Tool Development of Daubechies Wavelet Estimation for Stochastic Counting Processes

Jingchi Wu, Tadashi Dohi*, and Hiroyuki Okamura

1-4-1 Kagamiyama, Higashi-hiroshima, Hiroshima, Japan
d220580@hiroshima-u.ac.jp, dohi@hiroshima-u.ac.jp, okamu@hiroshima-u.ac.jp
*corresponding author

Abstract—Daubechies wavelets proposed by Ingrid Daubechies are a family of orthogonal wavelets and are used for frequency analysis, multi-resolution analysis, statistical feature value analysis, etc. Kuhl and Bhairgond (2000) applied the Daubechies wavelet to estimate stochastic counting processes with time-varying intensity functions. In this paper, we develop a Daubechies wavelet estimation tool for the stochastic counting processes; Daube-WET, as a web-based free application. A case study is given to illustrate how to use Daube-WET with real failure time data of a repairable system.

Keywords–Tool development; Daubechies wavelet; statistical estimation; stochastic point processes; web-based free application.

1. Introduction

Daubechies wavelets proposed by Ingrid Daubechies [3] are a family of orthogonal wavelets and used for frequency analysis, multi-resolution analysis, statistical feature value analysis, etc. in a variety of research areas including speech recognition, character recognition, image analysis. Recently, the wavelet-based approach has also received much attention for statistical inference of stochastic processes. Especially, the discrete Haar wavelets [8] have been used for alternative non-parametric estimation for stochastic counting processes with time-varying intensity functions. Donoho and Johnstone [4], [5], [7], Donoho et al. [6], Kolaczyk [9], Nason [13] developed the discrete-Haar wavelets shrinkage estimation methods with several kinds of denoising and thresholding rules to estimate non-stationary Poisson processes.

The above statistical inference approaches are categorized into non-parametric estimation for the non-stationary Poisson processes without the complete knowledge of intensity functions, where the underlying point process data are the group data, which are event-occurrence data during time intervals. Xiao and Dohi [16], [17] applied the above discrete Haar wavelets shrinkage estimation methods to analyze the software fault count data observed in the testing phases, and established non-parametric inference schemes for software reliability growth models [12]. Recently, Wu et al. [19] developed W-SRAT; a Wavelet-based software reliability assessment tool by implementing the existing Haar wavelets shrinkage estimation methods and a novel prediction algorithm for the future.

On the other hand, when the event-occurrence time data for non-stationary Poisson processes are available, the Haar wavelets shrinkage estimation methods do not work because they are regarded as the complete data but not the incomplete data as the group data. In fact, such data have been observed in the lifetime data for repairable systems [1], [2]. More specifically, when a system fails, the minimal repair is made instantaneously, where the system's function is recovered without changing its age. Under the minimal repair assumption, the cumulative number of failures/repairs is described by a non-stationary Poisson process. There are already well-known estimation tools such as CASRE [11] and SRATS [14] available for practitioners or researchers to use. However, these tools are based on the parametric assumptions of the non-stationary Poisson process. In other work, it is rare to know the parametric form of the non-stationary Poisson process in advance. In this scenario, statistical estimation tools based on non-parametric estimation methods are useful. Wavelet-based non-parametric estimation methods have demonstrated excellent estimation performance in many fields. Kuhl and Bhairgond [10] developed a Daubechies wavelet estimator and applied it to estimate the non-stationary Poisson process, where they dealt with a periodic intensity function to represent the cyclic behavior of event occurrence. Xiao and Dohi [15] applied the Kuhl and Bhairgond estimator [10] based on the Daubechies wavelet to software fault count data, and investigated applicability to software reliability estimation.

In this way, Daubechies wavelets have been known as a useful tool to estimate the stochastic counting processes and their associated codes are open in StackOverflow and MATLAB & Simulink. To our best knowledge, unfortunately, the Kuhl and Bhairgond estimator [10] has not been implemented yet. In addition, we point out that their estimator is regarded as an approximation of the well-known naive estimator, so it is possible to revisit the Kuhl and Bhairgond estimator [10]. In this paper, we develop a Daubechies wavelet estimation tool for the stochastic counting processes; Daube-WET, as a web-based free application.

The remaining part of this paper is organized as follows. In Section 2, we introduce the Daubechies wavelets. Section 3 concerns the non-parametric estimation for non-stationary Poisson processes, where the naive estimator and the Daubechies wavelet estimators are summarized. In Section 4, we overview Daube-WET from the viewpoints of the architecture and functionality. Section 5 is devoted to a case

2767-6684/23/$31.00 ©2023 IEEE
DOI 10.1109/DSA59317.2023.00123

study with real failure time data of a repairable system. Finally, the paper is concluded in Section 6.

2. Daubechies Wavelets

2.1. Definition

Daubechies [3] proposed a set of continuous and compactly supported wavelets, which are very popular in the wavelet analysis field. The Daubechies wavelets are not defined in closed form, where the Daubechies scaling function and wavelet function are defined in the following forms;

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

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

In Eqs.(1) and (2), the filter coefficients h_i are given in the reference [3], and n is the support width and determines the smoothness of the functions $\phi(t)$ and $\psi(t)$. The starting values $\{\phi(t),\ t = 1, 2, \ldots, n-1\}$ can be obtained by solving the recursive formula;

$$\begin{cases} \sum_{t=0}^{n} \phi(t) = 1, \\ \phi(0) = 0, \\ \phi(n) = 0, \\ \phi(t) = \sum_{i=0}^{n} h_i \phi(2t - i), \quad t = 1, 2, \ldots, n-1. \end{cases}$$

The other values of $\phi(t)$ with $t \in [0, n]$ and $t \neq 1, 2, \ldots, n-1$ are calculated in Eq. (1).

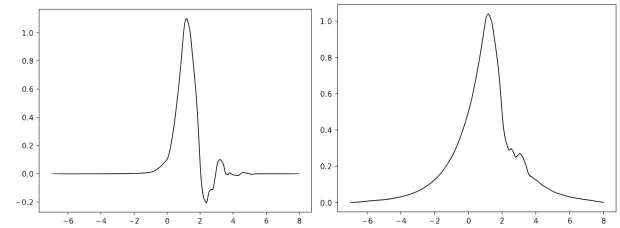
2.2. Wavelet Approximation

Dissimilar to discrete Haar wavelets [8], Daubechies wavelets [3] have smooth and continuous basis functions except at $n = 1$ in Eqs.(1) and (2). So, the Daubechies wavelets can approximate an arbitrary function such as an intensity function of a non-stationary Poisson process, $\lambda(t)$, which is assumed to be an absolutely continuous function. Since the Daubechies scaling function $\phi(t)$ in Eq.(1) can take negative values, one needs a positive basis function for representing a positive $\lambda(t)$. Walter and Shen [18] developed a positive basis function for estimating probability density functions.

Let $\phi(t)$ be the Daubechies scaling function having compact support. The positive basis function by Walter and Shen [18] is given by

$$P_r(t) = \sum_{j \in \mathbb{Z}} r^{\|j\|} \phi(t - j) \quad (3)$$

with parameter r satisfying $a \leq r < 1$, where $\mathbb{Z}$ is a set of all integers. The parameter r controls the minimum value of $P_r(t)$, so that the minimum value of $P_r(t)$ is greater than or equal to 0 when $r = a$ (>0). Figure 1 illustrates the positive basis functions with $r = 0.1$ and $r = 0.5$ for $\phi(t)$ with $n = 7$. In this case, we can see that the minimum value of $P_r(t)$ is less than 0 when $r = 0.1$.



(a) $n = 7$ and $r = 0.1$. (b) $n = 7$ and $r = 0.5$.

Figure 1: Positive basis functions.

By using the positive basis function, $P_r(t)$, a positive reproducing kernel, $k_r(t, s) \in V_0$, is given by

$$k_r(t, s) = \left(\frac{1 - r}{1 + r}\right)^2 \sum_{a=-\infty}^{\infty} P_r(t - a) P_r(s - a). \quad (4)$$

Let $k(t, s)$ denote a reproducing kernel satisfying

$$\int_{-\infty}^{\infty} k(t, s) \lambda(s) ds = \lambda(t), \quad (5)$$

where $\lambda(t)$ is an arbitrary continuous function. For $\lambda(t) \in L_2(\mathbb{R})$, an approximation of the function $\lambda_0(t) \in V_0$ is constructed as

$$\lambda_0(t) \approx \int_{-\infty}^{\infty} k_r(t, s) \lambda(s) ds. \quad (6)$$

In general, the approximation of an arbitrary function $\lambda(t) \in V_m$ is given by

$$\lambda_m(t) = \int_{-\infty}^{\infty} k_{r,m}(t, s) \lambda(s) ds, \quad (7)$$

where $k_{r,m}(t, s)$ in Eq.(7) is the positive reproducing kernel in V_m and is given by

$$k_{r,m}(t, s) = 2^m k_r(2^m t, 2^m s). \quad (8)$$

3. Daubechies Wavelet-based Estimation

3.1. Non-stationary Poisson Process

Let $\{N(t), t \geq 0\}$ be a stochastic counting process in the discrete integer space $N(t) = 0, 1, 2, \ldots$. The stochastic process $N(t)$ is said a non-stationary Poisson process if the following conditions hold.

- $N(0) = 0$,
- $\{N(t), t \geq 0\}$ has independent increment,
- $\Pr\{N(t + \Delta t) - N(t) \geq 2\} = o(\Delta t)$,
- $\Pr\{N(t + \Delta t) - N(t) = 1\} = \lambda(t; \boldsymbol{\theta}) \Delta t + o(\Delta t)$,

where the function $\lambda(t; \boldsymbol{\theta})$ is an absolutely continuous function, called the intensity function, $\boldsymbol{\theta}$ is the model parameter (vector), and $o(\Delta t)$ is the higher-order term of the infinitesimal time Δt, satisfying $\lim_{\Delta t \to 0} o(\Delta t)/\Delta t = 0$. From a few algebraic manipulations, it is not so difficult to obtain the probability mass function;

$$\Pr\{N(t) = n\} = \frac{\Lambda(t; \boldsymbol{\theta})^n}{n!} e^{-\Lambda(t; \boldsymbol{\theta})}, \quad (9)$$

where

$$\Lambda(t;\boldsymbol{\theta}) = \mathrm{E}[N(t)] = \int_0^t \lambda(u;\boldsymbol{\theta})du \tag{10}$$

is the mean value function with the parameter vector $\boldsymbol{\theta}$.

3.2. Non-parametric Estimation

Suppose that the mean value function $\Lambda(t;\boldsymbol{\theta})$ and the intensity function $\lambda(t;\boldsymbol{\theta})$ are known, but the parameters $\boldsymbol{\theta}$ are unknown. Then we can estimate a few representative statistical estimation methods. The commonly used technique to estimate the parameter $\boldsymbol{\theta}$ in $\Lambda(t;\boldsymbol{\theta})$ and $\lambda(t;\boldsymbol{\theta})$ is the maximum likelihood estimation. Suppose that n event-occurrence times $\boldsymbol{t} = (t_1, t_2, \ldots, t_n)$ with right truncation at $T\ (\geq t_n)$ are available. Then, the log-likelihood function for the time-domain data is given by

$$LLF(\boldsymbol{\theta};\boldsymbol{t}) = \sum_{i=1}^{n} \lambda(t_i;\boldsymbol{\theta}) - \Lambda(T;\boldsymbol{\theta}). \tag{11}$$

By maximizing $LLF(\boldsymbol{\theta};\boldsymbol{t})$ with respect to $\boldsymbol{\theta}$, we get the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$.

Next, we consider the case where the intensity function $\lambda(t;\boldsymbol{\theta}) = \lambda(t)$ is completely unknown. The most intuitive but the simplest method to estimate the intensity function is a piecewise-linear interpolation, which is called the naive (natural) estimate. For the n event-occurrence time data, $t_i\ (i = 1, 2, \cdots, n)$ with the right-truncation $T\ (\geq t_n)$, define

$$\hat{\lambda}(t) = \begin{cases} \frac{1}{t_i - t_{i-1}}, & t_0 \leq t \leq t_i;\ i = 1, \cdots, n, t_0 = 0, \\ \frac{1}{T - t_n}, & t_n \leq t \leq T. \end{cases} \tag{12}$$

Then we have the following step-function estimate with breakpoints t_i;

$$\begin{aligned} \hat{\Lambda}(t) &= \int_0^t \hat{\lambda}(x)dx \\ &= \begin{cases} i + \frac{t - t_i}{t_{i+1} - t_i}, & t_i \leq t \leq t_{i+1};\ i = 0, 1, \cdots, n-1, \\ n + \frac{0.5(t - t_n)}{T - t_n}, & t_n \leq t \leq T. \end{cases} \end{aligned} \tag{13}$$

The resulting naive estimate of the mean value function $\hat{\Lambda}(t)$ in Eq.(13) is obtained by plotting n event-occurrence time points and connecting them by line segments.

When only one sample path, $t_i\ (i = 0, 1, 2, \ldots, n)$, is available in a single non-stationary Poisson process, the naive estimate seems to be the straightforward but the most natural estimate of the cumulative number of events, because the mean squares error between the native estimate and the underlying time-domain data is always zero. However, it should be noted that the above the estimate of the intensity function is discontinuous everywhere and tends to fluctuate with big noise.

3.3. Wavelet-based Estimator

Kuhl and Bhairgond [10] proposed a wavelet estimate based on Eq. (7);

$$\begin{aligned} \hat{\lambda}_{r,m}(t;\boldsymbol{t}) =& 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{a=-k}^{k} \left\{ \sum_{i=1}^{n} P_r(2^m t_i - a) \right\} \\ & \times P_r(2^m t - a), \end{aligned} \tag{14}$$

where $t_i\ (i = 1, 2, \ldots, n)$ are the time-domain data, the parameter a is determined in the range in which the positive basis function covers the entire time, and the resolution level m is determined based on the detail of the approximation. Unfortunately, Kuhl and Bhairgond [10] did not clarify the derivation procedure of their wavelet estimate in Eq.(14). Here, we derive the same result to complete the discussion, and improve Kuhl and Bhairgond estimate [10].

Let $\lambda_{naive}(t)$ denote the naive estimate of intensity function for a non-stationary Poisson process when $T = t_n$;

$$\hat{\lambda}_{naive}(t) = \sum_{i=1}^{n} \frac{1}{t_i - t_{i-1}} I_i(t), \tag{15}$$

where

$$I_i(t) = \begin{cases} 1, & t_{i-1} < t \leq t_i \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

Substituting the naive estimate $\lambda(t)$ in Eq. (15) into Eq.(7), we can obtain the naive wavelet estimate (NWE);

$$\begin{aligned} \hat{\lambda}_{NWE}(t;\boldsymbol{t}) &= \int_{-\infty}^{\infty} k_{r,m}(t,s)\hat{\lambda}_{naive}(s)ds \\ &= \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} k_{r,m}(t,s) \frac{1}{t_i - t_{i-1}} ds \\ &= 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{i=1}^{n} \int_{t_{i-1}}^{t_i} \sum_{a=-\infty}^{\infty} P_r(2^m t - a) \\ &\quad \times Pr(2^m s - a) \frac{1}{t_i - t_{i-1}} ds \\ &= 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{a=-\infty}^{\infty} \Big\{ \sum_{i=1}^{n} \frac{1}{t_i - t_{i-1}} \int_{t_{i-1}}^{t_i} \\ &\quad \times Pr(2^m s - a)ds \Big\} P_r(2^m t - a). \end{aligned} \tag{17}$$

Hence it is seen that the exact form of NWE contains the integral parts. If each integral is approximated by an elementary rectangular approximation method, which is given by

$$\hat{\lambda}_{RNWE}(t;\boldsymbol{t}) = 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{a=-\infty}^{\infty} \left\{ \sum_{i=1}^{n} \frac{1}{t_i - t_{i-1}} \int_{t_{i-1}}^{t_i} \right.$$
$$\left. \times Pr(2^m t_i - a) ds \right\} P_r(2^m t - a)$$
$$= 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{a=-\infty}^{\infty} \left\{ \sum_{i=1}^{n} \frac{1}{t_i - t_{i-1}} \right.$$
$$\left. \times (t_i - t_{i-1}) Pr(2^m t_i - a) \right\} P_r(2^m t - a)$$
$$= 2^m \left(\frac{1-r}{1+r}\right)^2 \sum_{a=-\infty}^{\infty} \left\{ \sum_{i=1}^{N} P_r(2^m t_i - a) \right\}$$
$$\times P_r(2^m t - a). \tag{18}$$

We call the above estimate the rectangular approximate naive wavelet estimate (RNWE), which is equivalent to Kuhl and Bhairgond estimate [10]. Strictly speaking, RNWE is an approximation of NWE from the computational point of view. In order to compute $\hat{\lambda}_{NWE}(t)$ more accurately, we need to apply any numerical integration algorithm for Eq.(17).

4. Tool Development

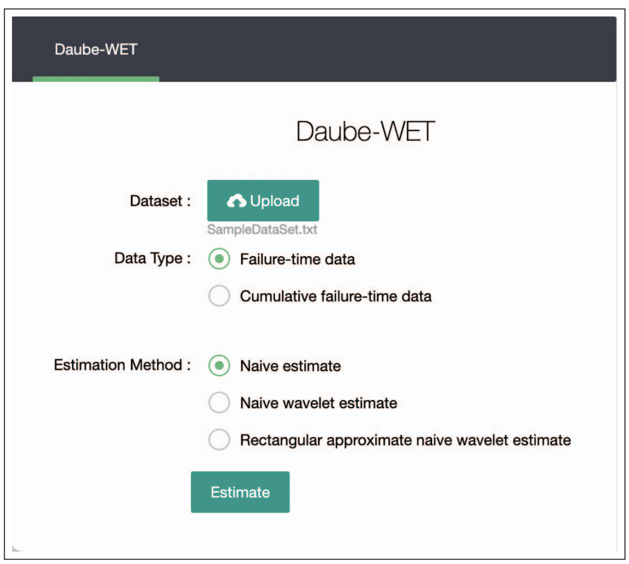


Figure 2: Interface of Daube-WET (screenshot).

4.1. Daube-WET

We develop a Daubechies wavelet estimation tool for the stochastic counting processes, Daube-WET [1], to estimate the non-stationary Poisson process with unknown intensity function. In Figure 2, we show the interface of Daube-WET. The Daube-WET is a web-based freeware without cumbersome installation and complicated deployment, and is a unique solution to support the wavelet-based estimation which is applicable to not only the non-stationary Poisson process estimation but also some signal processing problems. It runs in a cloud computing environment and does not depend on the kind of operating system.

[1]https://DaubeWET.wujingchi.com

4.2. System Architecture

The system process of Daube-WET is depicted in Figure 3. It consists of two software components; the user interface (front end) written by HTML, CSS, and JavaScript, the work end (back end) written by PHP and Python language. The front-end consists of two web pages; the main page and the estimation page.

The main page contains the following functions:

- `Upload dataset` - Users can upload the time-domain data using this function. The Daube-WET accepts text files in ".txt format". A sample data file with the correct format can be downloaded with the upload button. Detailed data format specifications are provided later. Before uploading one data set, users need to select the data type first, either event-occurrence time or event-occurrence time interval. Subsequently, users can click on the upload button to upload the data file. The uploaded event-occurrence time data will be sent to the cloud server for data format validation. When the data format is correct, it will be temporarily stored on the cloud server and the server will then inform the user total number of events and event-occurrence time length for the given dataset (see Figure 4 (a)), which serves as verification that the server has correctly read the data. If the data format is incorrect, the user will receive an error message and the line number in the text file where an error occurred (see Figure 4 (b)). If the data format is correct but some of its characteristics do not match the data type selected by the user, the user will receive a warning message (see Figure 4 (c)).
- `Select estimation method` - Users can choose a method for non-parametric estimation. The Daube-WET provides three estimation methods; naive estimation, naive wavelet estimation, and rectangular approximation naive wavelet estimation (see Figure 5 (a)). When the wavelet estimation or rectangular approximation wavelet estimation is selected, additional estimation parameters are needed (see Figure 5 (b)), so users need to input additional parameters.
- `Input estimation parameters` - Users can input additional estimation parameters using this function. When the wavelet estimation or rectangular approximation wavelet estimation is selected, users must input the tuning parameter and the resolution level parameter. Selecting the most appropriate tuning parameters and resolution level parameters requires a certain amount of knowledge about wavelets. However, since many users may not be well-versed in wavelet methods, we have given a guideline for recommended parameter values based on preliminary experimental results (see Figure 5 (b)).
- `Verify user input` - The Daube-WET is a web-based tool that allows public access. Hence, it may be intruded by some malicious attacks, or misused by poor skilled users who input some unexpected parameter values. The Daube-WET can check automatically the estimation parameters

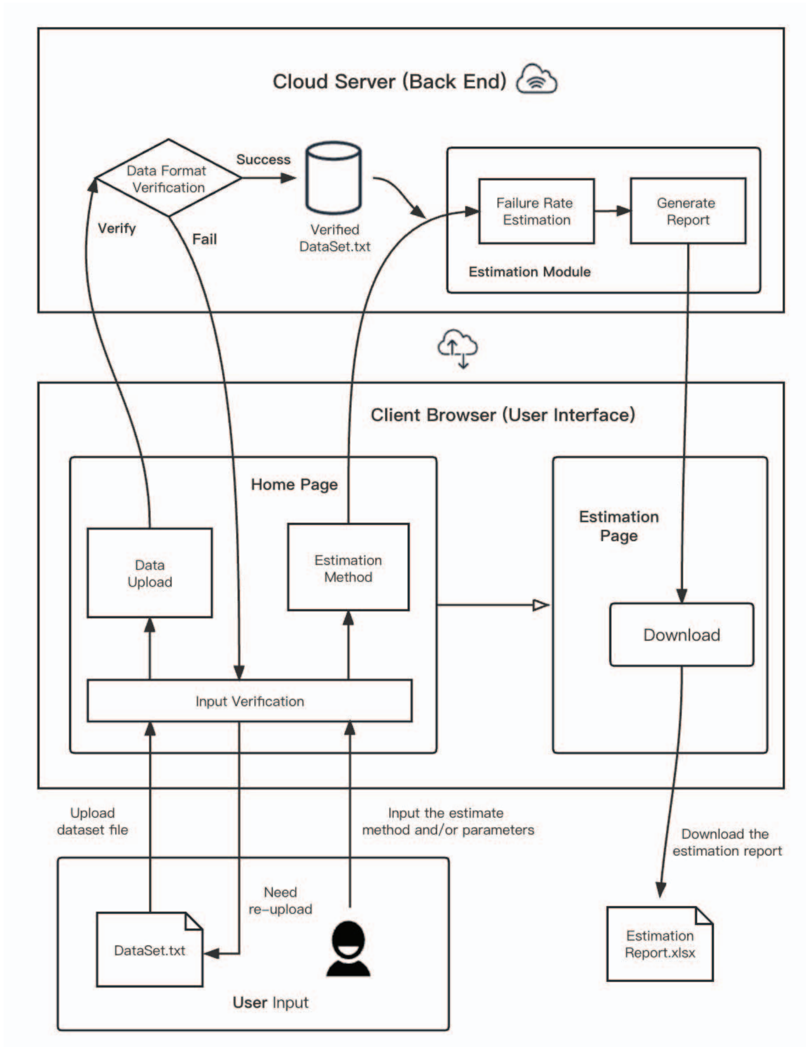


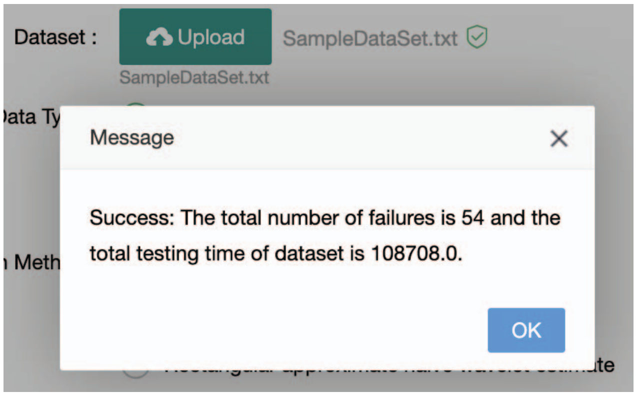
Figure 3: System process of Daube-WET.

input by the user. When the estimation parameters entered are not as expected, the user receives a corresponding warning message (see Figure 6).

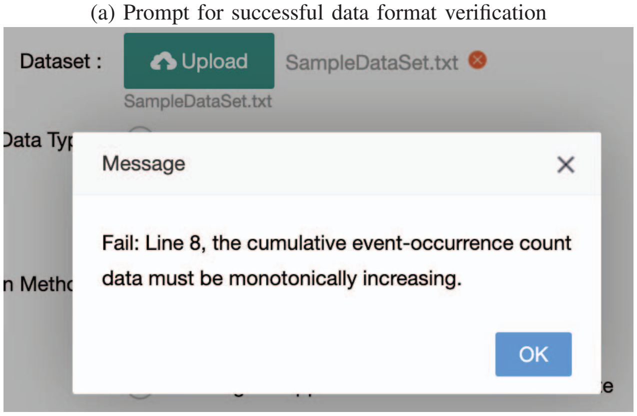
Once the user has uploaded the correct data file and selected the estimation method with the estimation parameters, he/she needs to click on the "Estimate" button on the main page to open the estimation page (see Figure 7). The estimate page has only two functions; confirmation of the options selected on the main page and issue of an estimation report. On the estimation page, the user can confirm the selected data type, the uploaded data file, the selected estimation method, and the estimation parameters. When the user confirms that the above information was correct, he/she can click the "Estimate" button on the page to start the estimation procedure. When the estimate is completed, the original "Estimate" button will change to "Download" button. The user can download the estimation report issued by the Daube-WET after clicking the "Download" button.

In the back end, we process the data sent from the front end, estimate the intensity function, and output the estimation report in ".xlsx form". The following two modules are used in the back end.

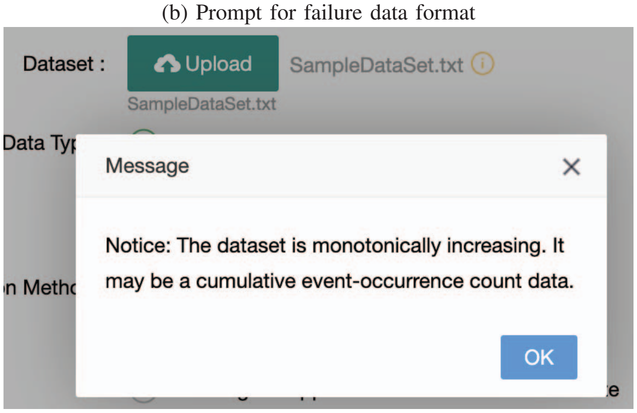
- `Verification Module` - The validation module accepts data files uploaded by users and validates them. The details of the validation are presented in the description of the "Upload Dataset" function on the main page. In the Daube-WET, the specific validation code is written in Python, and the PHP code is responsible for handling the HTTP request and calling the corresponding Python program.
- `Estimation Module` - The estimation module estimates the intensity function based on the verified data, the estimation method, and parameters input by the user. As with the validation module, the estimation algorithm and



(a) Prompt for successful data format verification



(b) Prompt for failure data format



(c) Prompt for warning

Figure 4: Data format validation prompt (screenshot).

the report issue are written in Python. PHP is responsible for handling the HTTP requests and calling the estimation program written in Python. The estimation report is saved in ".xlsx format". The report contains the following information; the original time-domain data, the intensity function (estimated value), an estimate of the cumulative number of events, information related to this estimate, an image of the intensity function, and an image of expected the cumulative number of events.

4.3. Data Format

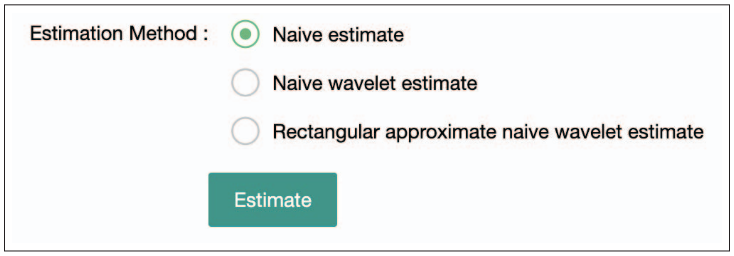
In Daube-WET, we deal with the time-domain data and the time interval data in text files in ".txt format". Let $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ be the time interval data between the $(i-1)$-st and i-th event occurrences, and $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ be the cumulative time data, where $y_i = \sum_{j=1}^{i} x_j$. In the Daube-WET, it can handle either of both kinds of data; $\boldsymbol{x}$ or $\boldsymbol{y}$. The first line indicates the number of events. The second row indicates the time interval between consecutive event occurrences. Figure 8 presents the data format for $\boldsymbol{x}$, where the first column denotes the number of events, and the second column does the time interval between consecutive two event occurrences. Once the data set was uploaded after selecting the data type, the format is checked with a verification function to ensure whether the data format is correct or not.

4.4. Functionality

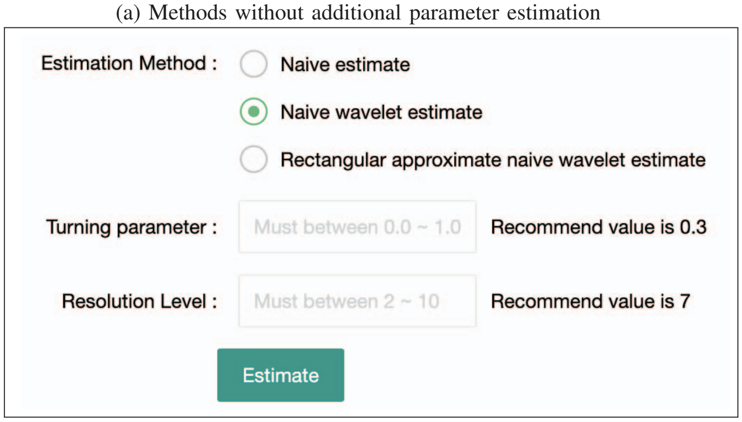
In Daube-WET, three estimation methods are prepared; naive estimation, naive wavelet estimation, and rectangular approximation naive wavelet estimation. The naive estimation is the fastest and least resource-intensive among the three methods. It generates an estimation report without additional input parameters. However, it is known that the intensity function obtained from the naive estimation tends to overfit. The rectangular approximation naive wavelet estimation is also relatively fast but slower than the naive estimation. This is due to the large number of recursive calculations required to obtain the values of the Daubechies scaling functions. Using this method yields a smoother and continuous intensity function compared to the naive estimation. However, because it is an approximation of the naive wavelet estimation, the estimated intensity function is less accurate, i.e., it shows a higher mean absolute error (MAE).

The naive wavelet estimation can provide a relatively more accurate intensity function but requires appropriate values for tuning parameter and resolution level parameter. It also needs a much longer computation time than both naive estimation and rectangular approximation naive wavelet estimation. For example, when using the sample data file provided on the Daube-WET website, it takes about 5 minutes to complete the computation. The computation time increases with the number of event occurrences. Table 1 describes the characteristics of the three methods on computational efficiency. The more detailed performance of the three methods will be shown in the next section.

Whichever estimation method in the Daube-WET is selected by the user, an estimation report in the same content is issued for further analysis by the user; the original time-domain data, the intensity function (estimate value) and its graph, the mean value function (estimated from the intensity function) and its graph, statistical information related to these estimates. Regarding the original time-domain data, it simply outputs the data file that the Daube-WET can read. As for the estimated intensity function and the mean value function, we need to make the following clarifications. Since the three estimation methods provided by the Daube-WET are all non-parameter



(a) Methods without additional parameter estimation



(b) Methods with additional parameter estimation

Figure 5: The estimation methods provided by Daube-WET (screenshot).

TABLE I: Features of different estimation methods in Daube-WET

Daube-WET			
Method Name	Naive Estimate	Rectangular approximation Naive Wavelet Estimate	Naive Wavelet Estimate
Computation time	Very short	Short	Very long
Memory Usage	Very low	High	High
MAE	Very low (may over-fitting)	High	Low
Additional parameters	Not need	Need	Need

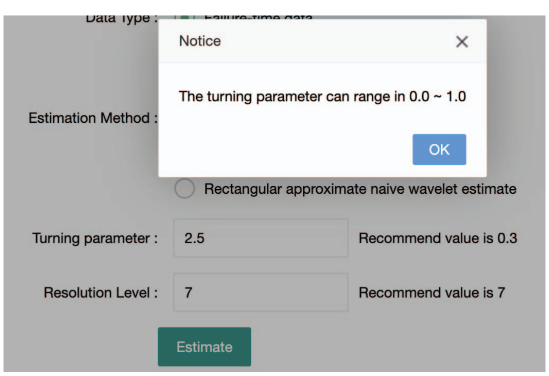


Figure 6: User input error prompt (screenshot).

estimation methods, we cannot provide some parameters as parametric methods. We can only provide estimated intensity values at pre-specified times. In particular, for the naive wavelet estimation and the rectangular approximation naive wavelet estimation, we calculate estimated intensity values at 100 equally spaced time points between $t_0 = 0$ and t_n, as well as at each event-occurrence time t_i $(i = 1, 2, \ldots, n)$. Similarly, from the same reason, we cannot obtain the mean value function at an arbitrary time by connecting the discrete estimates of the intensity function and calculating the area between the line connecting the points and the coordinate axis as an approximation of the integral of the intensity function. For the naive estimation, since it is discontinuous at the right margin of event-occurrence time, we cannot only seek the intensity values at 100 equally spaced time points and each event-occurrence time. We need to additionally estimate the intensity function at the right margin of each event-occurrence time.

Finally, the mean absolute error (MAE) of the estimated mean

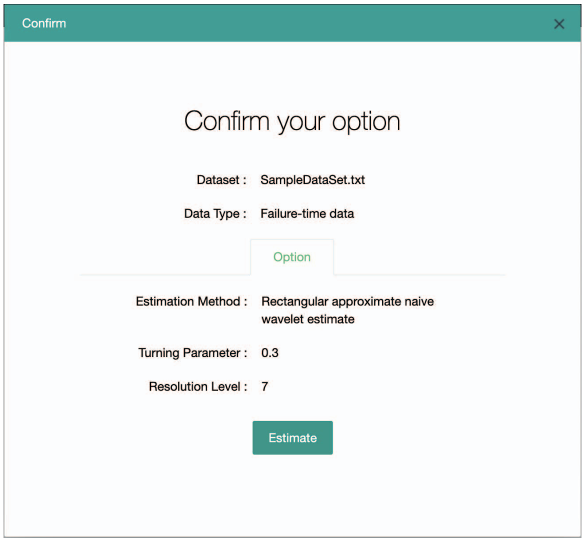


Figure 7: Interface of estimation page (screenshot).

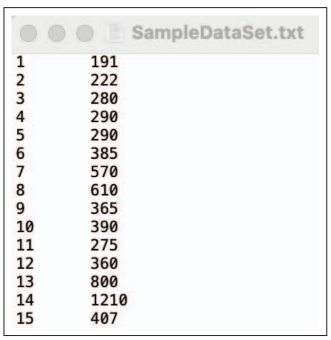


Figure 8: Sample data file formatted (screenshot).

value function is given in all estimation reports, where

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} \|i - \tilde{\Lambda}(y_i)\|}{n}, \qquad (19)$$

n is the number of events, y_i is the i-th event-occurrence time, and $\tilde{\Lambda}(\cdot)$ is an estimate of the mean value function. For users who choose the naive wavelet estimation or rectangular approximation naive wavelet estimation, the estimation report shows the values of the tuning parameter and resolution level parameter used in the estimation, as well as the Daubechies wavelets employed for the analysis. The kind of Daubechies wavelet is fixed to $db4$, which is an alias for the Daubechies wavelets with a support width of 7 [3], and cannot be changed at this time.

5. A Case Study

In this section, we demonstrate how to use the Daube-WET through an illustrative example. The sample dataset used in our case study can be downloaded from the Daube-WET official website. It consists of software fault-detection time data. It is observed in an actual software development project and is referred to as S2 in [12]. This data set consists of 54 time-domain data during 108708 CPU seconds. To use the cumulative failure time data, we simply select "Cumulative Failure Time Data" in the "Data Type" field before uploading the data. After uploading the data set, we select the estimation method. In our case study, we chose the "Naive wavelet estimation" and use the recommended parameter values (the tuning parameter is set to 0.3 and the resolution level parameter is set to 7).
Next, when one clicks the "Estimate" button, the "Estimate Page" appears. If we confirm that the options on the "Estimate Page" are correct, click the "Estimate" button and wait for the completion of calculations by the server. Once the server finished the calculation, click the "Download" button to issue the estimation report.
Even with the same dataset and the same method, the computation time required for each estimation method still varies due to network fluctuations and server-available resources. However, a benchmark for the computation time is useful for users to check a rough computational effort when using the Daube-WET. We used a sample dataset provided on the official website and conducted each 10 experiments with three methods, where the average time consumption of the 10 experiments represents the computation time overhead of the three methods. The results are shown in Table II. Computation time (total) refers to as average total time spent from clicking the "Estimate" button to the appearance of the "Download" button, including the time required for network transmission. Computation time (actual) refers to as the actual running time of the estimation program on the server side. MAE refers to as average MAE results in the 10 experiments.
Figure 9 shows the original software fault-detection time data, the intensity function (estimate value), and the mean value function (estimated from the intensity function). Figure 10 clarifies the method used for estimation, the kind of Daubechies wavelet, the values of the tuning, and resolution level parameters. Figure 11 depicts the behavior of the intensity function and mean value function.

6. Conclusions

In this paper we have developed a Daubechies wavelet estimation tool for the stochastic counting processes; Daube-WET, as a web-based free application, which contained the naive estimation method and two kinds of Daubechies wavelet estimation methods to estimate the intensity function of non-stationary Poisson process with the time-domain data. As an advantage of the Daube-WET, it can run in a cloud computing environment and is user-friendly, so the user can estimate the intensity function via the official Daube-WET website but does not need to configure any environment locally. The Daube-WET has a strict verification function for the user's input, which minimizes the human errors caused by the users. Since the naive wavelet estimation method requires a large number of recursive operations to calculate the Daubechies scaling functions and numerical integrations to calculate the reproducing kernel function, it usually takes more than 10

Number of Event-occurrence (Data)	Event-occurrence Time (Data)	Time	Intensity Function(Estimate)	Mean Value Function(Estimate)
1	191	0.00001	0.002561895	1.28095E-08
2	413	191	0.004057895	0.632189936
3	693	413	0.00396282	1.522489361
4	983	693	0.003560362	2.575734838
5	1273	983	0.003391939	3.58381848
6	1658	1087.08001	0.003316664	3.932934218
7	2228	1273	0.002981559	4.518416946
8	2838	1658	0.002185171	5.513012411
9	3203	2174.16001	0.00171705	6.520097647
10	3593	2228	0.001701466	6.612124072
11	3868	2838	0.002187849	7.798364914
12	4228	3203	0.002626375	8.676960737
13	5028	3261.24001	0.002636626	8.830219357
14	6238	3593	0.00304389	9.772503268
15	6645	3868	0.003047481	10.61006668
16	6695	4228	0.002011395	11.52066431
17	7355	4348.32001	0.001614265	11.73878405
18	8862	5028	0.001034874	12.63906753
19	9487	5435.40001	0.000835786	13.02012095
20	10399	6238	0.001763585	14.06324835
21	11037	6522.48001	0.003276744	14.78018472
22	11330	6645	0.004303793	15.2445684
23	12542	6695	0.004453521	15.46350128
24	13154	7355	0.00112003	17.30277341
25	13829	7609.56001	0.000737994	17.53926281
26	15044	8696.64001	0.000810173	18.38075384
27	17759	8862	0.001130054	18.54117179
28	21310	9487	0.001334576	19.31136861
29	22110	9783.72001	0.001126785	19.67653614

Figure 9: Estimation report 1 (screenshot).

Estimation Report	
Estimation Method	Naive wavelet estimate
Daubechies Wavelet	db4
Turning Parameter	0.3
Resolution Level	7
Mean Absolute Error	1.181

Figure 10: Estimation report 2 (screenshot).

TABLE II: Estimation performance under a sample dataset

Method Name	Naive Estimate	Rectangular approximation Naive Wavelet Estimate	Naive Wavelet Estimate
Computation time (actual)	133 ms	485 ms	342132 ms
Computation time (total)	4621 ms	5022 ms	346850 ms
MAE	0.222	13.81	1.181

minutes to complete the calculation, when using the naive wavelet estimation method.

The estimation report issued by the Daube-WET is available in an Excel sheet, which is easily utilized as the analysis report. The estimation report contains four main elements; original time-domain data, estimation values of intensity function, goodness-of-fit measure, and the graph of the estimation values.

In the future, we will improve the computational performance of the Daube-WET for the naive wavelet estimation method. We will also propose another non-parametric estimation method via Daubechies wavelets, replacing the naive estimation by the kernel estimation.

7. Acknowledgements

This work was supported by JST, the establishment of university fellowships towards the creation of science technology innovation, Grant Number JPMJFS2129.

References

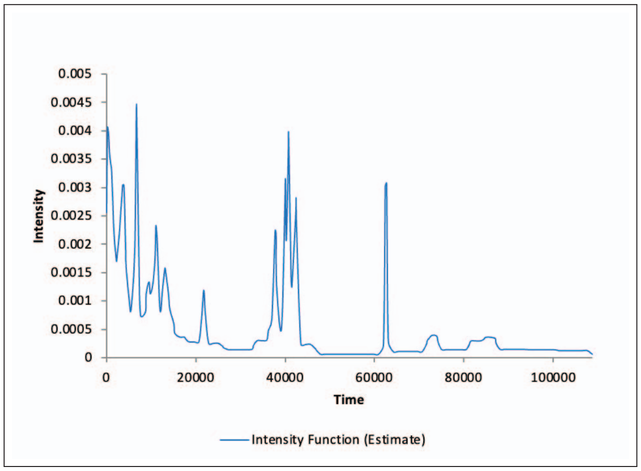
[1] H. Ascher and H. Feingold, *Repairable Systems Reliability*, Marcel Dekker, New York, 1984.

[2] R. E. Barlow and F. Proschan, *Statistical Theory of Reliability and Life Testing*, Holt, Rinehart and Winston, New York, 1975.

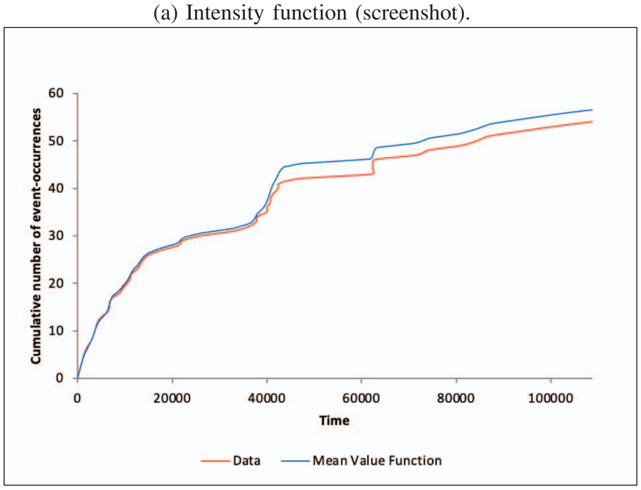
[3] I. Daubechies, Ten Lectures on Wavelets, *Society for industrial and applied mathematics*, 1992.

[4] D. L. Donoho and I. M. Johnstone, Ideal spatial adaptation by wavelet shrinkage, *Biometrika*, **81** (3), 1994, 425—455.

[5] D. L. Donoho and I. M. Johnstone, Adapting to unknown smoothness via wavelet shrinkage, *Journal of the American Statistical Association*, **90**, 1995, 1200-–1224.



(a) Intensity function (screenshot).



(b) mean value function (screenshot).

Figure 11: Estimation report 3.

[6] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian and D. Ricard, Wavelet shrinkage: asymptopia? (with discussion), *Journal of the Royal Statistical Society: Series B*, **57**, 11995, 301-–337.

[7] D. L. Donoho and I. M. Johnstone, Minimax estimation via wavelet shrinkage, *Annals of Statistics,* **26** (3), 1998, 879-–921.

[8] A. Haar, *Zur Theorie der Orthogonalen Funktionensysteme*, Georg-August-Universitat, Gottingen, 1909.

[9] E. D. Kolaczyk, Wavelet shrinkage estimation of certain Poisson intensity signals using corrected thresholding, *Statistica Sinica*, **9** (1), 1999, 119-–135.

[10] M. E. Kuhl and P. S. Bhairgond, Nonparametric estimation of nonhomogeneous Poisson processes using wavelets, *2000 Winter Simulation Conference Proceedings*, **1**, 2000, 562–571.

[11] M. R. Lyu and A. Nikora, "CASRE: a computer-aided software reliability estimation tool," *Proceedings of the Fifth International Workshop on Computer-Aided Software Engineering*, 264-275, IEEE CPS, 1992.

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

[13] G. P. Nason, Wavelet shrinkage using cross-validation, *Journal of the Royal Statistical Society: Series B*, **58**, 1996, 463—479.

[14] H. Okamura and T. Dohi, "SRATS: Software reliability assessment tool on spreadsheet (Experience report)," *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)*, 100-107, IEEE CPS, 2013.

[15] X. Xiao and T. Dohi, Software failure time data analysis via wavelet-based approach, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, **95** (9), 2012, 1490–1497.

[16] X. Xiao and T. Dohi, Wavelet shrinkage estimation for NHPP-based software reliability models, *IEEE Transactions on Reliability*, **62** (1), 2013, 211–225.

[17] X. Xiao and T. Dohi, Wavelet shrinkage estimation for NHPP-based software reliability models, *IEEE Transactions on Reliability*, **62** (1), 2013, 211–225.

[18] G. G. Walter and X. Shen, Positive estimation with wavelets, *Contemporary Mathematics*, **216**, 1998, 63–80.

[19] J. Wu, T. Dohi and H. Okamura, W-SRAT: Wavelet-based software reliability assessment tool, *Proceedings of the 21st IEEE International Conference on Software Quality, Reliability, and Security (QRS 2021)*, 564–573, IEEE CPS, 2021.