S-분포형 결함 발생률을 고려한 NHPP 소프트웨어 신뢰성 모형에 관한 비교 연구

김희철* · 김경수**

요 약

본 연구에서는 소프트웨어 제품 테스팅 과정에서 관측고장시간에 근거한 결함 발생률을 고려한 소프트웨어 신뢰성 모형에 대하여 연구 하였다. 신뢰성 분야에서 많이 사용되는 S-분포모형을 이용한 새로운 결함 확률을 추가한 문제를 제시하였다. 수명분포는 유한고장 비동질적인 포아송과정을 이용하였다 본 논문의 결함 발생률을 고려한 소프트웨어 고장 자료 분석에서는 고장 시간 자료를 적용하였으며 모수추정 방법은 최우추정법을 이용하여 결함 발생 확률에 대한 관계와 신뢰도를 추정 하였다.

The Comparative Software Reliability Model of Fault Detection Rate Based on S-shaped Model

Kim Hee Cheul* · Kyung-Soo Kim**

ABSTRACT

In this paper, reliability software model considering fault detection rate based on observations from the process of software product testing was studied. Adding new fault probability using the S-shaped distribution model that is widely used in the field of reliability problems presented. When correcting or modifying the software, finite failure non-homogeneous Poisson process model was used. In a software failure data analysis considering the time-dependent fault detection rate, the parameters estimation using maximum likelihood estimation of failure time data and reliability make out.

Key words : Fault Detection Rate of S-shaped distribution , NHPP, Mission Time.

접수일(2013년 2월 21일), 수정일(1차: 2013년 3월 17일), 게재확정일(2013년 3월 23일)

* 남서울대학교 산업경영공학과
** 백석문화대학교 인터넷정보학부

1. 서 론

소프트웨어 고장으로 인한 컴퓨터 시스템의 고장은 우리 사회에 엄청난 손실을 유발 할 수 도 있다. 따

라서 소프트웨어 개발 과정에서 소프트웨어 신뢰성은 중요한 문제이다. 이 문제는 사용자의 요구조건과 테스팅 비용을 만족시켜야 한다. 소프트웨어 테스팅(디버깅)면에서 비용을 줄이기 위해서는 소프트웨어의 신뢰성의 변동과 테스팅 비용을 사전에 알고 있어야 효율적이다. 따라서 신뢰도, 비용 및 방출 시간의 고려사항을 가진 소프트웨어 개발 과정은 필수 불가결하다. 결국 소프트웨어 제품의 결함내용을 예측하기 위한 모형 개발이 필요하다. 지금까지 많은 소프트웨어 신뢰성 모형이 제안 되었다. 이 중에서 비동질적 포아송 과정(non-homogeneous Poisson process; NHPP)에 의존한 모형은 에러 탐색 과정측면에서는 우수한 모형이고 이 모형은 결함이 발생하면 즉시 제거되고 디버깅 과정에서 새로운 결함이 발생되지 않는다는 가정을 하고 있다.

이 분야에서 Gokhale과Trivedi [1]은 고양된 비동질적인 포아송 과정 모형(enhanced NHPP) 모형을 제시하였고 Goel 과 Okumoto [2]은 지수적 소프트웨어 신뢰성 모형(exponential software reliability growth model)을 제안 하였다. 이 모형은 결함의 누적수가 S 형태나 지수적 형태(S-shaped or exponential-shaped)를 가진 평균값 함수(mean value function)를 이용하였다. 이러한 모형에 의존한 일반화 모형은 Yamada 와 Ohba [3]에 의해 지연된 S-형태 신뢰 성장모형(delayed S-shaped reliability growth model)과 변곡된 S-형태 신뢰성장모형(inflection S-shaped reliability growth model)이 제안되었다. Zhao [4]는 소프트웨어 신뢰도에서 변환점 문제를 제시하였고 Shyur [5]는 변환점을 이용한 일반화한 신뢰도 성장 모형을 제안하였다. Pham와 Zhang[6]는 테스팅 커버리지(coverage)를 측정하여 소프트웨어 안정도를 평가할 수 있는 소프트웨어 안정도 모형을 제시했다. 비교적 최근에, Huang [7]은 일반화 로지스틱 테스팅 노력 함수(generalized logistic testing-effort function)와 변환점 모수(change-point parameter)를 통합하여 효율적인 소프트웨어 신뢰성 예측 기술을 제시하기도 하였다. 그리고 최근에는 S-형태 모형은 소프트웨어 관리자들이 소프트웨어 및 검사 도구에 익숙해지는 학습과정을 설명할 수 있다고 하였고[8] 또한, 대수 선형 위험함수를 이용한 학습과정 특성을 연구하기도 하였다[9].

또한 대부분 연구에서는 테스팅하는 과정에서 새로운 결함이 나타나지 않는다는 가정을 하여 분석하는 연구가 대부분이었다[10]. 그러나 본 연구에서는 테스팅 하는 과정에서 새로운 결함이 나타날 확률을 반영하는 결함 탐색비율을 포함한 모형을 제시하고자 한다. 본 연구에서는 수명분포는 유한고장수를 가진 비동질적인 포아송 과정에 기초하고 S-모형을 이용한 소프트웨어 신뢰성모형에 대하여 논의 되었다.

2. 관련 연구

2.1 유한고장 NHPP 모형

신뢰도에서 관측시간 $(0, t]$ 사이에 발견된 고장 수 $N(t)$을 모형 화 하는데 비동질적 포아송 과정 (non-homogeneous Poisson process; NHPP)이 널리 사용하여 왔다. 이 과정(process)에서 강도함수(intensity function) 혹은 고장 발생률(rate of occurrence of failure; ROCOF) $\lambda(t) = d\,E[N(t)]/dt$ 은 t 에 대한 단조(monotonic)함수로 흔히 가정 한다[1]. 이 범주에서 지금까지 알려진 모형들은 Goel-Okumoto 모형, Weibull 모형 그리고 Cox-Lewis 모형 등이 있는데 이모형들에 대한 강도함수는 각 각 시간에 의존한 함수, 멱(power) 함수, 대수 선형(log-linear) 함수를 가정하였다[11].

NHPP 모형에서 평균값 함수 $m(t)$ (mean value function)와 강도 함수 $\lambda(t)$ 는 다음과 같은 관계로 표현할 수 있다.

$$m(t) = \int_0^t \lambda(s)ds, \quad \frac{dm(t)}{dt} = \lambda(t) \tag{1}$$

$N(t)$는 모수 $m(t)$을 가진 포아송 확률밀도함수(probability density function)로 알려져 있다. 즉,

$$P\{N(t) = n\} = \frac{[m(t)]^n \cdot e^{-m(t)}}{n!}, \ n = 0,1,2,\dots\infty \tag{2}$$

이처럼 시간관련 모형(time domain models)들은 NHPP에 의해서 확률 고장 과정으로 설명이 가능하다. 이러한 NHPP 모형들은 유한 고장 모형과 무한 고장 범주로 분류한다[12]. 유한 고장(finite failure) NHPP 모형들은 충분한 테스트 시간이 주어지면 결함들(faults)의 기대 값이 유한 값$\left(\lim_{t\to\infty} m(t) = \theta < \infty\right)$을 가지고 반면에 무한 고장(infinite failure) NHPP 모형들은 무한 값을 가진다고 가정 된다. 유한 고장 NHPP 모형에서 충분한 테스트 시간이 주어졌을 때 탐색되어 질 수 있는 결함의 기대 값을 θ라고 표현하고 $F(t)$를 분포함수라고 표현하면 유한 고장 NHPP모형의 평균값 함수는 다음과 같이 표현 할 수 있다[1][11][12].

$$m(t) = \theta F(t) \tag{3}$$

(3)식으로 부터 강도함수(failure intensity)$\lambda(t)$는 다음과 같이 유도된다.

$$\lambda(t) = \theta F'(t) = \theta f(t) \tag{4}$$

또한, 시간 $(0,t]$까지 조사하기 위한 시간 절단(time truncated)모형은 n번째 까지 고장시점 자료를

$$x_k = \sum_{i=1}^{k} t_k \ (k = 1,2,\cdots,n\,;\, 0 \le x_1 \le x_2 \le \cdots \le x_n) \tag{5}$$

이라고 하면 데이터 집합 D_t는 $\{n, x_1, x_2 \cdots x_n ; t\}$와 같이 구성된다. n번째까지 고장시점이 관찰된 고장 절단 모형일 경우에 데이터 집합 D_{x_n}은 $\{x_1, x_2 \cdots x_n\}$으로 구성되며 이 시간 절단 모형에서의 Θ를 모수공간이라고 표시하면 우도함수는 다음과 같이 알려져 있다[9][12].

$$L_{NHPP}(\Theta | \underline{x}) = \left(\prod_{i=1}^{n} \lambda(x_i)\right) \exp(-m(x_n)) \tag{6}$$

단, $\underline{x} = (x_1 < x_2 < x_3, \cdots, x_n]$

NHPP 모형에서 테스트 시점 x_n에서 소프트웨어 고장이 일어난다고 하는 가정 하에서 신뢰구간 $(x_n, x_n + t]$ (단, t는 임무시간(mission time))사이에서 소프트웨어의 고장이 일어나지 않을 확률인 신뢰도(reliability) $\hat{R}(t \mid x_n)$는 다음과 같이 됨이 알려져 있다[1][9].

$$\begin{aligned} \hat{R}(t \mid x_n) &= e^{-\int_{x_n}^{x_n+t} \lambda(\tau) d\tau} \\ &= \exp[-\{m(t+x_n) - m(x_n)\}] \end{aligned} \tag{7}$$

2.2 기본적인 Goel-Okumoto 신뢰성 모형

이 분야에서 기본 모형인 Goel-Okumoto모형은 유한고장 상황에서 고장의 원인이 되는 결함의 기대 값을 θ라고 표현하고 b을 결함 탐색률 이라고 하면 NHPP와 관련된 평균값 함수와 강도함수는 다음과 같이 정의 된다[10].

$$m(t|\theta, b) = \theta(1 - e^{-bt}) \ (\theta > 0\,,\, b > 0) \tag{8}$$

$$\frac{dm(t|\theta,b)}{dt} = \lambda(t|\theta,b) = b(\theta - m(t)) \tag{9}$$

위 모형에서 테스팅 하는 과정에서 하나의 결함을 제거하는 동안에 새로운 결함이 발생할 확률을 B라고 하면 NHPP와 관련된 결함의 기대값 θ와 결함 탐

색 b는 다음과 같이 변형된다고 하였다[13].

$$\theta^M = \frac{\theta}{1-B},\ b^M = (1-B)\phi = (1-B)kb \qquad (10)$$

2.3 효율적인 모형 선택 기준

최근에 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE(평균제곱오차)와 R^2(결정계수)를 사용한다[8][9].

2.3.1 평균제곱오차(Mean square error)

평균제곱오차는 실제 관찰 값과 예측 값에 대한 차이를 측정하는 도구로서 다음과 같이 정의 된다.

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

단, $m(x_i)$은 시간$(0,\ x_i]$까지 나타난 에러들의 누적분포함수를 의미하고 $\hat{m}(x_i)$는 x_i 시점까지 평균값 함수로부터 추정된 에러의 누적개수를 의미한다. 그리고 n은 관찰 값의 수이고 τ는 모수의 수를 의미한다. 즉, MSE의 값이 작으면 효율적 모형이 된다.

2.3.2 R^2

R^2(결정계수)는 관찰 값의 차이에 대한 설명력을 나타내는 도구로서 다음과 같이 정의 된다.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(m(x_i)-\hat{m}(x_i)\right)^2}{\sum_{i=1}^{n}\left(m(x_i)-\sum_{j=1}^{n}m(x_j)/n\right)^2}$$

따라서 R^2의 값이 클수록 설명력이 커져서 효율적 모형으로 간주된다.

3. 제안하는 NHPP 모형

S-모형[1]은 Goel- Okumoto모형의 확장 모형으로 결함과 결함 사이의 시간이 Goel- Okumoto 모형 보다 길고 고장 발생률이 감소하는 특성을 가지는 모형으로 평균값 함수와 강도함수는 다음과 같이 알려져 있다.

$$m(t|\theta,b) = \theta F(t) = \theta\left[1-(1+bt)e^{-bt}\right] \qquad (11)$$

$$\frac{dm(t|\theta,b)}{dt} = \lambda(t|\theta,b) = \theta f(t) = \theta b^2 t e^{-bt} \qquad (12)$$

따라서 (9)식과 (10)식을 이용한 결함발생 확률 B을 고려한 평균값 함수와 강도함수는 다음과 같이 유도된다.

$$m(t|\theta,b,B) = \frac{\theta}{1-B}\left[1-(1+(1-B)bt)e^{-(1-B)bt}\right] \qquad (13)$$

$$\frac{dm(t|\theta,b,B)}{dt} = \theta(1-B)b^2 t e^{-(1-B)bt} \qquad (14)$$

결국 (6)식을 이용한 우도함수는 다음과 같이 유도된다.

$$L_{NHPP}(\Theta|\ \underline{x}\ ) = \left(\prod_{i=1}^{n}\theta(1-B)b^2 x_i e^{-(1-B)bx_i}\right) \qquad (15)$$
$$\cdot\ \exp\left[-\theta/(1-B)\left[1-(1+(1-B)bx_n)e^{-(1-B)bx_n}\right]\right]$$

단, $\underline{x} = (x_1 < x_2 < x_3, \cdots, x_n]$. (15)식을 이용한 최우추정법(MLE)을 이용하기 위한 로그우도함수를 구하면 다음과 같다.

$$\ln L_{NHPP}(\Theta|\ \underline{x}\ ) = \qquad (16)$$
$$n\ln\theta + n ln(1-B) + 2n\ln b + \sum_{i=1}^{n}x_i - (1-B)b\sum_{i=1}^{n}x_i$$
$$\cdot\ \exp\left[-\theta/(1-B)\left[1-(1+(1-B)bx_n)e^{-(1-B)bx_n}\right]\right]$$

따라서 모수와 에 관한 편미분식은 다음과 같이 유도 할 수 있다[11].

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial \theta} = \frac{n}{\theta} - \frac{1}{1-B}\left(1-(1+(1-B)b\,x_n)e^{-(1-B)bx_n}\right) = 0 \quad (16)$$

$$\frac{\partial \ln L_{NHPP}(\Theta \mid \underline{x})}{\partial b} = \frac{2n}{b} - (1-B)\sum_{i=1}^{n} x_i - \theta\,(1-B)\,b\,x_n^2\ e^{-(1-B)bx_n} = 0 \quad (17)$$

각 모수에 대한 최우추정량 $\hat{\theta}_{MLE}$ 와 $\hat{b}_{MLE}$ 은 다음 식을 만족한다[12].

$$\hat{\theta} = \frac{n(1-B)}{1-(1+(1-B)b\,x_n)e^{-(1-B)bx_n}} \quad (27)$$

$$\frac{2n}{\hat{b}} = (1-B)\sum_{i=1}^{n} x_i + \theta\,(1-B)\,\hat{b}\,x_n^2\ e^{-(1-B)bx_n} \quad (28)$$

4. 결함발생률을 고려한 소프트웨어 고장 자료 비교 분석

이 장에서 소프트웨어 고장 간격 시간 자료[14][15](Failure interval time data)를 가지고 제시하는 신뢰모형들을 분석하고자 한다. 이 자료의 고장 시간은 $x_{30} = 738.68$시간단위에 30번의 고장이 발생된 자료이며 <표 1>에 나열 되어 있다.

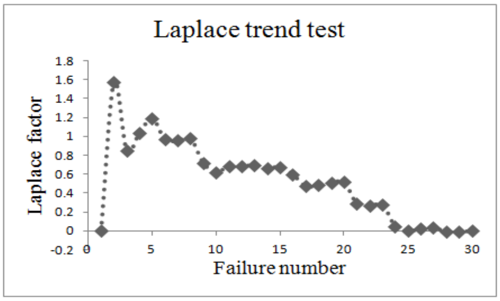
<표 1> 고장 발생 자료

Failure Number	Failure Time(hours)	Failure Interval (hours)
1	30.02	30.02
2	31.46	1.44
3	53.93	22.47
4	55.29	1.36
5	58.72	3.43
6	71.92	13.2
7	77.07	5.15
8	80.9	3.83
9	101.9	21
10	114.87	12.97
11	115.34	0.47
12	121.57	6.23
13	124.97	3.4
14	134.07	9.1
15	136.25	2.18
16	151.78	15.53
17	177.5	25.72
18	180.29	2.79
19	182.21	1.92
20	186.34	4.13
21	256.81	70.47
22	273.88	17.07
23	277.87	3.99
24	453.93	176.06
25	535	81.07
26	537.27	2.27
27	552.9	15.63
28	673.68	120.78
29	704.49	30.81
30	738.68	34.19

또 한 제시하는 신뢰 모형들을 분석하기 위하여 우선 자료에 대한 추세 검정이 선행 되어야 한다[16].

추세 분석에는 일반적으로 라플라스 추세 검정(Laplace trend test)을 사용한다. 이 검정을 실시한 결과 (그림 1)에서 라플라스 추세 검정의 결과는 라플라스 요인(Factor)이 -2와 2사이에 존재함으로서 신뢰성장(Reliability growth) 속성을 나타내고 있다. 따라서 이 자료를 이용하여 신뢰 성장모형을 제시하는 것이 효율적임을 시사하고 있다[16][17].

모수 추정은 최우추정법을 이용하고 비선형 방정식의 계산방법은 수치 해석적 기본 방법인 이분법(Bisection method)을 사용하였다. 이러한 계산은 초기값을 0.001와 1을, 허용 한계(Tolerance for width of interval)는 10^{-5} 을 주고 수렴성을 확인 하면서 충분한 반복 횟수인 100번을 C-언어를 이용하여 모수 추정을 수행하였다. 그리고 모형에 대한 효율성을 조사하기 위한 기준으로서 MSE(평균제곱오차)와 R^2(결정계수)를 추정값을 추정하였다. 결과는 <표 2>에 요약 되었다.

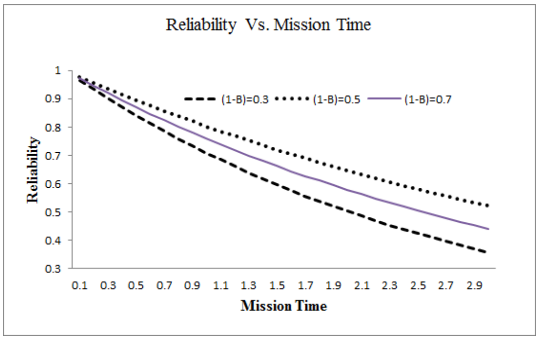


(그림 1) 라플라스 추세 검정

<표 2>에서 하나의 결함을 제거하는 동안에 새로운 결함이 발생할 확률 B을 비교한 결과 결함발생 확률이 낮은 경우가 높은 경우보다 효율적 모형으로 나타나고 있지만 B가 50%로 가정한 경우가 더 효율적 모형으로 나타나고 있고 모든 모형에서 추정치에 대한 차이를 설명하는 결정계수 측면에서는 결정계수가 0.8 이상으로 나타나 추정치의 효율성이 높게 나타나고 있다.

< 표 2> 결함발생 확률 B에 따른 MSE 와 R^2 추정값

결함 탐색률(B)	$\hat{b}_{MLE}$	$\hat{\theta}_{MLE}$	MSE	R^2
$B=0.7$	0.026415	9.180541	14.16723	0.823501
$B=0.5$	0.015854	15.30042	14.14864	0.823732
$B=0.3$	0.011322	21.42102	14.16053	0.823584



(그림 2) 새로운 결함이 발생할 확률 B을 고려한 신뢰도

그리고 (그림 2)에서는 임무시간 동안 새로운 결함이 발생할 확률 B을 고려한 신뢰도 그림이 요약되었다. 이 그림에서도 B가 30% 나 70%에 비해 50%로 가정한 경우가 보다 큰 신뢰도를 보이고 있다.

5. 결 론

일반적으로 테스팅하는 과정에서 새로운 결함이 나타나지 않는다는 가정 즉, 결함 탐색비율을 일정한 상수로 간주하여 분석하는 연구가 대부분이었다.

그러나 본 연구에서는 이 분야에서 기본 모형인 Goel- Okumoto모형의 확장 모형으로 알려진 S-모형을 적용하고 테스팅 하는 과정에서 결함이 발생할 확률을 고려한 소프트웨어 모형을 제시하였다. 수명분포는 유한고장수를 가진 비동질적인 포아송 과정에 기초한 소프트웨어 모형에 대하여 논의 되었다. 그 결과 결함이 발생할 확률이 50%로 가정한 경우가 더 효율적 모형으로 나타나고 있고 신뢰도 경우에도 다른 경우보다 높게 보여주고 있다.

따라서 테스팅 하는 과정에서 관측시간과 관련된 결함발생확률을 고려한 모형이 이 분야에서 적용 가능한 모형이 될 수 있음을 확인하였다. 경우에 따라서는 왜도와 첨도 측면에서 효율적인 카파분포, 지수화 지수분포 등 업데이트된 분포에 대한 결함이 발생할 확률을 고려한 문제를 비교 분석하는 연구도 가치 있는 일이라 판단되고 이 연구를 통하여 소프트웨어 개발자들은 고장발생확률을 사전정보로 활용하면 어느 정도 도움을 줄 수 있으리라 판단된다.

참고문헌

[1] Gokhale, S. S. and Trivedi, K. S. “A time/structure based software reliability model”, Annals of Software Engineering. 8, pp. 85-121. 1999.

[2] Goel AL, Okumoto K, “ Time-dependent

fault detection rate model for software and other performance measures", IEEE Trans Reliab 28, pp.206-11, 1978.

[3] Yamada S, Ohba H. " S-shaped software reliability modeling for software error detection", IEEE Trans Reliab, 32, pp.475-484, 1983.

[4] Zhao M. "Change-point problems in software and hardware reliability", Commun. Stat Theory Methods,22(3), pp.757-768, 1993.

[5] Shyur H-J. "A stochastic software reliability model with imperfect debugging and change-point", J Syst. Software 66, pp.135-141, 2003.

[6] Pham H, Zhang X. "NHPP software reliability and cost models with testing coverage", Eur J Oper Res, 145, pp.445-454, 2003.

[7] Huang C-Y. "Performance analysis of software reliability growth models with testing-effort and change-point". J Syst Software 76, pp. 181-194, 2005.

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

[9] 김희철, 신현철 "대수 선형 위험함수 학습효과에 근거한 NHPP 신뢰성장 소프트웨어 모형에 관한 비교 연구", 한국융합보안학회, pp. 19-26, 2012.

[10] Tao Li, Kaigui Wu."A NHPP Software Reliability Growth Model Considering Learning Process and Number of Residual Faults". Journal of Convergence Information Technology(JCIT), Volume 7, Number 13, July, pp.127-134, 2012

[11] J. F. Lawless. Statistical Models and Methods for Lifetime Data. John Wiley & Sons, New York, 1981.

[12] L. Kuo and T. Y. Yang."Bayesian Computation of Software Reliability".Journal of the American Statistical Association, Vol.91, pp. 763-773, 1996.

[13] Ye Zhang, Kaigui Wu, "Software Cost Model Considering Reliability and Time of Software in Use",Journal of Convergence Information Technology(JCIT), Volume 7, Number 13, July, pp. 135-142, 2012.

[14] R. Satya Prasad, K. R. H. Rao and R.R. L Kantha, " Software Reliability Measuring using Modified Maximum Likelihood Estimation and SPC", International Journal of Computer Applications (0975–8887) Volume 21, No.7, pp. 1-5, May, 2011

[15] M.Xie, T.N. Goh, P. Rajan, "Some effective control chart procedures for reliability monitoring; Elsevier science Ltd, Reliability Engineering and system safety 77, pp.143-150, 2002.

[16] K. Kanoun and J. C. Laprie, "Handbook of Software Reliability Engineering", M.R.Lyu, Editor, chapter Trend Analysis. McGraw-Hill New York, NY, pp. 401-437., 1996.

[17] Hee-Cheul KIM and Hyoung-Keun Park, " The Comparative Study of Software Optimal Release Time Based on Burr Distribution", International Journal of Advancements in Computing Technology, Volume 2, Number 3, pp. 119-128, 2010.

[저 자 소 개]

김 희 철(Hee-cheul Kim)

1992년 2월 동국대학교 통계학과 졸업(이학석사)
1998년 8월 동국대학교 통계학과 졸업(이학박사)
2005-현재 남서울대학교 산업경영공학과 교수

email : kim1458@nsu.ac.kr

김 경 수(Kyung-Soo Kim)

1991년 순천향대학교 전산학과 졸업(공학박사)
2005년~2006년 Virginia Common-wealth University(객원교수)
1998년~현재 백석문화대학교 인터넷정보학부 교수

email :kkskim@bscu.ac.kr