数理解析研究所講究録
第 1939 巻 2015 年 62-70

Wavelet Regression Model for Short-term Software Reliability Prediction[1]

首都大学東京・システムデザイン研究科　肖　霄
Xiao Xiao
Graduate School of System Design,
Tokyo Metropolitan University

1 Introduction

The non-homogeneous Poisson process (NHPP) has been applied successfully to model nonstationary counting phenomena for a large class of problems. In software reliability engineering, the NHPP-based software reliability models (SRMs) [4, 5, 8, 9, 12, 13, 14, 19, 20] are of a very important class. A unique parameter to govern the probabilistic properties of an NHPP is the *rate function*. Therefore, it is necessary to develop a method that can estimate the rate function of NHPP-based SRM with a high degree of accuracy, and this naturally results in trusted quantitative software reliability. According to this way of thinking, the problem of assessing the software reliability quantitatively can be reduced to the statistical estimation and prediction of the NHPP rate function.

In recent years, the wavelet-based statistical methods have been well established especially in several areas such as non-parametric regression, probability density estimation, time series analysis, *etc* (see [1, 3, 11]). Among a lot of techniques which have been proposed to account for the Poisson rate function estimation problem, Xiao and Dohi [16] proposed a non-parametric estimation framework for the NHPP-based SRMs, where the Haar-wavelet-based techniques were applied to estimate the software rate function. They treated with the software fault count (group) data, where the number of software failures is recorded. This kind of data is the observation of an NHPP when the NHPP is viewed as a counting process, and is known as an incomplete failure data since the exact detection time of each software fault is not recorded. Another type of software failure data is the so-called software failure time data, where the software failure time is observed and recorded. In this case, the NHPP is viewed as an arrival process. Kuhl and Bhairegond [6] proposed a Daubechies wavelet estimator for the NHPP rate function by considering NHPP as an arrival process. They presented simulation-based performance evaluation for their wavelet procedure, and succeeded in estimating three different types of NHPP rate functions. However, there are several mathematical difficulties when applying their procedure to the real software failure time data analysis. Their Daubechies wavelet estimator (i) is defined on compact support with length 7, but the software failure time data is observed in an arbitrary time interval, and (ii) consists of infinite summation of Daubechies scaling function, which is particularly difficult to implement in actual numerical computation. Therefore, a finite and reasonable range of the parameters included in their estimator should be determined depending on the nature of the data under consideration.

Xiao and Dohi [17] applied the Daubechies wavelet estimator to the estimation of the rate function from real software failure time data. They discussed the limitations of Kuhl and Bhairegond [6]'s work, and gave practical solutions to the technical difficulties in applying the procedure to the real software failure data. They presented a real data analysis to evaluate the goodness-of-fit performance of the Daubechies wavelet estimator, and concluded that the Daubechies wavelet estimation outperformed the existing estimation methods in most cases.

[1]This paper is an extended version of reference [18].

This paper extends Xiao and Dohi [17]'s work and aims to develop a Daubechies wavelet-based prediction method for software reliability assessment. The amazing merit of this method is that, it can provide mid-long term prediction of the software reliability measures such as the rate function and the mean value function, although it is a nonparametric estimation method which does not require prior knowledge or assumptions about the behavior of the process.

2 NHPP-based Software Reliability Modeling

Let $N(t)$ denote the number of software faults detected by testing time t, and be a stochastic point process in continuous time. We make the following assumptions:

Assumption A: Software faults occur at independent and identically distributed (i.i.d.) random times having a cumulative distribution function (c.d.f.) $F(t)$ with a probability density function (p.d.f.) $f(t) = dF(t)/dt$.

Assumption B: The initial number of software faults, N, is nonnegative and finite.

Under the above assumptions, the probability mass function (p.m.f.) of the number of software faults detected by time t is given by the binomial p.m.f.:

$$\Pr\{N(t) = n \mid N\} = \binom{N}{n} F(t)^n \overline{F}(t)^{N-n}, \quad (1)$$

where $\overline{F}(\cdot) = 1 - F(\cdot)$. If the initial number of faults N is unknown, it is appropriate to assume that N is a discrete (integer-valued) random variable. Langberg and Singpurwalla [7] proved that when the initial number of software faults N was a Poisson random variable with mean ω (> 0), the number of software faults detected before time t was given by the following non-homogeneous Poisson process (NHPP):

$$\Pr\{N(t) = n\} = \sum_{x=n}^{\infty} \Pr\{N(t) = n \mid x\} \frac{\omega^x e^{-\omega}}{x!} = \frac{\{\omega F(t)\}^n}{n!} e^{-\omega F(t)}. \quad (2)$$

Equation (2) is equivalent to the p.m.f. of the NHPP having a mean value function $\Lambda(t) = \omega F(t) = \mathrm{E}[N(t)]$, which means the expected cumulative number of software faults experienced by time t. In addition, we have

$$\Lambda(t) = \int_0^t \lambda(x)dx, \quad (3)$$

where $\lambda(t)$ is the rate function of NHPP, and implies the software failure rate at time t.

3 Daubechies Wavelet Estimator

Daubechies [2] defined a set of compactly supported wavelets, which gained much popularity in wavelet analysis. Generally, wavelets consist of two basis functions, the *scaling function* $\phi(t)$ and the *wavelet function* $\psi(t)$, that work together to provide wavelet approximations. These functions are orthonormal bases of Hilbert space, so that any signals or data in this vector space can be represented by linear combinations of scaling function and wavelet function. Since the rate function of NHPP is non-negative, we need positive orthonormal bases of a Hilbert space to approximate the rate function $\lambda(t)$ of an NHPP-based SRM. Walter and Shen [15] developed a positive wavelet estimator for estimating density functions. Let $\phi(t)$ and $\psi(t)$ be the Daubechies scaling function and wavelet function having compact support, respectively.

$$\phi(t) = \sum_{i=0}^{n} h_i \phi(2t - i), \quad (4)$$

$$\psi(t) = \sum_{i=0}^{n} (-1)^i h_{n-i} \phi(2t - i), \quad (5)$$

where n is the support of $\phi(t)$ and $\psi(t)$, and coefficients h_i ($i = 0, 1, \ldots, n$) for different supports are given in [2]. For $0 < r < 1$, a positive basis function is given by

$$P_r(t) = \sum_{j \in Z} r^{|j|} \phi(t-j), \tag{6}$$

where the constant value r is selected such that this positive basis developed is always greater than or equal to zero [15]. Figure 1 shows the positive basis function $P_r(t)$. It can be seen that $P_r(t)$ is non-negative and decays to 0 quickly. Using $P_r(t)$, a positive reproducing kernel, $k_{r,0}(t, t_i)$ in Hilbert space V_0 is constructed as follows:

$$k_{r,0}(t, t_i) = \left(\frac{1-r}{1+r}\right)^2 \sum_{n=-\infty}^{\infty} P_r(t-n) P_r(t_i - n). \tag{7}$$

Here, a kernel $k(t, t_i)$ is called a reproducing kernel if

$$\int_{-\infty}^{\infty} k(t, t_i) \times \lambda(t_i) dt_i = \lambda(t) \tag{8}$$

holds, where $\lambda(t)$ is an arbitrary function. From this reproductive property, we have the approximation of rate function $\lambda(t)$ in Hilbert space V_0, which is of the form:

$$\lambda_{r,0}(t) = \int_{-\infty}^{\infty} k_{r,0}(t, t_i) \times \lambda(t_i) dt_i. \tag{9}$$

Similarly, a positive reproducing kernel, $k_{r,m}(t, t_i)$ in Hilbert space V_m can be constructed and written as

$$k_{r,m}(t, t_i) = 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{n=-\infty}^{\infty} P_r(2^m t - n) P_r(2^m t_i - n). \tag{10}$$

Therefore, we have the approximation of rate function $\lambda(t)$ in Hilbert space V_m in the form of

$$\hat{\lambda}_{r,m}(t) = 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{n=-k}^{k} \left\{ \sum_{i=1}^{N} P_r(2^m t_i - n) \right\} \times P_r(2^m t - n), \tag{11}$$

where t_i are the arrival times of an NHPP whose rate function is to be approximated, and N is the number of arrivals in the interval under consideration. The resolution m is selected based on the level of detail of the approximation desired. This is the Daubechies wavelet estimator proposed by Kuhl and Bhairgond [6]. This wavelet estimator is used to approximate the rate function of an NHPP-based SRM.

The range for support n should be selected in such a way that the positive basis function $P_r(t)$ can translate through the entire range of arrival times. Note that the positive function $P_r(t)$ quickly decays to zero in both the positive and negative directions (see Figure 1), so we take the truncation for it from -7 to 8. The boundary is determined as -7 and 8 because the value of $P_r(t)$ outside the limit becomes negative. Walter and Shen [15] proved that there exists $0 < r < 1$ such that $P_r(t)$ satisfies $P_r(t) \leq 0$ ($t \in R$, where R is the set of all real numbers), but this holds only when parameter j in Equation (6) takes all values in Z. This is difficult in computation so that we have to select an appropriate range for parameter j. We use the determination method of reference [17], *i.e.*, parameter k in Equation (11) should be selected as [Integer Part of $2^m t_N + 7$], which ensures $2^m t_i - n$ is in the interval [-7, 8].

From Equations (4) and (5) we know, Daubechies scaling function and wavelet function are not defined in closed analytic forms. In fact, the scaling function is calculated by solving a simultaneous equation with the defined coefficients h_i and initial value $\phi(0) = \phi(n) = 0$. For example, the coefficients of Daubechies wavelet (support $n = 7$) are defined as

$h_0 = 0.3258034$, $h_1 = 1.0109457$, $h_2 = 0.8922014$, $h_3 = 0.0395750$,
$h_4 = 0.2645072$, $h_5 = 0.0436163$, $h_6 = 0.0465036$, $h_7 = 0.0149870$.

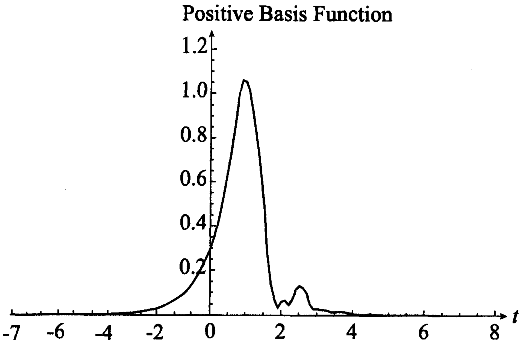


Figure 1: Positive basis function associated with Daubechies scaling function.

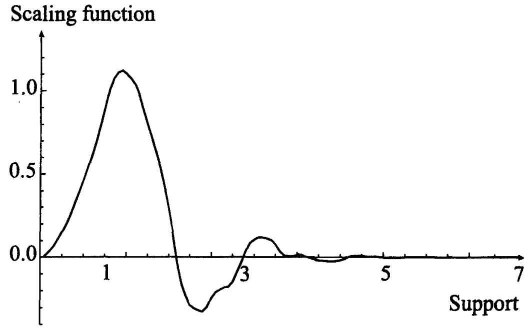


Figure 2: Daubechies scaling function with support [0, 7].

First, the starting values of Daubechies scaling function, $\phi(1),\ \phi(2),\ldots,\ \phi(6)$, can be calculated by solving

$$\begin{cases} \sum_{t=0}^{7} \phi(t) = 1, \\ \phi(1) = \sum_{i=0}^{7} h_i\phi(2-i), \\ \phi(2) = \sum_{i=0}^{7} h_i\phi(4-i), \\ \phi(3) = \sum_{i=0}^{7} h_i\phi(6-i), \\ \phi(4) = \sum_{i=0}^{7} h_i\phi(8-i), \\ \phi(5) = \sum_{i=0}^{7} h_i\phi(10-i), \\ \phi(6) = \sum_{i=0}^{7} h_i\phi(12-i), \end{cases} \tag{12}$$

where $\phi(0) = \phi(7) = 0$. Second, the values of the Daubechies scaling function at other points in time interval [0, 7] can be calculated by Equation (4) using the starting values and the coefficients h_i. For example, we have

$$\phi(0.5) \quad = \quad \sum_{i=0}^{7} h_i\phi(1-i) = h_0\phi(1) + h_1\phi(0). \tag{13}$$

A feature of Daubechies scaling function is that it only takes the value at such a time point t when $t = a{\cdot}2^b$ $(a,\ b \in Z$, where Z is the set of all integers). This kind of number is called a dyadic number if and only if, it is integral multiple of an integral power of 2 (see [10]). In other words, the Daubechies wavelet is defined in a set of discrete values. Therefore, it is classified as discrete wavelet with the same as Haar wavelet. However, if sufficient values of the Daubechies wavelet are calculated, a smooth scaling function can be obtained. This is the reason of why the Daubechies wavelet is effective in representing continuous function. An example of Daubechies scaling function with support $n = 7$, calculated in step size 0.0625, is given in Figure 2.

4 Mid-long Term Prediction using Daubechies Wavelet Estimator

The Daubechies scaling function and wavelet function have compact support. Daubechies [2] defines the coefficients h_i $(i = 0,\ 1,\ldots,\ n)$ for wavelets with different supports $n = 3,\ 5,\ 7,\ 9,\ 11,\ 13,\ 15,\ 17$ and 19. Therefore, the Daubechies wavelet estimator of the rate function $\lambda(t)$ is defined on a limited time interval. However, the real software failure time are observed in an arbatroy time interval. That is to say, the compact support is a weakness of Daubechies wavelet estimator in analyzing real world data. Therefore, a preprocessing of the data is absolutely necessary before using the Daubechies wavelet estimator to estimate the rate function of an NHPP-based SRM.

Table 1: Relation between Parameter b and Rescaled Data.

	t'_1	t'_2	...	t'_N
$b = 0$	3	33	...	88682
$b = 14$	1.831E-04	2.014E-03	...	5.413E+00
$b = 15$	9.155E-05	1.007E-03	...	2.706E+00
$b = 16$	4.578E-05	5.035E-04	...	1.353E+00
$b = 17$	2.289E-05	2.518E-04	...	6.766E-01

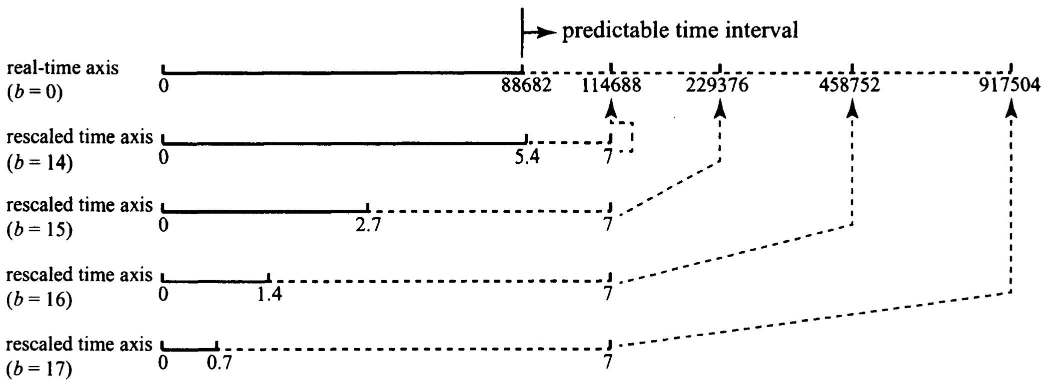


Figure 3: Rescale.

This paper makes use of this weakness in an artful way to achieve the mid-long term prediction of the software reliability measures. Suppose that a set of software failure time data t_i $(i = 1, 2, \ldots, t_N)$ is available, where t_i denotes the time of the i-th software failure, and N means the total number of failures in this data set. In other words, it is necessary to rescale the software failure time data $\{t_1, t_2, \ldots, t_N\}$ into interval $[0, n]$. Commonly, conceived idea will be that, firstly dividing the data $\{t_1, t_2, \ldots, t_N\}$ by t_N to normalize the data to $[0, 1]$, and secondly multiply n to get $[0, n]$. However, this method does not work in this case. It is clear from Equation (11) that t_i in $\lambda_{m,k}(t)$ must be a dyadic number, otherwise, $P_r(2^m t_i - n)$ can not be defined. Therefore, it is necessary to find a way, that not only ensures the rescaled data is between $[0, n]$, but also ensures that the rescaled failure time is a dyadic number. We suggest the following steps for the preprocessing:

i) If the values of the failure time data are recorded in integer, then go to the next step, else change the unit of the data set to a smaller one to obtain a set of data with interger value.

ii) Find a set of integer b that satisfies $t_N \times 2^{-b} \leq n$. Since an integer is a dyadic number and an integer divided by 2^b $(b \in Z)$ is still a dyadic number, we obtain the rescaled time data as $\{t'_1, t'_2, \ldots, t'_N\} = \{t_1, t_2, \ldots, t_N\} \times 2^{-b}$.

In this way, software failure time data with arbitrary ending time can be analyzed with the Daubechies wavelet estimator. Here, note that there exits multiple integer b. For example, consider the case with $t_1 = 3$, $t_N = 88682$, $N = 136$, and the Daubechies wavelet with support $n = 7$. The integer b that satisfies $t_N \times 2^{-b} \leq n$ are 14, 15, 16, 17,.... If we set $b = 14$, then $\{t_1, t_2, \ldots, t_N\} = \{3, 33, \ldots, 88682\}$ is rescaled to $\{1.831E-04, 2.014E-03, \ldots, 5.413E+00\}$. Table 1 shows $\{t'_1, t'_2, \ldots, t'_N\}$ when $b = 14$, 15, 16 and 17. For better understanding, we illustrate the corresponding relationship between the real-time axix and the rescaled time axis in Figure 3. It is clear from this figure that time point 7 of each rescaled time axis corresponds to different time points in the real-time axis. It corresponds to the time point 114688 when $b = 14$, while time point 917504 when $b = 17$. In other words, a larger b provides a longer predictable interval.

Table 2: Predictive Performance (PLSE).

r	b	$m = 2$	$m = 3$	$m = 4$	$m = 5$	$m = 6$	$m = 7$	$m = 8$	$m = 9$	$m = 10$
0.3	14	2.840	2.153	1.548	0.545	3.242	11.542	29.522	66.960	139.830
	15	5.492	3.991	3.027	2.181	0.785	4.549	16.145	41.219	93.560
	16	10.810	7.581	5.529	4.208	3.072	1.103	6.107	21.733	55.275
	17	-	14.883	10.453	7.643	5.819	4.891	1.710	7.657	27.699
0.4	14	3.291	2.454	1.866	1.040	1.395	6.927	19.458	46.120	100.353
	15	6.452	4.624	3.450	2.625	1.471	1.965	9.696	27.179	64.428
	16	12.927	8.890	6.396	4.787	3.666	2.066	2.618	13.032	36.476
	17	-	17.788	12.248	8.828	6.614	5.436	3.005	3.199	16.473
0.5	14	3.870	2.873	2.230	1.604	0.549	3.664	12.353	31.160	70.850
	15	7.613	5.435	4.036	3.135	2.258	0.791	5.135	17.268	43.522
	16	15.322	10.474	7.503	5.590	4.359	3.146	1.114	6.869	23.198
	17	-	21.074	14.419	10.345	7.717	6.252	4.341	1.744	8.475
0.6	14	4.752	3.615	2.895	2.352	1.520	1.114	6.883	19.805	47.888
	15	9.195	6.670	5.076	4.065	3.306	2.139	1.573	9.634	27.660
	16	18.071	12.632	9.191	7.012	5.632	4.585	2.982	2.074	12.933
	17	-	24.844	17.376	12.654	9.670	7.912	6.253	4.204	2.440
0.7	14	6.282	5.100	4.344	3.832	3.317	1.905	1.962	10.372	28.956
	15	11.529	8.815	7.156	6.094	5.381	4.657	2.676	2.754	14.482
	16	21.228	15.825	12.125	9.858	8.414	7.432	6.436	3.727	3.626
	17	-	29.176	21.751	16.667	13.577	11.661	10.153	8.811	5.511
0.8	14	8.987	7.981	7.329	6.895	6.584	5.825	3.799	1.297	11.823
	15	15.009	12.608	11.196	10.278	9.676	9.241	8.171	5.332	1.804
	16	24.856	20.599	17.321	15.394	14.153	13.323	12.727	11.235	7.428
	17	-	34.155	28.299	23.789	21.176	19.491	18.265	17.428	15.547
0.9	14	13.113	12.617	12.292	12.072	11.928	11.663	10.914	9.026	5.130
	15	19.620	18.396	17.701	17.243	16.939	16.737	16.364	15.314	12.670
	16	28.831	26.944	25.269	24.320	23.705	23.287	23.010	22.483	21.081
	17	-	39.618	37.017	34.707	33.430	32.585	31.985	31.601	30.892

5 Real Data Analysis

We apply the Daubechies wavelet estimator to the real project data set to estimate the rate function of the NHPP-based SRM. The used data set is from reference [8], where it is named as SYS1 and consists of 136 software fault data. Parameters r and m in Equation (11) can be considered as very important parameters that effect the accuracy of the estimator. If r is too small, $P_r(t)$ decays to negative value very fast. For example when $r = 0.1$ and 0.2, $P_r(t)$ provides negative values when t is greater than 2. On the other hand, m is the resolution of approximate so that the computation time becomes longer as m increases. We execute the Daubechies wavelet-based procedure by setting $r = 0.3, 0.4, \cdots, 0.9$ and $m = 1, 2, \cdots, 10$, to study the influence of these two parameters to the estimator. Moreover, the rescaled parameter b is set to be $b = 14, 15, 16$ and 17 in this paper.

We examine the prediction performance, where two prediction measures are used: predictive least square error (PLSE), and predictive log likelihood (PLL). The PLSE is defined as the least square error between the estimated intensity function and the future data from an observation point, and the PLL is the logarithm of the likelihood function with future data at an observation point. We set the observation point at $\hat{N} = 90\% * N$, and the PMSE and PLL are of the forms

$$\text{PLSE} = \frac{\sqrt{\sum_{i=\hat{N}}^{N}(\Lambda(t_i) - y_i)^2}}{N - \hat{N} + 1}, \tag{14}$$

Table 3: Predictive Performance (PLL).

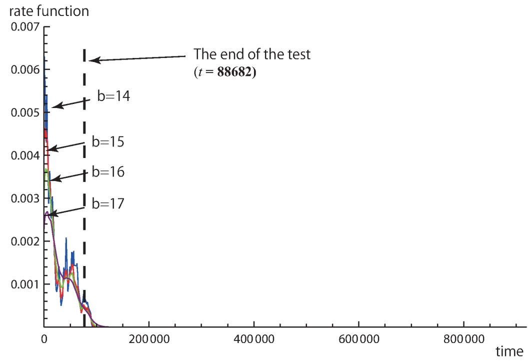
r	b	$m=2$	$m=3$	$m=4$	$m=5$	$m=6$	$m=7$	$m=8$	$m=9$	$m=10$
0.3	14	-787.818	-948.800	-539.718	-350.051	-251.931	-280.918	-392.775	-675.110	-1239.126
	15	-386.005	-453.220	-538.366	-337.502	-247.585	-210.804	-245.214	-371.713	-672.815
	16	-226.647	-253.150	-293.912	-334.449	-235.617	-182.650	-177.224	-226.424	-366.313
	17	-	-161.968	-181.228	-206.681	-231.824	-169.219	-148.646	-158.503	-218.944
0.4	14	-726.404	-844.466	-484.028	-313.835	-223.813	-235.848	-308.500	-507.338	-925.790
	15	-367.409	-420.385	-483.932	-306.197	-225.312	-188.263	-205.650	-291.315	-505.675
	16	-215.045	-240.814	-274.188	-303.902	-217.720	-169.460	-159.709	-190.197	-287.944
	17	-	-152.339	-172.510	-194.397	-214.392	-161.054	-141.230	-144.257	-185.184
0.5	14	-682.086	-768.302	-443.282	-288.329	-203.787	-203.967	-249.098	-386.972	-691.616
	15	-351.649	-395.794	-443.476	-283.413	-209.429	-172.096	-177.598	-234.626	-385.806
	16	-203.769	-229.659	-258.700	-281.121	-204.180	-159.854	-147.041	-164.411	-232.652
	17	-	-142.229	-163.993	-184.187	-200.700	-154.113	-135.453	-134.151	-160.912
0.6	14	-648.619	-709.669	-410.581	-267.406	-186.945	-178.656	-203.499	-295.686	-509.355
	15	-336.505	-375.570	-410.723	-263.810	-195.397	-158.013	-155.096	-191.008	-294.924
	16	-192.244	-217.808	-244.576	-261.286	-191.210	-150.351	-135.452	-143.579	-189.954
	17	-	-131.019	-154.025	-173.664	-187.407	-145.906	-128.543	-124.520	-141.264
0.7	14	-622.520	-662.616	-380.635	-245.578	-168.033	-153.439	-162.591	-220.003	-359.075
	15	-320.734	-356.489	-380.964	-242.653	-178.183	-140.915	-131.876	-151.491	-219.576
	16	-180.394	-203.477	-228.675	-240.192	-174.499	-136.245	-120.017	-121.576	-150.958
	17	-	-118.386	-140.564	-159.747	-170.698	-132.515	-116.114	-110.368	-120.378
0.8	14	-603.739	-624.763	-348.912	-216.724	-140.257	-120.423	-117.115	-146.812	-223.023
	15	-304.867	-336.247	-350.350	-214.666	-151.372	-113.852	-99.988	-106.827	-146.643
	16	-169.232	-185.460	-207.562	-213.218	-148.439	-110.982	-93.763	-90.364	-106.552
	17	-	-104.611	-121.495	-138.398	-145.607	-108.171	-91.630	-84.625	-90.051
0.9	14	-610.029	-610.501	-320.695	-181.111	-100.775	-75.220	-61.095	-66.330	-88.792
	15	-299.529	-322.658	-324.674	-180.822	-113.121	-73.070	-54.467	-50.808	-66.305
	16	-164.991	-169.058	-184.173	-181.817	-111.867	-71.529	-52.375	-44.640	-50.704
	17	-	-93.311	-99.291	-109.908	-111.444	-70.489	-51.520	-42.806	-44.674

$$\text{PLL} = \sum_{i=\hat{N}+1}^{N} y_i \ln[\lambda(t_i)] - \Lambda(t_N), \tag{15}$$

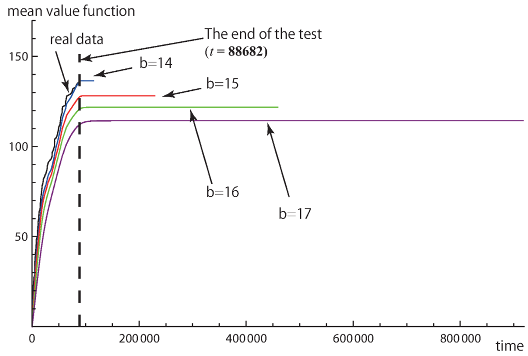
respectively, where $\lambda(t_i)$ are the Daubechies estimates, and y_i are the number faults detected by time t_i.

Table 2 and Table 3 present the prediction results at observation point $\hat{N}$ of a whole data set, and show the PLSE and PLL for $m = 2, 3, \cdots, 10$ and $b = 14, 15, 16, 17$. Here, the influence of the rescale parameter b is focused. It can be seen that, when approximation resolution m is fixed, larger value of parameter b provides smaller PLSE, while smaller b provides larger PLL. Note that, PLSE measures the physical distance between the estimate and the observation, PLL measures the preciseness of the estimate in a statistical sense. The result of PLL indicates that it is possible to find an appropriate rescale parameter b by tuning its magnitude to a larger one. Therefore, it is necessary to investigate the predictive performance with larger values of b in the future.

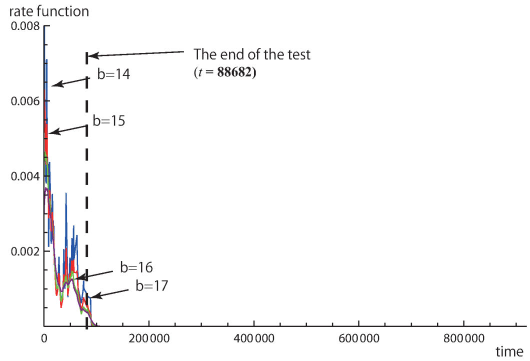
Figure 4 shows the Daubechies wavelet estimates of rate function and the mean value function with different settings of rescale parameter b. From (i) and (iii) of this figure we can see that, it tends to underestimate the failure rate function, when changing rescale parameter b from 14 to 17. This trend can also be found in (ii) and (iv). Furthermore, we show the end of the testing time in Figure 4. Taking look at (iii) and (iv), it is clearly to see that the predictable interval of $b = 17$ is longer than $b = 14$. This also motivates us to keep on studying the effect of rescale parameter b in future work.



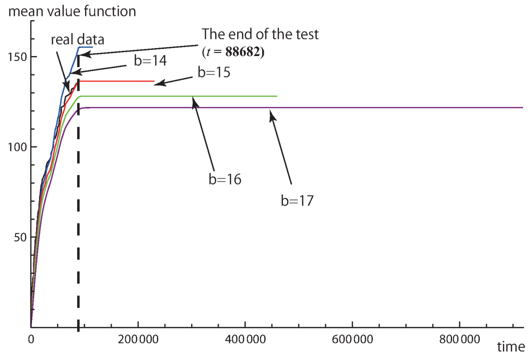
(i) rate function (m = 4).



(ii) mean value function (m = 4).



(iii) rate function (m = 5).



(iv) mean value function (m = 5).

Figure 4: Behavior of Predicted Reliability Measures (r = 0.3).

6 Conclusion

This paper has applied the Daubechies wavelet estimator to predict the rate function of NHPP-based SRM. Real data analysis has been presented to evaluate the predictive performance of the Daubechies wavelet estimator. We have given practical solutions to the technical difficulties in applying the procedure to the real software failure data. Throughout the numerical evaluation, we have established the credibility and the usefulness of the Daubechies wavelet estimation procedure in software failure data analysis. The prediction ability of this estimator can be improved by investigating the influence of rescale parameter b. Such studies will be made in subsequent work.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 26730039.

References

[1] A. Antoniadis, J. Bigot and T. Sapatinas, "Wavelet estimators in nonparametric regression: a comparative simulation study," *Journal of Statistical Software*, **6** (6), pp. 1–83 (2001).

[2] I. Daubechies, *Ten Lectures on Wavelets*, SIAM, Pennsylvania (1992).

[3] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Picard, "Density estimation by wavelet thresholding," *Annals of Statistics*, **24** (2), pp. 508-539 (1996).

[4] A. L. Goel, "Software reliability models: assumptions, limitations and applicability", *IEEE Transactions on Software Engineering*, **SE-11** (12), pp. 1411–1423 (1985).

[5] A. L. Goel and K. Okumoto, "Time-dependent error-detection rate model for software reliability and other performance measures," *IEEE Transactions on Reliability*, **R-28**, pp. 206–211 (1979).

[6] M. E. Kuhl and P. S. Bhairgond "Nonparametric estimation of nonhomogeneous Poisson processes using wavelets," *Proceedings of the 2000 Winter Simulation Conference*, pp. 562–571 (2000).

[7] N. Langberg and N. D. Singpurwalla, "Unification of some software reliability models", *SIAM J. Sci. Comput.*, **6**, pp. 781–790, 1985.

[8] M. R. Lyu (ed.), *Handbook of Software Reliability Engineering*, McGraw-Hill, New York (1996).

[9] J. D. Musa and K. Okumoto, "A logarithmic Poisson execution time model for software reliability measurement," *Proceedings of the 7th International Conference on Software Engineering* (ICSE' 84), pp. 230–238 (1984).

[10] Y. Nievergelt, *Wavelet Made Easy*, Birkhauser, Boston (1999).

[11] D. B. Percival and A. T. Walden, *Wavelet Methods for Time Series Analysis*, Cambridge University Press (2000).

[12] H. Pham, *Software Reliability*, Springer, Singapore (2000).

[13] H. Pham and X. Zhang, "NHPP software reliability and cost models with testing coverage," *European Journal of Operational Research*, **145**, pp. 443–454 (2003).

[14] N. D. Singpurwalla and S. P. Wilson, *Statistical Methods in Software Engineering: Reliability and Risk*, Springer-Verlag, New York (1999).

[15] G. G. Walter and X. Shen, "Positive Estimation with Wavelets," *Contemporary Mathematics*, **216**, pp. 63–79 (1998).

[16] X. Xiao and T. Dohi, "Wavelet-based approach for estimating software reliability," *Proceedings of 20th International Symposium on Software Reliability Engineering* (ISSRE'09), pp. 11–20 (2009).

[17] X. Xiao and T. Dohi, "Software failure time data analysis via wavelet-based approach," *IEICE Transactions on Fundamentals*, vol. E95-A, no. 9, pp. 1490–1497 (2012).

[18] X. Xiao, "Daubechies wavelet-based software reliability prediction," *IEICE Technical Report*, vol. 114, no. 256, pp. 7–12 (2014).

[19] M. Xie, *Software Reliability Modelling*, World Scientific, Singapore (1999).

[20] S. Yamada, M. Ohba and S. Osaki, "S-shaped reliability growth modeling for software error detection," *IEEE Transactions on Reliability*, **R-32**, pp. 475–478 (1983).