖에도 소프트웨어의 개발이 종료되기 전에는 정확한 LOC 추정이 어렵다는 단점을 지니고 있다. 특히 근자에 들어서 소프트웨어의 기능이 복잡화되면서 단지 코드의 길이만을 고려하여 개발노력을 추정한다는 것은 매우 비현실적이 되었다.

2.2 기능점수 기반 모델

프로젝트의 규모 측정 방식으로서 LOC의 단점을 극복하기 위한 대안으로 프로젝트의 복잡도와 기능성에 대한 광범위한 연구가 이루어졌다. 복잡도를 측정하는 척도로는 Halsted's Software Science와 McCabe's Cyclomatic Number가 있으며 개발 초기에 계산될 수도 있지만 대부분의 복잡도 척도가 LOC에 기초하고 있다. 기능성 척도로는 시스템의 기능성으로 소프트웨어 규모를 측정하는 FPA (Function Point Analysis)와 Demarco's Bang Metrics가 있다. 과거 20년 이상 기능적 규모 척도가 소프트웨어공학 분야에서 연구되었으며, Bang Metrics에 대한 많은 연구를 수행하였으나 상업적으로 널리 사용되지 못하였는데 비해 FPA는 개발과정 초기에 프로그램의 기능적인 측면에서 소프트웨어 생산성을 측정하기 위한 척도로 IBM에 근무하는 Albrecht (1979)가 제안하였다. FPA는 사용자에게 양도될 시스템의 기능에 기초하여 소프트웨어의 규모와 복잡도를 정량화하는 방법이다. 이 기법은 소프트웨어 프로젝트를 개발하기 위해 사용되는 언어 또는 도구와 독립적이며, 개발 생명주기의 초기단계인 요구분석 단계에서 측정 가능한 장점이 있다.

기능점수를 이용한 개발노력 추정에 관한 연구로서, Albrecht et al. (1981)은 IBM Data Processing Services에서 개발된 응용 프로그램에 대해 개발노력과 기능점수 식의 선형형태를 취함을 연구하였고, Matson et al. (1994)은 이들 데이터에 대해 개발노력이 기능점수에 대해 식의 비선형 형태를 취하는 것이 보다 정확한 추정을 할 수 있음을 보였다.

Venkatachalam (1993)이 지적하였듯이

기능점수 방식에서의 문제점은 현재의 프로그래밍 언어, 하드웨어, 소프트웨어 엔지니어링 방법론, 커뮤니케이션, 네트워크 기술 등의 발전하는 기술들을 반영하고 있지 못하며, 또한 소프트웨어의 개발노력 추정을 위하여 기능점수가 고려하고 있지 않은 많은 요소들이 있음이 지적되어지고 있다. Mukhopadhyay (1992)은 전문가의 관찰에 의해서 추정능력이 질적으로 향상된다고 하였으며, Vicinanza et al. (1991)는 경험이 많은 소프트웨어 개발 관리자에 의해서 추정능력이 또한 향상될 수 있다고 제안하였으며, Bergeron & St-Arnne (1992)은 설문을 통하여, 가장 중요한 추정 모델은 경험과 전문성, 유사성의 접근방법이라고 지적하였다. 따라서 선행연구에서 사용되어진 기능점수는 현재의 기술발전과 전문가들의 추정 능력을 반영하지 못하고 있다. 개발노력 추정에 사용되어지는 주요한 모델로서, 알고리즘적 추정 모델은 측정된 출력을 얻기 위해 척도를 가진 입력 또는 개발노력과 연관된 수학적 공식을 사용해 분석된다. 이 공식은 과거 이력자료 분석으로부터 제시된 형식적인 모델(Formal Model)에 사용되며, 모델의 정확도는 특정 개발환경을 반영하여 가중치 조절을 포함한 모델의 조정(Calibration)으로 향상된다.

2.3 인공지능 모델

2.3.1 기능점수 이용 개발노력 추정 신경망 모델

이상운 (2001)은 개발노력 추정을 위하여 기능점수를 이용한 단일 은닉층을 가진 FFN(FeedForward Network) 신경망 모델을 제시하였다. 신경망을 이용한 실험에서, 은닉 뉴런 수를 1개에서 30개 까지 변화시키면서 실험을 수행한 결과 최적의 은닉 뉴런 수(6개)와 가중치가 결정되었다. FFN 신경망을

이용한 경우 주어진 데이터에 비선형으로 근사시켜 Albrecht et al. (1981)의 선형회귀모델 보다 정확한 개발노력을 추정 할 수 있었다.

또한 이상운,박영목,박재홍 (2001)은 RBF망을 이용하여 개발노력 추정을 하였다. RBF망은 FFN과 마찬가지로 은닉층과 출력층으로 구성되어 있지만, FFN과 같이 정적으로 은닉 뉴런 수를 결정하는 방법 대신, 주어진 문제에 적합하도록 은닉 뉴런수를 0개로부터 시작하여 점진적으로 추가시키면서 주어진 문제에 적합한 뉴런 수를 찾는 방법을 사용할 수 있다.

선행연구에서 활용되었던 FFN 신경망의 경우는 모델의 구조를 결정하는 것으로 최적의 은닉 뉴런 수를 찾는 문제가 제기된다. FFN의 1개의 은닉층에 시그모이드 작동함수를 가진 뉴런을 무한히 갖고 있으면 어떠한 함수도 표현할 수 있다는 보편적 근사 이론이 증명되었다

(Bergeron, 1992). 그러나 최적의 은닉 뉴런 수는 주어진 문제의 복잡도에 따라 결정되는 단점이 있다. 이에 반해, RBF 신경망의 최대 은닉 뉴런 수는 최대로 입력 데이터 개수가 되며, FFN에 비해 수렴 속도가 빠르고, 작은 오차와 높은 신뢰성을 가지는 장점이 있다. 그러나 선행연구에서 FFN 모델의 설계를 위하여 사용되어진 데이터가 프로젝트의 최근의 소프트웨어 개발환경, 개발 특성 등을 반영하고 있지 않아 최근의 다양한 개발환경과 언어를 사용한 충분한 프로젝트 규모를 갖고 있는 데이터에 적합한지 여부는 검증되지 못하였다.

2.3.2 사례기반(Case-based Reasoning) 모델

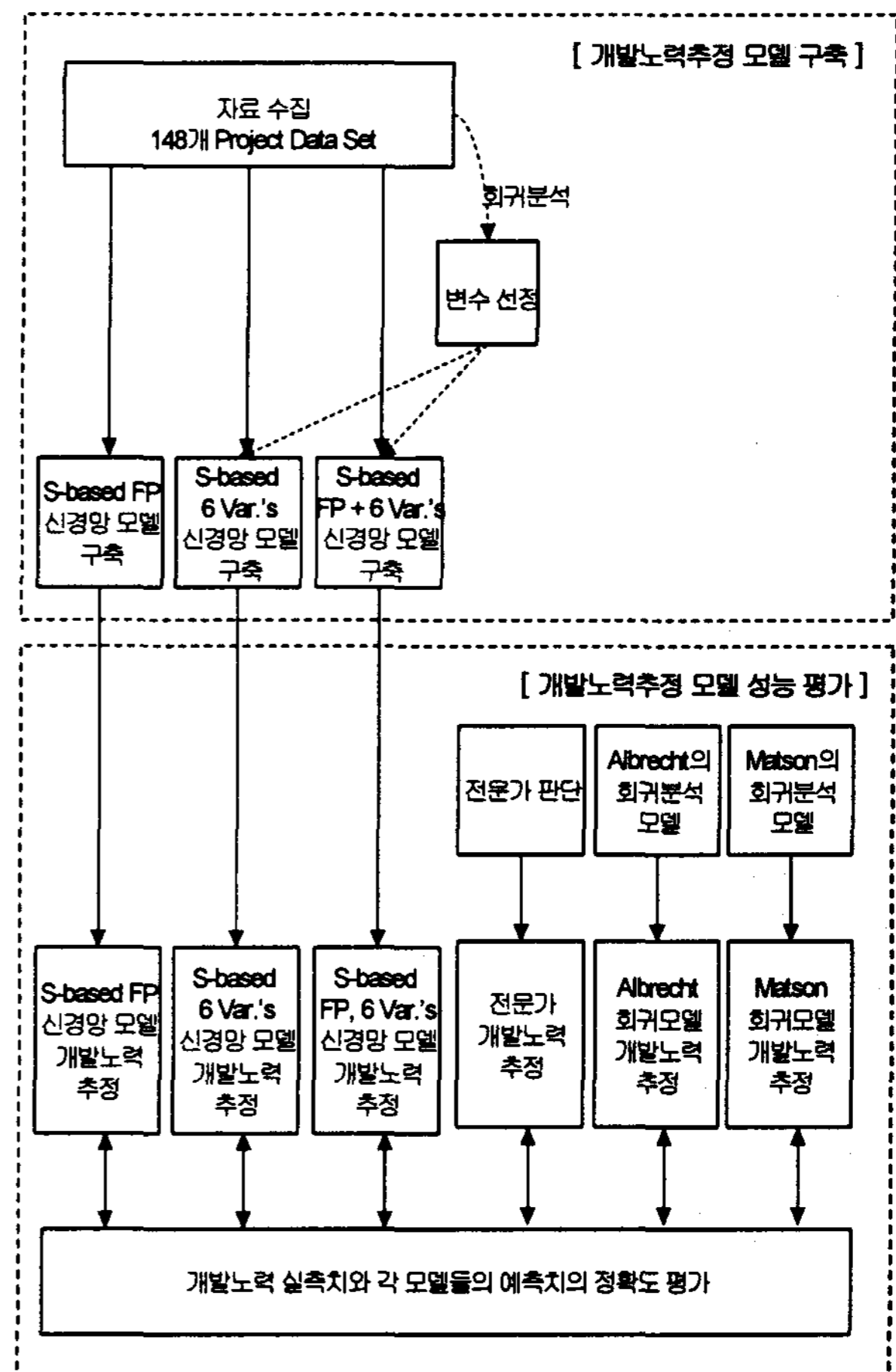
유추에 의한 문제의 해결 방식이 일반적으로 많이 사용됨에도 불구하고, 이러한 영역에 대해서 이론적으로나 경험적으로 연구가 많이 이루어지지 않았다. 이러한 유추에 의한 문제해결 방식에 대해서 정리하여 모델을 세운

것이 사례기반(Case-based Reasoning) 모델이다. 일반적인 연구들에서 유추에 의한 문제해결 방식은 그러한 유형의 방식을 사용하는 전문가가 프로젝트의 추정을 위하여 진행하는 유추의 과정을 묘사하고 있다. 그러나 이러한 이론들을 소프트웨어 개발노력의 방안으로 적용하기에는 너무 일반적인 모델만을 제공한다. 따라서 이러한 유추의 과정을 소프트웨어 개발노력 추정에 연계시킨 모델 중에 하나가 Estor의 사례기반 유추 모델 (Mukhopadhyay, 1992)이다.

3. 연구방법

본 논문에서는 [그림 1]와 같은 절차에 의해서 모델의 구축 및 성능평가를 진행하였다.

[그림 1] 모델 분석 모형



3.1 자료수집

신경망 모델 구축을 위하여 국내 S사가 지식관리 시스템에 의해서 관리하고 있는 프로젝트들의 산출물 중 총 148건을 사용하였으며, 추가 9건은 구축한 모델의 성능을 타 모델들과 비교 평가하는데 사용하였다. 데이터로 활용한 프로젝트들은 최근의 유형을 최대한 반영코자 1999년부터 2003년까지 약 4년 동안의 최근 프로젝트들을 대상으로 선정하였으며, 기술적인 측면에서는 Host, Client/Server, Web 개발, Package 이용 개발 등 다양한 기술이 포함될 수 있도록 하였고, 규모면에서는 3M/M에서 약 3,000M/M까지, 기간측면에서는 단기간 프로젝트로는 약 1개월, 최장기간 프로젝트로는

약 2년 이상을 포함하도록 하였으며, 적용된 분야에서도 다양한 업종에 분포토록 하여, 구축하는 신경망의 왜곡을 막고, 선정된 변수들의 값이 골고루 분포될 수 있도록 노력하였다.

3.2 변수선정

기존의 정량적이며, 시스템 위주의 기능점수(FP)만을 고려한 개발노력 추정에는 한계가 있음을 선행 연구의 분석을 통해서 확인할 수 있었다. 따라서 본 연구에서는 개발노력 추정 신경망 구축을 위하여 기능점수 이외에 다양한 기타 Input 항목의 변수 선정 고려하였으며, 다음과 같은 절차를 거쳤다.

- (1) 프로젝트의 기능점수 산정 : IFPUG 공인 계산방식 활용
- (2) 타 연구모델 Input 항목 참조 : 전용섭(2001)의 신경망 Input Factors
- (3) 정보통신부 “소프트웨어사업대가기준”의 “생산성보정계수” 항목 : 정보통신부 고시(2003)
- (4) S사 지식관리시스템 프로젝트 분류 참조 추가 항목 선정

전용섭(2001)이 제시한 신경망 Input Factors에서 네 가지 범주인 Project, Product, Staff, Technology가 Boehm (1984)과 Bergeron & St-Arn명 (1992)가 조사한 범주와 일치함을 보였으며, 정보통신부에서 제시하고 있는 소프트웨어사업대가기준의 “생산성보정계수” 항목은 크게 프로젝트 형태 분류, 개발언어별 분류, 적용대상 기종별 분류로 구성되어 있다(정보통신부, 2003). 여기서는 앞에서 기술한 기능점수 및 전용섭의 신경망 설계 Input Factor들과 정보통신부의 “소프트웨어사업대가기준”의 “생산성보정계수” 항목을 참고로 하고, S사의 프로젝트 지식관리

시스템을 근거로 하여, 개발노력 추정을 위한 신경망 구축을 위하여 총 39개의 입력변수들을 도출 한 후, 개발노력에 영향력을 행사할 수 있는 변수의 선정을 위하여, 1차로 전문가들의 판단을 추가로 고려하였다. 전문가들의 판단은 설문을 통하여 반영을 하였으며, 설문은 국내 평균 10년 이상의 소프트웨어 개발 및 프로젝트 관리 경험이 있는 총 300여명의 전문가를 대상으로 실시하였으며, 회수된 설문은 64개였다. 회수된 설문을 근간으로 항목들의 가중치에 대한 빈도분석을 실시하였다. <표 1>는 본 연구에서 입력 변수로 선택되어진 변수들을 요약하여 놓았다.

<표 1> 입력 변수

변수	항목	Value	Type
I_1	기능점수 (FP)	No.	QT
I_3	프로젝트 기간	Month	QT
I_7	방법론 사용비율	Percentage	QT
I_{14}	주사업자 특/고급 투입 인력 수	No. of Persons	QT
I_{15}	주사업자 중급 투입 인력 수	No. of Persons	QT
I_{16}	주사업자 초급 투입 인력 수	No. of Persons	QT
I_{17}	주사업자 참여 인력 비율	Percentage	QT

3.3 모델 구축 및 평가

3.3.1 개발노력 추정 신경망 모델 구축

앞에서도 언급하였듯이, 본 연구에서는 148개의 최신 소프트웨어 개발 사례를 기반으로 개발노력 추정 모델을 신경망을 활용하여 구축을 했다. 기능점수와 S-based Factors에서 선정된 6개 항목들을 기반으로 하는 다양한 모델들의 성능을 고려하고자, 3가지 조합의

모델을 구축했다. 구축한 3가지 모델의 유형은 Input 항목들의 조합에 따라서 다음과 같다.

- 모델 1: FP (I_1) 기반 신경망 모델
- 모델 2: S-based 6 Factors ($I_3, I_7, I_{14}, I_{15}, I_{16}, I_{17}$) 기반 신경망 모델
- 모델 3: FP 포함 S-based 7 Factors ($I_1, I_3, I_7, I_{14}, I_{15}, I_{16}, I_{17}$) 기반 신경망 모델

신경망 모델 구축을 위하여, SAS사의 Enterprise Miner v8.2를 활용하였으며, 위의 세 가지 예측 모형에 대해서 148개 데이터 셋을 대상으로 Training과 Validation을 7 대 3의 비율로 이용하였다.

3.3.2 개발노력 추정 타 모델과의 비교 평가

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미 있는 척도로서 모델의 추정 정확도를 평가하는 것을 필요로 하는데 본 논문에서는 모델간의 상대적인 성능비교를 위하여 MRE(Magnitude of Relative Error)를 각 모델에 적용하였다. Kemerer (1993)에 의해서 개발노력 추정 모델의 검증을 위하여 사용되었다. MRE는 프로젝트 수행에 필요한 개발노력의 실제 양과 비교하여 추정치 오차의 백분율이다.

$$MRE = \frac{|MM_{est} - MM_{act}|}{MM_{act}} * 100$$

MM_{est} 는 추정 Man-Months이고, MM_{act} 는 실제 프로젝트에서 적용된 개발노력 Man-Months이다. 또한 오차의 평균값 비교를 통하여 직관적인 성능의 비교를 할 수 있도록 하였다.

본 논문에서 구축되어진 모델의 성능평가를 위하여, 기존의 주요 모델들과의 비교를 하였다. 비교 대상으로 선정된 모델들은 기능점수 기반의 회귀분석 모델인 Albrecht & Gaffney (1983)과 Matson (1994)의 논문에서 강조되었던 전문가들의 판단에 의한 추정 능력이다. 성능평가를 위하여, 추가로 9개 소프트웨어 개발 프로젝트의 사례를 수집하였으며, 전문가들의 개발노력 추정 비교를 위하여, 프로젝트 경험이 10년 이상인, 프로젝트 관리자(Project Manager)들에게 추정에 관한 의뢰를 하였다. 성능평가의 결과는 <표 2>에서 보는 바와 같이 기능점수를 포함한 S-based 7 Factors에 의해 구축된 신경망 모델이 가장 우수한 것으로 나타났다. MRE의 평균값은 59.4%로 가장 낮게 나타났으며, MRE의 표준편차도 30.5%로 가장 낮게 나타났다. 또한 오차의 평균값에서도 가장 적은 값을 나타내었다.

<표 2> 모델 평가 결과값

Results Models	MRE(%)				Average of Error
	Avg	Standard Deviation	Max	Min	
S-based FP Neural Network Model	110.6	80.6	230.0	24.3	66.9
S-based 6 Var's Neural Network Model	63.6	52.3	142.8	7.7	51.0
FP 포함 S-based 7 Var's Neural Network Model	59.4	30.5	101.7	8.0	31.7
Expert Judgement Estimation Model	76.6	47.8	142.4	25.9	49.2
Albrecht's Regression Model	150.4	110.1	287.5	5.3	117.3
Matson's Regression Model	417.3	543.6	1718.0	19.1	445.0

또한 주목할만한 내용은 전문가들의 판단 및 S-based 6 Factors에 의한 예측치가 기타 모델들에 비교하여 우수한 것으로 나타난 것으로, 특히 전문가들의 경험에 의한 판단의

정확성은 이전의 연구 논문들에서 강조하여 왔던 것을 다시 한번 입증하는 결과를 나타냈다. 또한 Albrecht (1981)와 Matson et al., (1994)의 개발노력 추정 회귀분석모델은 현재의 소프트웨어 개발환경에서 사용하기에는 많이 부적합한 것으로 나타났다. 즉 각각의 모델들이 만들어졌을 때의 소프트웨어 개발환경과 비교하여, 현재의 개발환경은 추가적인 많은 기술적, 형태적 복합변수들의 작용을 고려해야만 한다.

4. 결 론

본 연구에서는 소프트웨어 개발노력 추정을 위한 모델로서 신경망 시스템의 예측력을 평가하기 위하여, 전통적인 예측 방법인 회귀분석과 전문가의 예측력과의 비교하였다. 모델 성능의 비교 평가에서 FP를 포함한 S-based 7 Factors로 구축된 신경망 모델이 가장 우수한 것으로 나타났으며, 전문가의 예측력도 매우 뛰어난 것으로 판단되었다. 그러나 기존의 개발노력추정 회귀분석 모델은 현재의 소프트웨어 개발노력 추정에는 많은 한계점을 보이는 것으로 나타났다.

참고 문헌

이상운 (2001), "신경망을 이용한 소프트웨어 개발노력 추정", 한국정보처리학회 논문지 (D), 제8-D권 제3호, pp.241-246

이상운 (2002), "주성분분석을 이용한 소프트웨어 개발노력 추정능력 향상", 한국정보처리학회논문지(D), 제9-D권 제1호, pp.75-80

이상운, 강정호, 박중양 (2002), "기능점수를 이용한 소프트웨어 개발노력 추정",

한국정보처리학회 논문지(D), 제9-D권 제4호, pp.603-612

이상운, 박영목, 박재홍 (2001), "RBF 망 이용 소프트웨어 개발노력 추정 성능향상", 한국정보처리학회 논문지(D), 제8-D권 제5호, pp.581-586

전용섭 (2001), "축약형 신경망과 휴리스틱 검색에 의한 소프트웨어 공수 예측모델", 한국정보처리학회 논문지, 제8-D권, 제2호

정보통신부 (2003), "소프트웨어사업대가의 기준", 정보통신부 고시 2003-14호

Albrecht, A. J. (1979), "Measuring Applications Development Productivity", Processing of IBM Application Dev., Joint SHARE/GUIDE Symposium, Monterey, CA, pp.83-92

Albrecht, A. J. (1981), "Measuring Application Development Productivity", in Programming Productivity : Issues for the Eighties, C. Jones, ed. Washington, DC : IEEE Computer Society Press

Albrecht, A. J. and Gaffney, J. E. (1983), "Software Function, Source Line of Code and Development Effort Prediction : A Software Science Validation", IEEE Trans. on Software Eng., Vol.SE-9, No.6, pp.639-648

Bergeron, F. and St-Arnaud, J. Y. (1992), "Estimation of Information Systems Development Efforts", Information and Management, Vol.22, pp.239-254

Boehm, B. W. (1981), "Software Engineering

Economics", Prentice Hall

Boehm, B. W. (1984), "Software Engineering Economics", IEEE Trans. On Software Eng., Vol.10, No.1, pp.7-19

Bradley, M. (1999), "Function Point Counting Practices Manual, Release 4.1", International Function Point Users Group (IFPUG)

Kemerer, C. F. (1993), "Reliability of Functional Point Measurement-A Field Experiment", Communications of ACM
Matson, J. E., Barrett, B. E. and Mellichamp, J. M. (1994), "Software Development Cost Estimation Using Function Points", IEEE Trans. on Software Eng., Vol.20, No.4, pp.275-287

Mukhopadhyay, V.P. (1992), "Examining the Feasibility of a Case-Based Reasoning Model for Software Effort Estimation", MIS Quarterly

Venkatachalam, A. R. (1993), "Software Cost Estimation Using Artificial Neural Networks", in Proceedings of 1993 International Joint Conference on Neural Networks, pp.987-990

Vicinanza, S. S., Mukhopadhyay, T., and Prietula, M. J. (1991), "Software-Effort Estimation : An Explolatory Study of Expert Performance", ISR, Vol.2, No.4, pp.243-262