

# 그룹 고장 데이터의 소프트웨어 신뢰성 예측에 관한 신경망 모델

이 상 운<sup>†</sup> · 박 영 목<sup>††</sup> · 박 수 진<sup>†††</sup> · 박 재 흥<sup>††††</sup>

## 요 약

많은 소프트웨어 프로젝트는 시험이나 운영단계에서 고장 시간이나 고장 수 데이터 보다는 그룹 고장 데이터 (여러 고장 간격에서 또는 가변적인 시간 간격에서의 고장 들)가 수집된다. 본 논문은 그룹 고장 데이터에 대해 가변적인 미래의 시간에서 누적 고장 수를 예측할 수 있는 신경망 모델을 제시한다. 신경망의 입-출력으로 무엇을 선택하고 어떤 순서로 훈련을 수행하느냐에 따라 신경망의 예측력에 영향을 미친다. 따라서, 신경망의 입-출력에 대한 11개의 훈련제도가 고려되었으며, 모델의 성능을 평가하기 위해 다음 단계 평균 상대 예측 오차 (AE)와 정규화된 AE (NAE) 측도에 의해 최적의 훈련제도가 선택되고, 다른 잘 알려진 신경망 모델과 통계적 소프트웨어 신뢰성 성장 모델과 비교되었다. 실험 결과, 가변적인 미래의 시간 간격에서 누적 고장 수를 예측하기 위해서는 신경망 모델에 가변 시간간격 정보가 필요함을 보였다.

## Neural Network Modeling for Software Reliability Prediction of Grouped Failure Data

Sang-Un Lee<sup>†</sup> · Yeong-Mok Park<sup>††</sup> · Park-Soo Jin<sup>†††</sup> · Jae-Heung Park<sup>††††</sup>

### ABSTRACT

Many software projects collect grouped failure data (failures in some failure interval or in variable time interval) rather than individual failure times or failure count data during the testing or operational phase. This paper presents the neural network (NN) modeling that is able to predict cumulative failures in the variable future time for grouped failure data. A NN's predictive ability can be affected by what it learns and in its learning sequence. Eleven training regimes that represents the input-output of NN are considered. The best training regimes are selected based on the next-step average relative prediction error (AE) and normalized AE (NAE). The suggested NN models are compared with other well-known NN models and statistical software reliability growth models (SRGMs) in order to evaluate performance. Experimental results show that the NN model with variable time interval information is necessary in order to predict cumulative failures in the variable future time interval.

### 1. 서 론

최근들어 컴퓨터 시스템의 급격한 활용에 따라, 소프트웨어 시스템이 대부분의 복잡하고 중요한 컴퓨터

시스템의 가장 중요한 부분을 차지하고 있으며, 소프트웨어 시스템의 고장은 인간의 생활, 환경 또는 경제적 손실에 치명적인 영향을 유발시킬 수 있다. 따라서, 소프트웨어 시스템 개발은 사용자가 의도하는 목적에 적합한 충분한 품질 수준에 일치해야 하며, 요구되는 품질 수준에 일치하는 지 여부를 평가할 수 있는 측도가 필요하다. 일반적으로 소프트웨어의 품질을 정량적

† 정 회 원 : 국방품질관리소 항공전자장비 품질보증 담당  
†† 준 회 원 : 경상대학교 대학원 컴퓨터학과  
††† 준 회 원 : 경상대학교 컴퓨터학과 박사수료  
†††† 정 회 원 : 경상대학교 컴퓨터학과 교수  
논문접수 : 2000년 7월 8일, 심사완료 : 2000년 10월 9일

으로 표현하는 속성중의 하나가 소프트웨어 신뢰성이다[12, 17]. 소프트웨어 신뢰성은 주어진 환경하에서 주어진 시간동안 소프트웨어가 고장없이 작동할 확률로 정의된다[4, 12].

소프트웨어 신뢰성을 평가하기 위해 고정된 시간 간격 (Fixed time interval)에서의 고장 발생 현상이 주어진 분포를 따른다는 가정하에 유도된 수학적 모델에 관찰된 고장 데이터를 이용해 추정된 모수를 적용한 통계적 신뢰성 성장 모델 (Software reliability growth models : SRGMs)이 일반적으로 사용되고 있다. 최근 들어 모델의 모수 추정과 모델 자체 학습으로 미래의 결과를 예측하는데 신경망 (Neural networks : NNs)이 적용되고 있다[1, 7]. NNs는 사전에 고장 발생 현상이 주어진 분포를 따른다는 가정이 필요 없이 단지 입력된 고장 데이터에 적합하도록 학습을 통해 자율적으로 망의 내부 모델을 개발하고 미래의 고장 발생 과정에 대한 예측을 할 수 있는 장점을 갖고 있다. 즉, 고장 발생 이력의 복잡성에 적합하도록 모델의 복잡성을 자율적으로 조절하기 때문에 널리 사용되고 있는 통계적 모델보다 정확한 결과를 얻을 수 있다[5, 10, 17]. 소프트웨어 신뢰성 예측 관련 대표적인 신경망 연구로는 Karunanithi et al. [8, 9]과 Park et al. [16]이 있다.

소프트웨어 시험이나 운영중에 수집되는 고장 데이터는 소프트웨어 신뢰성을 추정하거나 예측하는데 사용된다. 일반적으로 수집되는 고장 데이터 형태는 고장 수 (일정한 시간 간격동안 발생한 고장 수 기록)나 고장 시간 (각 고장이 발생한 시점을 기록하는 고장 시간 데이터)이며, 고장시간 데이터 형태가 아닌 고장 시간 데이터를 그룹화한 형태인 그룹 데이터 (Grouped data) 형태만 활용할 수 있는 경우도 있다. 이 데이터 형태는 고장 수 데이터와 동일하나 다른 점은 가변적인 시험 간격 (Variable time interval of Testing) 형태를 취한다.

고정된 시간 간격에서 수집된 고장 수를 이용할 경우, 현재 시점의 신뢰성을 추정하거나 고정된 미래 시점의 신뢰성을 예측하는데는 문제가 되지 않는다. 그러나 그룹 데이터의 경우, 가변적인 시간 간격을 가지므로 고정된 시간 간격에 적합한 모델로 가변적인 미래 시점에서의 신뢰성을 예측하는데는 문제가 발생할 수 있다. 기존의 통계적 SRGMs 들은 가변 시간 간격을 가진 그룹 데이터에 대한 많은 연구가 이루어지지 못하였으며, 특히 신경망 모델에 대한 연구는 없는 실정이다.

따라서, 본 논문에서는 가변적인 시간 간격을 가진 그룹 데이터에 적합한 신경망 모델을 개발하는 연구를 수행하고자 한다. 2장에서는 관련 연구 및 문제점을, 3장에서는 가변 시간간격 데이터에 적합한 신경망을 제안하며, 4장에서는 제안된 모델 13가지에 대해 전향 신경망 (FeedForward Network : FFN)을 이용해 모델의 예측 결과를 실험한 후 최적의 모델을 선정하고 널리 알려진 신경망과 통계적 SRGM들과 상호 비교함으로써 제안된 모델의 적합성을 살펴보고자 한다.

## 2. 관련 연구 및 문제점

데이터 쌍들  $(t_i, C_i), i=1, 2, \dots, n$  이 소프트웨어 시험 결과 관찰되었다고 가정하자. 여기서  $C_i$  는  $i$  번째 시험 시간인  $t_i$  까지 발견된 누적 고장 수를 의미하며, 등 간격을 유지하는 단위 시험시간  $\Delta t_i = t_i - t_{i-1}$  로,  $i$  번째 단위 시험시간에서 발견된 고장 수  $\Delta C_i = m_i = C_i - C_{i-1}$  로 정의된다. 이들 데이터 쌍들을 고장 수 데이터라 칭한다.

소프트웨어 신뢰성 분야에서, 미래에 발생되는 고장 수를 예측하는데 가장 널리 알려진 망으로는 FFNs와 PRNs (Partial Recurrent Networks)인 Jordan과 Elman 망이 있다. Karunanithi et al. [8, 9]는 신경망을 활용하여 소프트웨어 신뢰성 예측을 모델링하였으며, 소프트웨어 신뢰성 예측에 캐스케이드-상관 학습 알고리즘 [2] (Cascade-correlation learning algorithm : CC)을 이용하여 FFNs과 PRNs이 적용될 수 있음을 보였다. 신경망의 예측력을 평가하기 위해 2가지 훈련 제도 (Training regimes)를 고려하였다.

- 일반 훈련제도 (Generalization training regime)
  - : 입력 (Input)  $t_i$  는 목표 (Target)  $C_i$  에 관련
- 예측 훈련제도 (Prediction training regime)
  - : 입력  $t_{i-1}$  는 목표  $C_i$  에 관련

또한, 고장 수 데이터에 대해 잘 알려진 통계적 SRGMs 들과 비교하였다. 실험에 사용된 14개의 데이터를 살펴보면, 10개 (Data1, Data3-10와 Data14)는 고장 수 데이터이며, 나머지 4개 (Data2와 Data11-13)는 고장 시간 데이터이다. 고장 시간 데이터에 대해 고장 수 데이터 신경망 모델을 잘못 적용한 점을 보완하기 위해 Park et al. [16]은 고장 시간 데이터에 대해 유사한 연

구를 수행하였다. 10개의 고장 수 데이터 중 Data4와 Data5는 데이터 수집 시간 간격이 일정하지 않은 그룹 데이터의 경우이고 나머지는 데이터 수집 시간 간격이 일정한 경우이다.

신경망을 함수 근사, 회귀분석 또는 시계열 분야에 이용하는 경우, 주어지는 시간 간격은 일정한 데이터를 활용하는 것이 일반적이며, 또한 소프트웨어를 시험하면서 수집되는 고장 수 데이터는 일반적으로 시간 간격이 일정한 형태를 취한다. 그러나 <표 1>과 같이

<표 1> 그룹 데이터 특성 및 고장 데이터

데이터	참고 문헌	LOC	고장 수	Data Size	Application (Time interval)
Data 4	[15]	1,317,000	328	17	PL1 database application software (CPU execution time in Hours)
Data 5	[15]	35,000	279	10	Hardware control program (Time of Observation in Months)
System T1	[14]	21,700	136	21	Real time command control (CPU execution time in Hours)
System T38	[14]	Not known	32	11	Not known (CPU execution time in Hours)

Data4				Data5			
$t_i$	$\Delta t_i$	$\Delta C_i$	$C_i$	$t_i$	$\Delta t_i$	$\Delta C_i$	$C_i$
2.45	2.45	15	15	1.0	1.0	10	10
4.9	2.45	29	44	1.5	0.5	64	74
6.86	1.96	22	66	2.0	0.5	18	92
7.84	0.98	37	103	3.0	1.0	43	135
9.52	1.68	2	105	4.5	1.5	44	179
12.89	3.37	5	110	6.0	1.5	13	192
17.1	4.21	36	146	8.0	2.0	34	226
20.47	3.37	29	175	11.0	3.0	28	254
21.43	0.96	4	179	12.0	1.0	15	269
23.35	1.92	27	206	13.0	1.0	10	279
26.23	2.88	17	223				
27.67	1.44	32	255				
30.93	3.26	21	276				
34.77	3.84	22	298				
38.61	3.84	6	304				
40.91	2.3	7	311				
42.67	1.76	17	328				

System T1				System T38			
$t_i$	$\Delta t_i$	$\Delta C_i$	$C_i$	$t_i$	$\Delta t_i$	$\Delta C_i$	$C_i$
0.0092	0.0092	2	2	5	5	1	1
0.0192	0.01	0	2	15	10	0	1
0.0222	0.003	0	2	25	10	16	17
0.0452	0.023	1	3	35	10	1	18
0.0862	0.031	1	4	45	10	1	19
0.0902	0.004	2	6	50	5	0	19
0.1152	0.025	1	7	65	15	1	20
0.4172	0.302	9	16	75	10	3	23
1.3902	0.973	13	29	95	20	2	25
1.4102	0.02	2	31	120	25	7	32
1.8602	0.45	11	42	125	5	0	32
2.1102	0.25	2	44				
3.0502	0.94	11	55				
4.3902	1.34	14	69				
7.7102	3.32	18	87				
11.2702	2.56	12	99				
13.9302	2.66	12	111				
17.7002	3.77	15	126				
21.1002	3.4	6	132				
23.5002	2.4	3	135				
25.3002	1.8	1	136				

데이터 수집 시간 간격이 가변적인 그룹 데이터의 경우도 발생한다.

가변 시간간격을 가진 그룹 데이터를 기존의 고정된 시간간격에 적합한 신경망 모델을 이용하여 미래의 가변 시간 간격 시점에서의 고장 수를 예측할 경우 문제가 발생한다. 예를 들면, Karunanithi et al. [8,9]이 제안한 예측 훈련제도의 경우 입력은  $t_{i-1}$ 인데 목표는 고정된 시간간격인  $t_i$  시점에서의 누적 고장 수인  $C_i$ 이다. 따라서, 그룹 데이터의 가변적 시간간격인  $t_i$  시점까지의 누적 발견 고장 수를 예측하려면 가변 시간 간격  $\Delta t_i = t_i - t_{i-1}$  정보를 알아야 한다. 그러나 기존 모델링에서는  $\Delta t_i$  정보가 전혀 들어있지 않음으로 문제가 발생하며, 이 모델은  $\Delta t_i$ 가 일정할 때만 적용 가능하다. 이때는  $t_i$ 들을  $i$ 로 대체하여도 동일한 결과를 얻을 수 있다. 즉,  $\Delta t_i$ 가 가변 시간 간격일 경우, 기존 신경망 모델로는  $\Delta t_i$  시간 후에 발견되는 고장 수는 예측이 불가능하다. 이 문제를 해결하기 위해서는 현재 시점인  $t_{i-1}$ 에서 얼마 동안의 시간에 발생하는 고장 수를 예측하는가 하는 가변 시간간격 ( $\Delta t_i$ )이 주어져야 한다. 이러한 그룹 데이터에 적합한 신경망 연구는 수행되지 않았다.

따라서, 3장에서는 가변적인 시간 간격 데이터에 적합한 신경망 모델을 얻기 위해 수집된 데이터를 이용하여 Karunanithi et al. [8,9]이 제안한 훈련 체도를 변경시켜 적절한 신경망 모델을 제시하고자 한다.

### 3. 그룹 데이터에 적합한 신경망 모델

소프트웨어 시험 또는 운영 중에  $l$  시점까지 획득된 가변 시간간격 고장 수 데이터 ( $t_i, C_i$ ),  $i=1,2,\dots,l$ 를 활용하여 미래의 가변 시간 간격 시점  $t_{l+r}$ 에서의 고장 수  $C_{l+r}$ 를 신경망을 이용하여 예측하는 경우, 최적의 신경망을 어떻게 선택할 것인가가 문제로 제기된다. 본 장에서는 이러한 문제점을 해결하는 방법을 제시한다.

최근들어, 다층 (Multi-layer) FFNs이 패턴 분류, 함수 근사와 회귀분석 분야에 일반적으로 널리 사용되고 있다 [3, 11]. 따라서, 본 연구에서도 FFNs을 적용한다. 주어진 데이터에 적합한 모델은 모델의 복잡성 뿐만 아니라 신경망을 훈련시키는데 있어서 입력-출력 데이

터를 어떻게 선택하고 어떤 순서로 배열하는가에 따라 모델의 성능에 영향을 미친다. 따라서, 가변적인 시간 간격을 가진 그룹 고장 데이터에 적합한 신경망 구조를 얻기 위해 입력을 어떤 것을 선택할 것인가에 대한 연구를 수행하고자 한다. 따라서, 그룹 고장 데이터에 적합한 입력을 얻기 위해 획득된 가변 시간간격 고장 수 데이터  $(t_i, C_i)$ ,  $i=1, 2, \dots, l$ 에서 필요한 정보를 모두 추출하여 보자.  $t_i$ 로부터  $t_i, t_{i-1}, \Delta t_i = t_i - t_{i-1}$ 를,  $C_i$ 로부터  $C_i, C_{i-1}, \Delta C_i = m_i = C_i - C_{i-1}$  정보를 얻을 수 있을 것이다. 미래의 가변 시간간격 시점  $t_{i+r}$ 에서의 누적 고장 수  $C_{i+r}$ 를 예측하기 위한 훈련제도로서, 신경망의 출력 목표 값으로  $C_i$ 로 설정한다. 나머지 정보를 조합하여 신경망의 입력으로 설정함으로써 13개의 훈련제도를 <표 2>와 같이 고려할 수 있다.

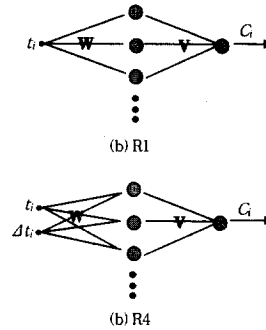
<표 2> 신경망 훈련 제도

훈련제도	입력 값					목표 값
	$t_{i-1}$	$t_i$	$\Delta t_i$	$C_{i-1}$	$\Delta C_i$	$C_i$
Karunanithi et al.[8, 9]	R1		○			○
	R2	○				○
제안된 훈련제도	R3	○	○			○
	R4		○	○		○
	R5	○		○		○
	R6		○		○	○
	R7	○			○	○
	R8		○			○
	R9	○				○
	R10		○	○	○	○
	R11	○		○	○	○
	R12		○	○		○
	R13	○		○	○	○

<표 2>에서 R1과 R2는 각각 Karunanithi et al. [8, 9]가 제시한 일반 훈련제도와 예측 훈련제도에 해당되며, R3-R13은 새로 제안된 훈련제도이다. 새로 제안된 훈련제도는 다음과 같이 표현된다.

- R3 : 입력  $t_i$  와  $t_{i-1}$ 는 목표  $C_i$ 에 관련
- R4 : 입력  $t_i$  와  $\Delta t_i$ 는 목표  $C_i$ 에 관련
- ⋮
- R13 : 입력  $t_{i-1}$ ,  $\Delta t_i$  와  $\Delta C_i$ 는 목표  $C_i$ 에 관련

즉, Karunanithi et al.이 제안한 훈련제도 1 (R1)과 본 논문에서 제안하고 있는 훈련제도 4 (R4)에 대한 신경망의 구조는 (그림 1)과 같다.



(그림 1) 훈련제도 1 (R1)과 4 (R4)의 신경망 구조

훈련제도에 따라 현재의  $l$  시점까지 획득된 데이터로 신경망의 입력과 목표 값으로 설정하고 신경망을 훈련시킨다. 훈련이 완료된 신경망에 대해 미래의 가변 시점인 다음 단계 (Next-step)에 해당하는 입력 값을 신경망에 가하면 훈련된 신경망의 학습 능력에 의해 다음단계의 누적 고장수인  $\hat{C}_{i+1}$ 이 출력되며, 이 값을 다음 단계의 누적 고장 수 목표값인  $C_{i+1}$ 와 비교하여 훈련된 신경망의 예측 성능을 검증할 수 있다.

#### 4. 예측력 실험 및 결과 분석

다른 여러 가지 모델들을 비교하는데 있어, 어떤 의미있는 측도로서 모델의 예측 정확도를 정량화하는 것이 필요하다. 소프트웨어 신뢰성 분야에서 일반적으로 사용되는 방법은 적합도 검증(Goodness-of-fit), 다음 단계 예측력(Next-step predictability)과 가변 미래시점 예측력(Variable-term predictability)이다 [9, 13]. 이들 접근법을 사용하여, Malaiya et al. [13]는 2가지 예측 측도로 평균 상대 예측 오차 (Average relative prediction error : AE)와 평균 편향(Average bias : AB)을 제안하였다. AE는 모델이 얼마나 잘 예측하는가의 측도이며, AB는 모델의 일반적 편향이다. 본 논문에서는 표 1의 4개 데이터에 대한 다음 단계 예측력을 평가하기 위해 AE 측도를 사용한다. 신경망의 구성 및 훈련은 MATLAB 5.2 Neural Network Toolbox Ver. 3.0의 FFNs를 구성하여 실험하였다.

##### 4.1 예측력 실험

본 연구에서는 은닉층이 1개인 FFN에 대해 역전파 알고리즘 (Backpropagation algorithm : BP)과 최적 시점에서 신경망의 훈련을 종료하는 기법을 사용하였다.

왜냐하면, Hornik et al. [6]은 은닉층이 1개인 FFN에 대해 시그모이드 작동함수를 가진 뉴런을 은닉층에 무한히 추가시키면 보편화된 함수근사기로서의 역할을 수행할 수 있음을 제시하였기 때문에 1개의 은닉층을 가진 FFN을 사용해도 되기 때문이다.

신경망의 은닉층 작동함수로는 시그모이드를, 출력층 뉴런의 작동함수로는 선형을 사용한다. 출력층 뉴런의 작동함수로 선형을 사용하는 이유는 다음과 같다. 일반적인 신경망에 대해 관찰된 데이터를 정규화하지 않고 신경망에 직접 적용할 수 있는 방법은 출력층 뉴런의 작동함수로 선형을 사용하는 경우이다. 이에 대해 Cybenko [1]는 시그모이드 작동함수를 가진 1개의 은닉층으로 주어진 모든 문제를 표현할 수 있다는 보편적 근사 이론 (Universal Approximation Theorem)을 증명해 보였으므로 출력층 뉴런의 작동함수로 선형을 사용할 수 있고, 또한 이상운[18]은 출력층 뉴런의 작동함수로 선형을 사용하여 정규화 시키지 않은 데이터를 이용하는 경우가 로지스틱 함수를 사용하여 정규화 시킨 데이터 보다 신뢰성 예측에 있어서 보다 좋은 결과를 얻었다. 그러나 출력층 뉴런에 로지스틱 함수를 사용하면 출력되는 값이 [0, 1] 범위로 한정되므로 수집된 데이터를 [0, 1] 범위로 정규화하여 신경망에 적용하고 출력된 결과를 다시 역정규화하는 불편이 따른다. Karunanithi et al. [8, 9]는 출력층 뉴런의 활성화 함수로 로지스틱 함수를 사용하였다. 이 경우 입력-출력 변수 값의 변환 (정규화)이 필요함을 알 수 있다. 따라서, 본 논문의 접근방법은 어떠한 데이터 변환 과정도 필요없이 수집된 본래의 데이터를 바로 사용할 수 있는 장점이 있다.

신경망의 적절한 구조를 선택하기 위해 시행착오법 (Trial-and-Error)을 사용하고자 한다. 즉, 은닉 뉴런 수는 1개에서 30개까지 1개 단위로 변화시켜 가면서 FFN을 구성하고 최적시점 종료 기법 (Early Stopping 또는 Optimal Stopping)을 사용하여 오차가 최소가 되는 시점에서 신경망의 훈련을 멈추고 다음단계 예측 오차인 AE를 계산한다. 신경망의 다음단계 (Next-step) AE는  $e_{i,1} = ((\hat{C}_{i+1} - C_{i+1}) / C_{i+1}) \times 100$ 로 정의된다. 이 30개의 예측 오차 값 중 가장 최소의 예측 오차를 가진 뉴런이 해당 시점에서의 신경망의 최적의 뉴런수로 결정된다.

13개의 훈련제도에 대한 예측력을 평가하기 위해 다

음단계 AE의 절대값의 평균인  $\bar{e}_{i,r} = \sum_{j=1}^r |e_{i,j}| / r, r=1$  을 사용한다 [9]. 즉,  $e_{i,1} = \bar{e}_{i,1}$ . 그러나, 다른  $r$ 의 값 (즉, 보다 먼 미래시점)이 고려될 수도 있다.

주어진 데이터로부터 신경망의 적절한 구조와 가중치 값을 정확히 알 수 없기 때문에 대부분의 신경망 훈련방법은 신경망 가중치들의 초기 값을 랜덤하게 설정함으로써 인해 신경망을 훈련시킬 때마다 예측력에 차이가 발생한다. 따라서 Karunanithi et al. [9]의 실험방법과 동일하게 50회의 훈련을 통해 평균값을 취한 것으로 예측 오차를 구하고자 한다. 실험 결과 얻어진 다음단계의 전체 평균 예측 오차  $\bar{e}_{i,1}$ 는 <표 3>에 표기하였다.

<표 3> 다음 단계의 전체 평균 오차  $\bar{e}_{i,1}$

훈련제도	AE				
	Data 4	Data 5	System T1	System T38	
Karunanithi et al. [9]	R1	8.6566	10.5199	6.1444	19.6587
	R2	7.2819	4.6284	9.2793	21.3171
제안 훈련제도	R3	6.7723	12.4995	4.0958	22.4497
	R4	5.9137	7.7722	5.0835	21.7219
	R5	5.4105	11.9716	5.4402	22.3162
	R6	8.0187	13.4587	8.9349	23.0162
	R7	12.5464	14.3547	9.5820	21.2572
	R8	11.2911	16.8212	7.0710	21.2088
	R9	8.0208	12.8525	7.6602	20.4389
	R10	10.8478	18.7086	12.1971	23.7348
	R11	9.9429	13.8741	10.4018	13.6549
	R12	7.7233	15.2045	6.2945	20.1998
	R13	9.9101	17.1473	5.4160	23.7650

#### 4.2 예측력 결과 분석

AE는 단일 데이터에 대해 모델들의 예측 정확도를 비교하는데 사용될 수 있으며, 다른 여러 가지 다양한 데이터들에 대해 모델을 비교하기 위해서는 다른 측도가 필요하다. Karunanithi et al. [9]는 다양한 데이터들에 적합한 모델을 비교하는 측도로서 정규화된 AE (Normalized AE : NAE) 를 제시하였다. 데이터  $s$ 에 대해 최대의 AE를 갖는 모델 ( $m$ )을  $m_{max}$ 로, 이 모델의 AE를  $AE_s^{m_{max}}$ 라 하면 정규화된 상대오차  $NAE_s^m = AE_s^m / AE_s^{m_{max}}, (0.0 < NAE_s^m < 1.0)$ 가 되며, NAE 를 구한 후, 각 모델에 대한 순위  $R_m = \sum_{s=1}^{\# \text{ of data set}} NAE_s^m$  을 구하여 모델의 예측력을 평가할 수 있다. <표 3>의 결과에

대한 NAE 및 순위에 따른 모델의 예측력 결과가 <표 4>에 제시되어 있다.

<표 4> 모델의 예측력

훈련 제도	NAE				$R_m$	순위
	Data 4	Data 5	System T1	System T38		
R1	0.6900	0.5623	0.5038	0.8272	2.5832	5
R2	0.5804	0.2474	0.7608	0.8970	2.4855	3
R3	0.5398	0.6681	0.3358	0.9446	2.4883	4
R4	0.4713	0.4154	0.4168	0.9140	2.2176	1
R5	0.4312	0.6399	0.4460	0.9390	2.4562	2
R6	0.6391	0.7194	0.7325	0.9685	3.0595	10
R7	1.0000	0.7673	0.7856	0.8944	3.4473	13
R8	0.8999	0.8991	0.5797	0.8924	3.2712	12
R9	0.6393	0.6870	0.6280	0.8600	2.8143	7
R10	0.8646	1.0000	1.0000	0.9987	3.8633	8
R11	0.7925	0.7416	0.8528	0.5746	2.9615	9
R12	0.6156	0.8127	0.5161	0.8500	2.7943	6
R13	0.7899	0.9165	0.4440	1.0000	3.1505	11

실험 결과 다음과 같은 결론을 얻을 수 있었다.

- (1) 가변 시간 데이터에 대해 미래의 얼마 동안의 기간에 발생하는 고장 수를 예측하는 가 하는 기간 ( $\Delta t_i$ )이 주어진 신경망 모델(R4와 R5)의 예측력이  $\Delta t_i$ 가 주어지지 않은 Karunanithi et al. [9]가 제안한 훈련제도(R1과 R2) 보다 좋은 결과를 나타내었다.
- (2) 제안된 훈련제도 11개(R3-R13) 중 미래의 예측 가변 시간 간격인  $\Delta t_i$  정보가 없고 다른 정보들로만 결합되어 입력되는 경우(R3, R6-R9), 모델의 예측력이 나쁜 결과를 나타내었다.
- (3) 또한,  $\Delta t_i$  정보가 들어 있지만 여러 가지 다른 정보가 결합되어 입력되는 경우(R10-R13), 모델의 예측력이 좋은 결과를 나타내지 못하였다.

즉, 가변 시간간격 데이터에 대한 미래의 가변 시간 간격 시점에서의 누적 고장 수를 예측하기 위해서는 미래의 가변 시간 간격인  $\Delta t_i$  정보가 들어가면서  $t_i$  와  $t_{i-1}$  과 같이 필수적인 정보만 입력되면 적합한 신경망 모델을 얻을 수 있음을 알 수 있다. 이는 복잡한 구조를 가진 모델보다는 유사한 성능을 발휘하면서도 보다 단순한 모델을 선호한다는 Occam's Razor Principle에도 일치하는 결과임을 알 수 있다.

### 4.3 제안된 모델 평가

제안된 모델이 일반적으로 많이 사용되는 통계적 SRGMs 들과 신경망 모델보다 좋은 결과를 나타내는 지를 살펴보고자 한다. 본 비교에는 Data4와 Data5에 대한 Karunanithi et al. [9]의 신경망과 통계적 SRGMs 모델의 예측 결과와 제안 모델의 좋은 예측 결과를 나타내는 훈련제도 (R4-R5)에 대한 NAE를 사용하였으며, <표 5>에 기술되어 있다.

<표 5> 다양한 모델의 예측력 순위

모 델	훈련제도	Data 4		Data 5		$R_m$	순위	
		AE	NAE	AE	NAE			
신 경 망	CC 알고리즘 [9]	FFN-R1	5.28	0.5617	10.00	0.4274	0.9891	5
		FFN-R2	4.64	0.4936	6.95	0.2970	0.7906	1
		Jordan-R1	8.84	0.9404	5.09	0.2175	1.1579	10
		Jordan-R2	6.11	0.6500	8.67	0.3705	1.0205	6
통 계 적 SRGMs [9]	제안 모델 (BP 알고리즘)	Logarithmic	5.93	0.6309	6.42	0.2744	0.9052	3
		Inverse Polynomial	7.95	0.8457	9.71	0.4150	1.2607	8
		Exponential	6.01	0.6394	6.15	0.2628	0.9022	2
		Power	9.40	1.0000	23.36	0.9983	1.9983	11
		Delayed S-shape	6.25	0.6649	10.90	0.4658	1.1307	9

<표 5>에서 신경망 모델과 통계적 SRGMs들을 비교한 결과 FFN을 사용한 제안된 모델의 예측력이 중간 정도의 성능을 나타내었다. 그러나 신경망을 훈련시키는 과정에서 모델을 자동적으로 결정하는 알고리즘인 CC 알고리즘을  $\Delta t_i$  정보를 포함한 훈련제도 R4와 R5 입력자료를 사용하면 보다 좋은 모델을 찾을 수도 있을 것이다.

### 5. 결론 및 향후 과제

본 논문에서는 가변 시간간격을 가진 그룹 고장 데이터에 대한 소프트웨어 신뢰성 예측을 하기 위한 신경망 모델을 연구하였다. 수집된 가변 시간간격 고장 데이터에 대해 필요한 모든 정보를 발췌하여 신경망의 입력으로 하는 훈련제도 11개를 제안하고 그 중에서 가장 좋은 훈련제도를 갖는 모델을 선정하였다. 본 연구에는 4개의 데이터가 사용되었으며, FFN 신경망에 한정하여 모델의 예측력을 평가하기 위해 다음 단계

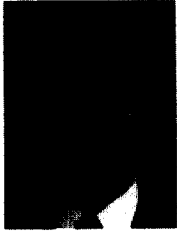
고장 수에 대한 AE를 계산하고, NAE와 순위를 계산하여 다양한 모델의 결과와 비교함으로써 제안된 모델의 적합성을 평가하였다.

실험 결과 가변 시간 간격 데이터에 대해 미래의 특정 시점에서의 고장 수를 예측하기 위해서는 미래의 특정 시점에 대한 정보인  $\Delta t_i$ 를 **신경망의 입력으로** 이용하는 모델(R4-R5)의 예측력이 가장 좋음을 알 수 있었다.  $\Delta t_i$  정보를 활용하지 않은 널리 사용되는 기존의 대표적인 통계적 SRGMs와 신경망 모델들과 비교 결과 특별히 좋은 결과를 얻지 못하였다. 이는 데이터의 크기가 적고, 데이터 수집의 어려움으로 다양한 많은 데이터에 적용하지 못한 이유로 발생한 결과로 생각된다.

본 연구 결과를 기반으로 하여,  $\Delta t_i$  정보를 활용하여 다른 잘 알려진 신경망에 적용하거나 보다 많은 데이터를 이용한 일반화된 모델의 개발이 필요하다. 따라서, 추후로는 이 부분에 대한 연구가 수행될 것이다.

## 참 고 문 헌

- [1] G. Cybenko, "Approximation by Super-positions of A Sigmoidal Function," *Mathematics of Control, Signals and Systems*, Vol.2, pp.303-314, 1989.
- [2] S. E. Fahlman and C. Lebiere, "The Cascade- Correlation Learning Architecture," *Advances in Neural Information Processing Systems II*, pp.525-532, 1990.
- [3] J. Ghosh and K. Tumer, "Structural Adaptation and Generalization in Supervised Feedforward Networks," *Journal of Artificial Neural Networks*, Vol. 1, No.4, pp.431-458, 1994.
- [4] A. L. Goel, "Software Reliability Models Assumptions, Limitation, and Applicability," *IEEE Trans. on Software Eng.* Vol.SE-11, No.12, pp.1411-1423, 1985.
- [5] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja, "Neural and Statistical Classifiers-Taxonomy and Two Case Studies," *IEEE Trans. on Neural Networks*, Vol.8, No.1, pp.5-17, 1997.
- [6] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer Feedforward Networks are Universal Approximators." *Neural Networks*, Vol. 2, pp. 359-366, 1989.
- [7] R. H. Hou and S. Y. Kuo, "Applying Various Learning Curves to Hypergeometric Distribution Software Reliability Growth Model," **IEEE**, 1994.
- [8] N. Karunanithi, D. Whitley and Y. K. Malaiya, "Using Neural Networks in Reliability Prediction," *IEEE Software.*, pp.53-59, 1992.
- [9] N. Karumanithi, D. Whitley, and Y. K. Malaiya, "Prediction of Software Reliability Using Connectionist Models," *IEEE Trans. on Software Eng.*, Vol. 18, No.7, pp.563-574, July. 1992.
- [10] T. M. Khoshgoftaar, E. B. Allen, J. P. Hudepohi, and S. J. Aud, "Application of Neural Networks to Software Quality Modeling of a very Large Telecommunications Systems," *IEEE Trans. on Neural Networks*, Vol.8, No.4, pp.902-909, 1997.
- [11] T. Y. Kwok and D. Y. Yeung, "Constructive Algorithms for Structure Learning in Feedforward Neural Networks for Regression Problems," *IEEE Trans. on Neural Networks*, Vol.8, No.3, pp.630-645, 1997.
- [12] M. R. Lyu, "*Handbook of Software Reliability Engineering*," IEEE Computer Society Press, 1996.
- [13] Y. K. Malaiya, N. Karunanithi, and P. Verma, "Predictability Measures for Software Reliability Models," *IEEE Trans. on Reliability*, Vol.41, No.4, pp.539-546, 1992.
- [14] J. D. Musa, A. Iannino, and K. Okumoto, "*Software Reliability : Measurement, Prediction, Application*," McGraw-Hill, 1987.
- [15] M. Ohba, "Software Reliability Analysis Models," *IBM Journal of Research and Development*, Vol.21, No.4, pp.428-443, 1984.
- [16] J. Y. Park, S. U. Lee, and J. H. Park, "Neural Network Modeling for Software Reliability Prediction from Failure Time Data," *Journal of Electrical Eng. and Information Science*, Vol.4, No.4, pp.533-538, 1999.
- [17] F. Poptentiu and D. N. Boros, "Software Reliability Growth Supermodels," *Microelectron. Reliab.* Vol. 36, No.4, pp.485-491, 1996.
- [18] 이상운, "비정규화 데이터를 이용한 신경망 소프트웨어 신뢰성 예측", 정보처리학회 논문지, 제7권, 제5호, pp.1419-1426, 2000. 5.



### 이 상 운

e-mail : sangun\_lee@hanmail.net  
 1983년~1987년 한국항공대학교  
 항공전자 공학과(학사)  
 1995년~1997년 경상대학교 컴퓨터  
 과학과(석사)  
 1998년~2000년 경상대학교 컴퓨터  
 과학과 박사 수료

1992년~현재 국방품질관리소 항공전자장비 품질보증 담당  
 관심분야 : 소프트웨어 공학(소프트웨어 시험 및 품질  
 보증, 소프트웨어 신뢰성), 신경망, 퍼지



### 박 영 목

e-mail : ympark@cjcc.chinju.ac.kr  
 1983년~1989년 경상대학교 전산통  
 계학과 학사  
 1997년~1999년 경상대학교 산업정  
 보공학과 석사  
 2000년~현재 경상대학교 컴퓨터  
 과학과 박사과정

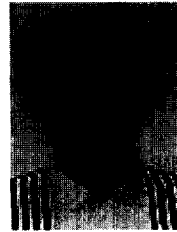
관심분야 : 소프트웨어 공학(소프트웨어 시험 및 소프  
 트웨어 신뢰성), 신경망, 멀티미디어



### 박 수 진

e-mail : Lelia@thrunet.com  
 1986년~1990년 경상대학교 전산통  
 계학과 이학사  
 1993년~1995년 경상대학교 컴퓨터  
 과학과 공학석사  
 1998년~2000년 경상대학교 컴퓨터  
 과학과 박사 수료

관심분야 : 소프트웨어 공학(소프트웨어 테스트 및 품  
 질보증, 소프트웨어 신뢰성)



### 박 재 홍

e-mail : pjh@nongae.gsnu.ac.kr  
 1973년~1978년 충북대학교 수학과  
 학사  
 1978년~1980년 중앙대학교 전산  
 학과 석사  
 1985년~1988년 중앙대학교 전산  
 학과 박사

1983년~현재 경상대학교 컴퓨터과학과 교수  
 관심분야 : 소프트웨어 신뢰성, 시험도구 자동화, 시스  
 템 분석 및 설계, 신경망