한국정보전자통신기술학회논문지(jkiiect)'18-12, Vol.11 No.6
논문 18-11-06-800 http://dx.doi.org/10.17661/jkiiect.2018.11.6.800

다양한 위험함수에 의존한 소프트웨어 신뢰모형의 적용에 대한 비교 평가에 관한 연구

양 태 진*

A Study on Comparative Evaluation of Application of Software Reliability Model Dependent on Various Hazard Functions

Tae-Jin Yang*

요 약 소프트웨어 효율성은 운용 환경에서 사용시간에 따라 고장없이 사용할 수 있는 확률이며, 소프트웨어 시스템 안정성에 영향을 미치는 가장 근본적인 요인이다. 정보기술 분야에서 활용되고 있는 컴퓨터 시스템의 오작동은 관련 산업분야에 중요한 손실을 야기할 수도 있다. 따라서, 본 연구에서는 소프트웨어 고장시간 자료를 가지고 유한고장 NHPP 모형에 기반하여 다양한 위험함수에 의존한 소프트웨어 신뢰성 모형의 속성을 분석 하였다. 제안한 모형의 위험함수 패턴은 Goel-Okumoto모형은 상수가 되고, Minimax 모형과 Rayleigh모형은 증가패턴을 따르지만, 위험함수의 증가폭은 Minimax 모형이 Rayleigh모형과 Goel-Okumoto모형 보다 작은 것으로 나타났다. 또한, 평균값 함수m(t)의 참값 오차와 평균제곱오차(MSE)는 Minimax 모형이 모두 Rayleigh 모형과 Goel-Okumoto모형 보다 작아서 상대적으로 효율적이였다. 본 연구의 결과는 소프트웨어 개발자에게 위험함수에 관한 기본정보로 활용될 수 있을 것으로 기대한다.

Abstract Software efficiency is the probability of failure free use in operating environments, and is the most fundamental factor affecting software system stability. The malfunction of the computer system used in the information technology field may cause a significant loss in the related industry. Therefore, in this study, we analyze the attributes of software reliability models that depend on various hazard functions based on finite fault NHPP model with software failure time data. The hazard function pattern of proposed model is constant for the Goel-Okumoto model, and the Minimax and Rayleigh models follow the incremental pattern, but the hazard function increase value of the Minimax model is smaller than that of the Rayleigh model and the Goel-Okumoto model. Also, the Minimax model was relatively efficient because the true value error of the mean value function m(t) and the mean square error (MSE) of the Minimax model were smaller than those of the Rayleigh and Goel-Okumoto models. The results of this study are expected to be useful for software developers as basic information about the hazard function.

Key Words : Box-plot, hazard function, Minimax model, Mission time, MSE, NHPP.

1. 서론

소프트웨어 효율성은 운용환경에서 사용시간의 따라 고장 없이 사용 될 수 있는 확률이라고 정의 할 수 있다. 따라서 소프트웨어 효율성은 소프트웨어 시스템 안정성에 영향을 미치는 근본적인 핵심요인이며 따라서 컴퓨터 시스템의 오작동은 우리사회의 큰 손실을 유발 할 수도 있다. 결국 소프트웨어 운용과정에서 소프트웨어 시스템의 정상운용 문제

Funding for this paper was provided by Namseoul University year 2018
*Corresponding Author : Department of Electronic Engineering, Namseoul University(solomon645@nsu.ac.kr)
Received November 29, 2018 Revised December 12, 2018 Accepted December 17, 2018

는 주요 근본적인 과제이다. 이러한 상황을 분석하기 위해서는 운영자 소프트웨어 요구 사항을 반영해야한다. 그러므로 소프트웨어 개발의 비용 측면에서 효율적으로 경제적 비용을 투자하기 위해서는 소프트웨어의 테스트 비용과 신뢰성의 정도는 미리 고장현상을 예측된 경우에는 효율적인 개발 과정이 된다. 따라서 소프트웨어를 소비자에게 방출 예상시간을 고려한 신뢰성 비용 및 소프트웨어 개발 과정을 파악하는 작업 과정은 필수 불가결하다 [1, 2]. 따라서 소프트웨어 오작동 탐색 과정에서 소프트웨어 신뢰성모형은 비동질 포아송 과정(NHPP)을 사용하여 새로운 결함이 유발되면 즉시 제거되고 새로운 결함은 디버깅 프로세스에서 유발되지 않은 다는 가정하에서 제안되어 왔다[3, 4].

본 연구에서는 소프트웨어 고장시간 자료를 적용한 NHPP 모형을 기반으로 위험함수의 형태가 상수인 Goel-Okumoto 모형과 증가형 패턴을 따르는 Minimax 모형과 Rayleigh 모형을 적용하여 소프트웨어 신뢰성 모형의 속성을 비교, 평가하고자 한다.

2. 배경연구

2.1 기존의 Goel-Okumoto-NHPP 모형

소프트웨어 신뢰성 분야에서 기본모형인 Goel-Okumoto모형[5]은 유한고장 상황에서 고장의 원인이 되는 결함의 기대 값을 θ 라고 표현하고 결함 탐색비율을 β 라고 하면 NHPP 모형에서 결함 탐색비율 β 는 고정상수로 간주하여 정의하였다. 따라서 이 상황을 $f(t)$을 고장밀도함수이고 $F(t)$을 누적분포함수라고 하면 유한고장 NHPP 모형의 강도함수와 평균값함수를 적용시키면 다음과 같다 [1,3].

$$\lambda(t|\theta,\beta) = \theta f(t) = \theta\beta e^{-\beta t} \quad (1)$$

$$m(t|\theta,\beta) = \theta F(t) = \theta(1-e^{-\beta t}) \quad (2)$$

한편, (3)식을 이용한 NHPP모형에 대힌 우도함수는 다음과 같다 [6].

$$L_{NHPP}(\Theta|\underline{x}) = \left(\prod_{i=1}^{n}\lambda(x_i)\right)\exp[-m(x_n)] \quad (3)$$

단, $\underline{x}=(0\le x_1\le x_2\le\cdots\le x_n)$

Goel-Okumoto인 경우 (3)식을 이용한 최우추정법을 이용한 최우추정량 $\hat{\theta}_{MLE}$와 $\hat{\beta}_{MLE}$ 모수 추정은 과정조건을 만족해야 한다.

$$\frac{n}{\hat{\theta}} = 1-e^{-\hat{\beta}x_n} \quad (4)$$

$$\frac{n}{\hat{\beta}} = \sum_{i=1}^{n}x_i + \hat{\theta}\,x_n\,e^{-\hat{\beta}x_n} \quad (5)$$

2.2 수명분포를 미니맥스와 레일리 분포에 적용한 유한고장 NHPP 모형

미니맥스 분포(Minimax distribution)는 베타분포(Beta distribution)에 비해 사용하기가 비교적 쉽다는 이유로 베타분포의 대안으로 사용되어 왔고 미니맥스 분포의 확률밀도 함수(PDF)와 누적분포함수(CDF)는 각각 다음과 같다[5].

$$f(t) = ab\,t^{a-1}(1-t^a)^{b-1}, \qquad t\in(0,1] \quad (6)$$

$$F(t) = 1-(1-t^a)^b, \quad (7)$$

(6)식과 (7)식에서 a와 b는 형상모수를 의미한다. 그리고 미니맥스 분포에서 Oguntunde와 Adejumo[6]는 $a=1$ 인 경우를 분석하였다. 따라서, 본 연구도 형상모수 a가 1 인 경우를 선택하여 소프트웨어 신뢰성 모형을 분석하고자 한다.

(3)식에 (6)식과 (7)식을 이용하여 미니맥스 고장확률 밀도함수를 적용하면 다음과 같은 형태로 유도할 수 있다.

$$L_{NHPP}(\Theta|\underline{x}) = \left(\prod_{i=1}^{n}ab\,x_i^{a-1}(1-x_i^a)^{b-1}\right) \cdot\exp[-(1-(1-x_n^a)^b)] \quad (8)$$

최우추정법(MLE)을 위하여 (8)식을 이용한 로그우도함수는 다음과 같이 유도할 수 있다[2].

$$\begin{aligned}&\ln L(\Theta|\underline{x}) \quad (9)\\ &= n\ln\theta + n\ln a + n\ln b + (a-1)\sum_{i=1}^{n}\ln x_i\\ &-(b-1)\sum_{i=1}^{n}\ln(1-x_i^a) - \theta[1-(1-x_n^a)^b]\end{aligned}$$

(9)식을 이용하고 형상모수 $a=1$인 경우를 이용하

면 다음 조건식을 만족하는 최우추정량 $\hat{\theta}_{MLE}$와 $\hat{b}_{MLE}$ 을 추정 할 수 있다[12].

$$\frac{\partial \ln L(\Theta \mid \underline{x})}{\partial \theta} = \frac{n}{\theta} - [1-(1-x_n^a)^b] = 0 \quad (10)$$

$$\frac{\partial \ln L(\Theta \mid \underline{x})}{\partial b} = \frac{n}{b} + \sum_{i=1}^{n} \ln(1-x_i^a) + \theta(1-x_n^a)^b \ln(1-x_n^a) = 0 \quad (11)$$

레일리(Rayleigh)형 모형에서도 같은 방법으로 최우추정값 $\hat{\theta}_{MLE}$와 $\hat{\beta}_{MLE}$을 추정 할 수 있다 [6].

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial \theta} = \frac{n}{\theta} - 1 + e^{-\beta x_n^2} = 0 \quad (12)$$

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial \beta} = \frac{n}{\beta} - \sum_{i=1}^{n} x_i^2 - \theta x_n^2 e^{-\beta x_n^2} = 0 \quad (13)$$

2.3 제안 모형에 대한 위험함수

본 연구에서 제안한 모형의 확률밀도함수와 위험함수는 [표 1]과 같다.

표 1. 확률밀도함수와 위험함수
Table 1. Probability density function and hazard function

Model	$f(t)$	$h(t)=\frac{f(t)}{1-F(t)}$
Goel-Okumoto	$\beta e^{-\beta t}$	β
Minimax-type	$b(1-t)^{b-1}$	$\frac{b}{(1-t)}$
Rayleigh-type	$2\beta t \exp(-\beta t^2)$	$2\beta t$

위험 함수는 지정된 시간에서의 순간 고장률의미하고, 다음과 같이 정의 된다 [7].

$$h(t) = \frac{f(t)}{1-F(t)} \quad (14)$$

단, $f(t)$는 확률밀도 함수이고 $F(t)$는 누적분포함수를 의미한다.

3. 소프트웨어 고장시간 분석 및 신뢰속성 비교

표 2. 고장시간자료
Table 2. Failure time data

Failure number	Failure time (hours)	Failure time$\times 10^{-2}$ (hours)
1	0.479	0.00479
2	0.745	0.00745
3	1.022	0.01022
4	1.576	0.01576
5	2.610	0.02610
6	3.559	0.03559
7	4.252	0.04252
8	4.849	0.04849
9	4.966	0.04966
10	5.136	0.05136
11	5.253	0.05253
12	6.527	0.06527
13	6.996	0.06996
14	8.170	0.08170
15	8.863	0.08863
16	10.771	0.10771
17	10.906	0.10906
18	11.183	0.11183
19	11.779	0.11779
20	12.536	0.12536
21	12.973	0.12973
22	15.203	0.15203
23	15.640	0.15640
24	15.980	0.15980
25	16.385	0.16385
26	16.960	0.16960
27	17.237	0.17237
28	17.600	0.17600
29	18.122	0.18122
30	18.735	0.18735

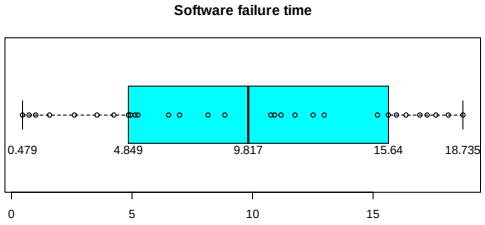

그림 1. 박스플롯의 결과
Fig. 1. Result of Box-plot

이 장에서 소프트웨어 고장시간 정보데이터[8](Failure time information data)를 활용하여 신뢰모형들의 속성을 비교 분석하고자 한다. 이 데이터는 [표 2]에 나열 되었고 이상값(극단값)이 존재하는지를 판단하기 위하여 데이터 대한 추이 검정이 수행해야 하는데 본 논문에서는 추이검정을 박스-플롯(Box-plot)d을 활용하였다[6]. 따라서 [그림 1]의 결과와 같이

$$\text{상한}(=15.04+1.5*(15.04-4.849)=30.3265)\text{과}$$
$$\text{하한}(=4.849-1.5*(15.04-4.849)=-40.6408)$$

을 만족하지 않는 데이터 값이 존재하지 않기 때문에 이상 값(극단 값)이 발생하지 않음을 확인하였 다. 따라서, 이 자료를 사용하여 신뢰성모형의 속성을 추정하는 작업이 가능하고, 효율적이다[10].

모수의 수렴성을 위하여 원래 데이터를 수치변환(Failure time$\times 10^{-2}$)하였고, 모수추정은 최우추정법을 이용하였으며, 비선형 방정식의 계산방법은 수치 해석적 방법인 이분법(Bisection method)을 활용하였다. 이러한 계산은 초기값을 0.01과 10^2을, 허용한계(Tolerance for width of interval)는 10^{-5}을 적용하고 수렴성을 확인하면서, R 소프트웨어를 이용하여 모수추정을 수행하였다. 그 결과는 [표 3]에 나타내었다.

표 3. 모수추정 및 MSE , R^2
Table 3. Parameter estimation and MSE , R^2

Model	MLE	Model Comparison	
		MSE	R^2
Goel-Okumoto	$\hat{\theta}=32.9261$ $\hat{\beta}=12.97$	32.3491	0.916
Minimax - type	$\hat{\theta}=477.188$ $\hat{b}=6.29\times10^{-1}$	1.782	0.9897
Rayleigh - type	$\hat{\theta}=30.0412$ $\hat{\beta}=188.4$	32.2867	0.897

Note.
MLE: Maximum Likelihood Estimation
MSE:Mean square error
R^2: Determination coefficient

또한, [표 3]에서 실제값과 예측값에 대한 차이에 대한 척도를 의미하는 평균제곱오차는 다음과 같이 정의 된다[9].

$$MSE=\frac{\sum_{i=1}^{n}\left(m(x_i)-\hat{m}(x_i)\right)^2}{n-k} \qquad (15)$$

단, $m(x_i)$은 시간$(0,\ x_i]$까지 나타난 에러들의 누적수를 의미하고, $\hat{m}(x_i)$는 x_i 시점까지 평균값 함수로부터 추정된 에러수를 의미한다.

따라서, Minimax 모형인 경우가 Rayleigh 모형과 Goel-Okumoto모형보다 상대적으로 가장 작으므로 Minimax 모형인 경우가 효율적이라 판단된다. 이러한 평균오차제곱을 구체적으로 알기 위하여 각 시점 $(i=1,2,\cdots,30)$에 대한 평균제곱오차($MSE(i)=\left[m(x_i)-\hat{m}(x_i)\right]^2$)에 대한 시뮬레이션은 [그림 2]에 나타내었다[11].

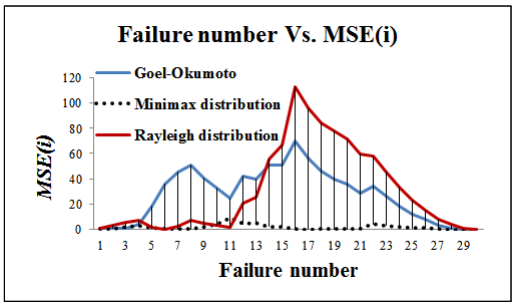

그림 2. 각 시점에 대한에 평균제곱오차 추이
Fig. 2. Transition of mean square error for each failure number

[그림 2]에서 각 시점에서 Minimax 모형인 경우가 제일 낮게 나타나고, Rayleigh 모형은 초반부에서 낮게 나타나고 일정 시점이후에는 크게 증가하고 있음을 알 수 있다.

표 4. 강도함수의 추정
Table 4. Estimation of the intensiry function

Failure time	Goel-Okumoto	Minimax distribution	Rayleigh distribution
0.00479	12.97	0.314506486	1.8049678
0.00745	12.97	0.315349353	2.807309
0.01022	12.97	0.31623189	3.8511004
0.01576	12.97	0.318011867	5.9386832
0.0261	12.97	0.321388233	9.835002
0.03559	12.97	0.324550762	13.4110238
0.04252	12.97	0.326899779	16.0223864
0.04849	12.97	0.328950826	18.2720018
0.04966	12.97	0.329355809	18.7128812
0.05136	12.97	0.329946028	19.3534752
0.05253	12.97	0.330353468	19.7943546
0.06527	12.97	0.334856055	24.5950414
0.06996	12.97	0.336544665	26.3623272
0.0817	12.97	0.340847218	30.786194
0.08863	12.97	0.343438998	33.3975566
0.10771	12.97	0.350782817	40.5872822
0.10906	12.97	0.351314342	41.0959892
0.11183	12.97	0.352410012	42.1397806
0.11779	12.97	0.354790809	44.3856278
0.12536	12.97	0.35786152	47.2381552
0.12973	12.97	0.359658497	48.8848586
0.15203	12.97	0.369116832	57.2879446
0.1564	12.97	0.371028924	58.934648
0.1598	12.97	0.37253035	60.215836
0.16385	12.97	0.374334749	61.741957
0.1696	12.97	0.376926782	63.908672
0.17237	12.97	0.378188321	64.9524634
0.176	12.97	0.379854369	66.32032
0.18122	12.97	0.382276069	68.2873204
0.18735	12.97	0.385159663	70.597227

[표 4]에서는 제안 모형에 대한 강도함수의 추정값을 나타내었다. Goel-Okumoto모형은 고장시간에 독립적으로 상수가 되고 Minimax 모형과 Rayleigh모형은 증가패턴을 따르지만 Rayleigh 모형과 보다 Minimax 모형이 증가폭이 낮은 것(소폭)으로 나타나는 형태을 보이고 있다. 이러한 추세는 [그림 3]에 요약되었다[8].

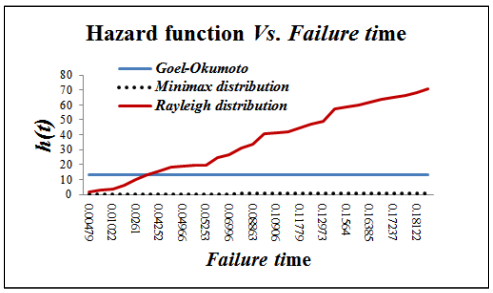

그림 3 강도함수의 추세
Fig. 3. Transition of Intensity function

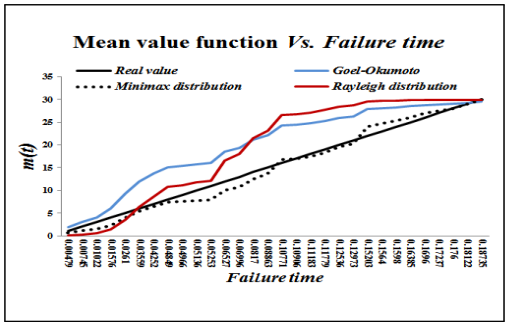

그림 4. 각 모형에 대한 평균값 함수 추이
Fig. 4. Mean value function trend for each model

[그림 4]에서는 각 모형에 대한 평균값 함수 패턴에 대한 추세를 나타내었다. 적용고장시간에서 대체적으로 비감소 특성을 가진다. 특히 Minimax 모형이 Goel-Okumoto모형과 Rayleigh모형 보다 상대적으로 참값(Observed number)과의 차이의 폭이 작게 예측되는 패턴을 보여 효율적이다.

NHPP 모형에서 테스트 시점 $x_{30}=0.18735$ 에서 소프트웨어 고장이 발생하고 $(0.18735, 0.18735+Mission\ time]$사이에서 소프트웨어 고장이 발생하지 않을 확률인 신뢰도(Reliability)$\hat{R}(Mission\ time \mid 0.18735)$는 다음과 같은 형태로 표현된다[6].

$$\hat{R}(Mission\ time \mid 0.18735) = e^{-\int_{0.18735}^{0.18735+Mission\ time}\lambda(\tau)d\tau}$$

$$= \exp[-\{m(Mission\ time+0.18735)-m(0.18735)\}] \quad (16)$$

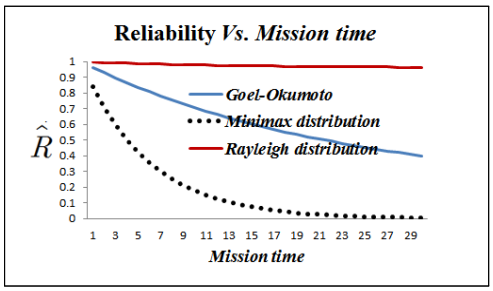

그림 4 신뢰도의 추세
Fig. 4. Transition of reliability

[그림 4]의 임무시간에 대한 신뢰도 그림에서는 실행시간에 따라 Rayleigh 모형인 경우가 Goel-Okumoto 모형과 Minimax 모형보다 상대적으로 높고 안정적인 신뢰도를 보이고 있다.

4. 결론

소프트웨어 개발 과정에서 테스트 과정이나 실제 소프트웨어 사용과정에서 내재된 고장발생 현상 혹은, 고장발생 원인을 예측 할 수 있으면 소프트웨어 성능을 비교 분석한 후 상대적으로 효율성 비교 평가를 수행 할 수 있다. 본 논문에서는 소프트웨어 고장시간 자료를 가지고, 유한고장 NHPP 모형을 기반으로 위험함수의 형태가 상수인 Goel-Okumoto 모형과 증가형 패턴을 따르는 Minimax 모형과 Rayleigh 모형을 적용하여 소프트웨어 신뢰성 모형의 속성을 비교, 분석하였다.

본 연구의 결과를 다음과 같이 요약 할 수 있다.

첫째, 본 연구에서 제안한 모형들의 위험함수의 형태는 Goel-Okumoto 모형은 고장시간에 독립적으로 상수가 되고, Minimax 모형과 Rayleigh모형은 증가형 패턴을 따르지만, 위험함수의 증가폭은 Minimax 모형이 Rayleigh모형과 Goel-Okumoto 모형 보다 작은 것으로 나타났다.

둘째, 효율성의 척도인 평균제곱오차 (MSE)는 Minimax모형이 Rayleigh 모형과 Goel-Okumoto 모형보다 작으므로 Minimax모형이 가장 효율적이였다.

셋째, 제안한 모형에 대한 평균값 m(t)는 참값(True value)과 비교했을 때 Minimax 모형이 Goel-Okumoto 모형과 Rayleigh 모형보다 추정폭(에러폭)이 작게 예측되여 가장 효율적이였다.

넷째, 임무시간에 대한 신뢰도는 실행시간에 따라 Rayleigh 모형인 경우가 Goel-Okumoto모형과 Minimax 모형보다 높고 안정적인 추세를 보였다.

본 연구의 결과는 소프트웨어 개발자에게 위험함수에 관한 사전 기본정보를 제공 할 수 있으리라 판단된다.

REFERENCES

[1] S.S. Gokhale & K.S. Trivedi, "A time/structure based software reliability model", Annals of Software Engineering, Vol.8, pp.5-12, 1999.

[2] T. J. Yang, “The Comparative Analysis of Software Failure Time Based on Software Reliability Model and Nonlinear Regression Model”, Korea Knowledge Information Technology Society, Vol.9, No.6, pp.723-731, 2014.

[3] Goel A L, Okumoto K, "Time-dependent fault detection rate model for software and other performance measures", IEEE Transactions on Software Engineering, Vol.28, pp.206-211, 1978.

[4] Kuei-Chen, C., Yeu-Shiang, H.,&Tzai-Zang, L.,“A study of software reliability growth from the perspective of learning effects”, Reliability Engineering and System Safety, Vol.93, pp.1410-1421, 2008

[5] P. E, Oguntunde, A. O. Adejumo, “ANote on the Minimax Distribution”, Covenant Journal of Physical and Life Sciences (CJPL), Vol.3, No.1. pp.1-8, 2015.

[6] H. C. KIM, "A Comparative Study for Statistical Process Control of Software Reliability Model on Finite and Infinite NHPP Using Rayleigh Distribution", International Journal of Soft Computing, Vol.11, No.3,

pp.165-171, 2016.
[7] T.H. Yoo, "The Infinite NHPP software reliability model based on Monotonic Intensity Function", Indian Journal of Science and Technology, Vol.8, No.14, pp.1-7, 2015.
[8] Y. Hayakawa, G. Telfar "Mixed poisson -type processes with application in software are reliability", Mathematical and Computer Modelling, Vol.31, pp.151-156, 2000.
[9] T. J. Yang, "A Comparative Study on Reliability Attributes for Software Re-liability Model Dependent on Lindley and Erlang Life Distribution", The Journal of Korea Institute of Information, Electronics and Commmunication Technology, Vol.10, No.5, pp.469-475, 2017.
[10] T. J. Yang, "A Performance Comparative Evaluation for Finite and Infinite Failure Software Reliability Model using the Erlang Distribution", The Journal of Korea Institute of Information, Electronics and Commmunication Technology, Vol.9, No.4, pp.351-358, 2016.
[11] T. J. Yang, "The Performance Analysis Comparative Study depend on Software Reliability Model and Curve Regression Model", Medwell Journals, Vol.12, No.5, pp313-317, 2017.
[12] T. J. Yang, J. G. Park, "A Comparative Study of the Software NHPP Based on Weibull Extension Distribution and Fle- xible Weibull Extension Distribution", International Journal of Software Computing, Vol.11, No.4, pp.276-281, 2016.
[13] Jeong-Joon Kim, Kwang-Jin Kwak, Don-Hee Lee, Yong-Soo Lee, 'Study of Trust Bigdata Platform', The Journal of The Institute of Internet, Broadcasting and Communication VOL. 16 No. 6, 2016

저자약력

양 태 진 (Tae-Jin Yang) **[정회원]**

- 1992년 2월 : 한양대학교 전자공학과 공학석사
- 1995년 2월 : 한양대학교 전자공학과 공학박사(수료)
- 1986년 3월 ~ 1992년 2월 : 현대 그룹 신규사업부 기술과장
- 1993년 3월 ~ 2013년 12월 : 호서 대학교 호서전문학교 정보통신과 교수
- 2014년 3월 ~ 현재 : 남서울대학교 전자공학과 교수

〈관심분야〉 소프트웨어 신뢰성공학, 인공지능 (Artificial-Intelligent), Intelligent-Network & Network-Security