IIE Transactions (2007) **39**, 303–315

ISSN: 0740-817X print / 1545-8830 online
DOI: 10.1080/07408170600847168

Failure event prediction using the Cox proportional hazard model driven by frequent failure signatures

ZHIGUO LI[1], SHIYU ZHOU[1,*], SURESH CHOUBEY[2] and CRISPIAN SIEVENPIPER[2]

[1] *Department of Industrial and Systems Engineering, University of Wisconsin, Madison, WI 53706, USA*
E-mail: szhou@engr.wisc.edu
[2] *GE HealthCare, Global Service Technology, Waukesha, WI 53186, USA*

Received August 2005 and accepted January 2006

The analysis of event sequence data that contains system failures is becoming increasingly important in the design of service and maintenance policies. This paper presents a systematic methodology to construct a statistical prediction model for failure event based on event sequence data. First, frequent failure signatures, defined as a group of events/errors that repeatedly occur together, are identified automatically from the event sequence by use of an efficient algorithm. Then, the Cox proportional hazard model, that is extensively used in biomedical survival analysis, is used to provide a statistically rigorous prediction of system failures based on the time-to-failure data extracted from the event sequences. The identified failure signatures are used to select significant covariates for the Cox model, i.e., only the events and/or event combinations in the signatures are treated as explanatory variables in the Cox model fitting. By combining the failure signature and Cox model approaches the proposed method can effectively handle the situation of a long event sequence and a large number of event types in the sequence. Its effectiveness is illustrated by a numerical study and analysis of real-world data. The proposed method can help proactively diagnose machine faults with a sufficient lead time before actual system failures to allow preventive maintenance to be scheduled thereby reducing the downtime costs.

Keywords: Cox proportional hazard model, discrete event sequence, failure event prediction, failure signatures

1. Introduction

The method of servicing equipment (e.g., medical equipment, photocopy machines and computer hardware) is moving from reactive firefighting to preventive (proactive) maintenance. The reactive servicing of equipment is expensive and results in equipment downtime which negatively affects customer satisfaction and customer profitability. Therefore, current emphasis is being placed on predicting machine faults with a sufficient lead time before actual failure to allow a preventive repair action to be scheduled.

Error/event logs and system performance data can be used to determine preventive maintenance cycles that allow downtime to be avoided. The prediction of machine failure requires a formal framework to specify causal links between failure modes and *failure indicators* (*failure signatures*). These indicators can be generated from the error and event sequences, i.e., a series of events marked with their occurrence times, logged in the system's log files.

For example, a system error/event log file for a Computerized Tomography (CT) machine can consist of several thousand records associated with several hundred different event types and their associated occurrence times during machine usage. The recorded events can be related to various machine activities and behaviors, system failures, operator/user actions, or status of a subsystem task, etc. In practice, people use event sequence data to manually identify failure signatures within a time frame, which is specified by area experts based on experience and the physical operation principles of the system. Clearly, this is a time-consuming and labor intensive method.

A simple case of such an event sequence is illustrated in Fig. 1. In this figure, A, B, C and K are the different event types that occur at various points along the time line. Hereafter we let K represent the key failure event we are interested in, and in most cases event K occurs recurrently in the event sequence as shown in Fig. 1.

The event sequence contains considerable system information which can be used to monitor and diagnose faults in the process, or predict the future behavior of the process, say, the occurrence of some event(s) of interest. For instance, by analyzing the log file of a CT imaging system, service engineers can identify a frequently occurring failure signature (event sequence segment) consisting of five events with the last event representing the scan hardware error. The last event in this signature is a *failure event*, whereas the first four events contained in the signature are called *trigger events*.

*Corresponding author

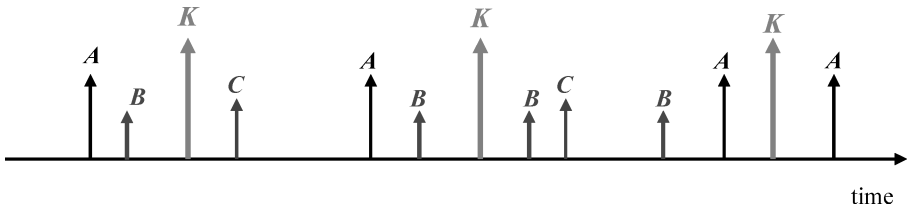

Fig. 1. An example of a timed event sequence.

Knowledge of this failure signature allows the identification of the root cause of a system failure, and thus creates the potential for opportunistic maintenance, for example, part replacement, etc. On the other hand, if the occurrence of a failure could be predicted based on the trigger events, then preventative maintenance measures could be taken before the system breakdown and thus the downtime cost will be reduced.

In this paper, we are interested in building a statistical failure prediction model for a single event sequence based on failure signatures. Formally, an *event sequence* **S** is a triple $(T_s^{\mathbf{s}}, T_e^{\mathbf{s}}, \mathbf{s})$ on a set of event types **E**, where $T_s^{\mathbf{s}}$ and $T_e^{\mathbf{s}}$ are the starting time and ending time respectively, and $\mathbf{s} = \langle (E_1, t_1^{\mathbf{s}}), (E_2, t_2^{\mathbf{s}}), \ldots, (E_m, t_m^{\mathbf{s}}) \rangle$ is an ordered sequence of events such that $E_i \in \mathbf{E}$ for all $i = 1, 2, \ldots, m$ and the individual $t_i^{\mathbf{s}}$ are the occurrence time of the corresponding event with $T_s^{\mathbf{s}} \leq t_1^{\mathbf{s}} \leq \cdots \leq t_m^{\mathbf{s}} \leq T_e^{\mathbf{s}}$ (Mannila *et al.*, 1997). The problem of building a failure prediction model is formulated as follows: given the event sequence **S** containing failure event K, how do we construct a statistical model that can predict the occurrence of system failure K, i.e., during what time interval and with what probability will the failure event K occur in the system?

Some techniques to predict failure event(s) based on the analysis of event sequence data already exist. These methods can be roughly classified into design-based methods and data-driven rule-based methods. Design-based methods tend to be applied to logic fault diagnosis in automated manufacturing systems. In a design-based method, the expected event sequence is obtained from the system design and is compared with the observed event sequence. A system logic failure can be identified by use of this comparison. Sampath *et al.* (1994) and Chen and Provan (1997) proposed untimed and timed *automata* models to diagnose the faults in an automated systems. Untimed and timed *Petri net* models were developed by Valette *et al.* (1989) and Srinivasan and Jafari (1993) to represent the behavior of manufacturing systems and determine if a fault occurs. *Time template* models (Holloway and Chand, 1994; Holloway, 1996; Das and Holloway, 1996; Pandalai and Holloway, 2000) make use of timing and sequencing relationships of events, which are generated from either timed automata models (system design) or observations of manufacturing systems, to establish when events are expected to occur. The construction of all the abovementioned models requires us to know the designed or expected event sequences of the system. The major disadvantage of this method is that in many cases, the event occurring is random and thus there is no predefined system design information and hence no temporal relationship knowledge available.

In contrast with design-based methods, data-driven rule-based methods do not require system logic design information. Instead, they first identify the temporal patterns, i.e., the sequences of events that frequently occur, and then prediction rules are developed based on these patterns. Mannila *et al.* (1997) analyzed the event sequence data by identifying frequently occurring *episodes* (temporal patterns) through the "WINEPI" approach, in which computationally efficient algorithms are developed to identify frequent episodes and episode rules. In Klemettinen (1999), a method for recurrent pattern identification in alarm data for a telecommunications network was proposed to recognize episode rules. The technique of sequential pattern detection has also been applied to web log files by Agrawal (1996) and Xiao and Dunham (2001). Once the temporal patterns are identified, the time relationships among events in the pattern can be used to predict the occurrence of a failure event. To reach this goal, prediction rules, such as *temporal association rules* (Dunham, 2003) and *episode rules* (Mannila *et al.*, 1997; Klemettinen, 1999), can be generated based on the identified temporal patterns. An example of a prediction rule based on a temporal pattern consisting of events A, B and K is:

IF the events A and B occur in the system
THEN the failure event K will occur
WITH [Time Interval] confidence (c%)

which means that if we observe events A and B occurring in the system, then we can predict that failure event K will occur within the *time interval* specified by [Time Interval] with a *confidence* of c%. If we try to predict the occurrence of a failure event, the prediction process begins by searching through the space of prediction rules generated from the identified temporal patterns. The available data-driven rule-based methods do not build rigorous statistical prediction models for event sequence data and thus they only provide heuristic prediction results. We would encounter the following two difficulties when using these rules for prediction.

1. Once temporal patterns are identified, the corresponding prediction rules are fixed with their parameters, i.e., the values of [Time Interval] and confidence (c%) are fixed

in the above prediction rule. If people are interested in a different time interval, new temporal patterns need to be identified in terms of the changed parameters. If we need to predict the occurrence of events of interest with varying parameters, the space of prediction rules could be very large for a long event sequence.

2. The prediction becomes more complicated, if not impossible, for the case in which different trigger event sets, say, $\mathbf{T}_{r1}$ and $\mathbf{T}_{r2}$, occur in the system. Now we have different rules based on different trigger event sets, therefore we will have different prediction results. It is hard for us to combine all the associated prediction rules together to reach a final conclusion.

In this paper, we would like to develop a systematic methodology to construct a rigorous prediction model for failure events based on a single event sequence collected from in-service equipment. At the first step, we will isolate the meaningful failure signatures, *which are a special temporal pattern*, *namely*, *a set of events that occur together frequently in the event sequence and end with the failure event*, and then screen out trigger events which could affect the occurrence of failure events. Next, the *Cox proportional hazard model* (Klein and Moeschberger, 2003) will be built to provide rigorous statistical predictions for the system failures based on the identified failure signatures. In the procedure, we take advantage of both *temporal pattern identification* techniques originating from temporal data mining and *the Cox PH model* that predominates in biomedical survival analysis. Our approach is data-driven, which means that we do not need detailed physical models for the relationship between the trigger event(s) and the failure event. Another advantage of our approach is that no assumption of a parametric distribution for the event sequence data is needed, which could result in the discovery of information that may be hidden by the assumption of a specific distribution.

The remainder of this paper is organized as follows. In Section 2, the problem formulation and the data-driven procedure to construct the prediction model are presented. We illustrate the effectiveness of the developed procedure through a numerical case study and a real-world example in Section 3. Finally, conclusions are drawn in Section 4.

2. Failure event prediction using the Cox proportional hazard model

2.1. *Basics of failure prediction using the Cox proportional hazard model*

As stated in the Introduction, we will consider an event sequence in which the failure event K occurs recurrently along the time line. Suppose event K occurred n times in the event sequence; hereafter we will call the interval $[t_i^s, t_{i+1}^s]$ between two adjacent failure events, K_i and K_{i+1}, $i = 1, 2, \ldots, (n-1)$ the *Time Interval Between Failures* (TIBF) as illustrated in Fig. 2. It should be noted that if an event K does not occur at T_s^s or T_e^s, the start and end points of the event sequence; the interval $[T_s^s, t_1^s]$ or $[t_n^s, T_e^s]$, is also the TIBF. In this case, there are $N = n + 1$ time intervals in the event sequence data in total. In many cases, it is reasonable if we assume that all the time intervals are independent of one another, that is, the current occurrence of failure event K is assumed to be unaffected by any previous occurrence of an event K. In Fig. 2 the symbol "□" represents the occurrence of an event K, and the symbol "o" means that the last TIBF is censored for the case in which no failure event K occurs at the end of the event sequence. If applicable, the censoring time of the last TIBF is assumed to be independent of the failure times in the event sequence.

Based on the above assumptions, the data we now have are the *time-to-failure data* of event K, also referred as *survival data*. Let T denote the time intervals, i.e., T is a random variable that indicates the waiting time from the start of the current TIBF to the next failure. As stated above, all the T_i ($i = 1, 2, \ldots, n + 1$) in the event sequence are assumed to be independent of one another.

Some basic quantities are needed in order to analyze the time-to-failure data. If the density function of T, $f(t)$, exists, then the *survival function* of T can be written as

$$S(t) = \Pr(T > t) = \int_t^{+\infty} f(x)\mathrm{d}x, \tag{1}$$

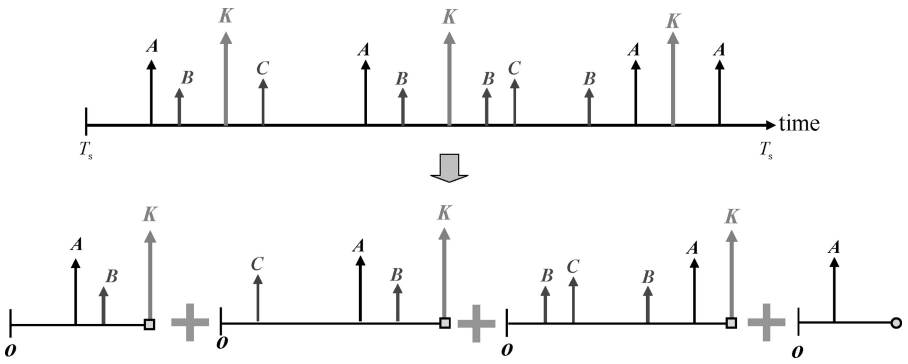

Fig. 2. An example of independent time intervals between failures in an event sequence.

which can be interpreted as the probability that the length of the TIBF is larger than a specified value t. Another basic quantity used is the *hazard function*, also called the *conditional failure rate function* in the reliability literature, which can be written as,

$$h(t) = \lim_{\Delta t \to 0} \frac{\Pr(t \le T < t + \Delta t \mid T \ge t)}{\Delta t} = \frac{f(t)}{S(t)}. \quad (2)$$

We can interpret the hazard function as the "instantaneous" probability that the failure event occurs at time t, given that no failure event occurs before t. Thus, $h(t)\Delta t$ can be viewed as the "approximate" probability that a failure event will occur in the small time interval between t and $t + \Delta t$. This quantity is very useful in describing the chance of experiencing a failure event. It is particularly useful in reliability studies because it can help determine the correct failure distributions (Klein and Moeschberger, 2003). Various models have been built for the hazard rate function, for example, multiplicative hazard rate models in survival analysis and the bathtub hazard model in reliability (Misra, 1992). If $h(t)$ is known, we can calculate the function $S(t)$ using the following equation:

$$S(t) = \exp\left[-\int_0^t h(x)\mathrm{d}x \right] \quad (3)$$

Based on $S(t)$, we can predict that failure event K can occur in a specified time interval $[t_i, t_j]$ with a probability of $S(t_i) - S(t_j)$. If needed, other quantities, such as the *mean time between failures* in reliability, can also be derived.

Among the abovementioned multiplicative hazard rate models developed to analyze biomedical survival data, the Cox Proportional Hazard Model (Cox, 1972) is a particularly powerful regression model. In clinical trials, for example, the Cox model is used to investigate how some covariates affect the hazard rate and survival of patients who have been given a kidney transplant. Time-to-death data for these patients are analyzed and the covariates examined include the sex and age of the patients (Klein and Moeschberger, 2003). Let $h[t|\mathbf{Z}(t)]$ be the hazard rate at time t for a TIBF with covariate vector $\mathbf{Z}(t)$; the basic Cox model is as follows (Klein and Moeschberger, 2003):

$$h[t \mid \mathbf{Z}(t)] = h_0(t)\exp[\boldsymbol{\beta}^T\mathbf{Z}(t)] = h_0(t)\exp\left[\sum_{k=1}^{y}\beta_k Z_k(t)\right], \quad (4)$$

where $h_0(t)$ is the baseline hazard rate function. The use of a proportional hazards model means that the hazard rate of a subject is proportional to its baseline hazard rate $h_0(t)$, which is the basic assumption of Cox's model. In the model, $\boldsymbol{\beta}$ is the coefficient vector and $\mathbf{Z}(t) = [Z_1(t), Z_2(t), \ldots, Z_y(t)]^T$ is the covariate vector. $Z_i(t)$, $i = 1, 2, \ldots, y$, is a *time-dependent* covariate if its value varies with time. The Cox model can be used to build a predictive model for a failure event based on trigger events. For example, in the first TIBF shown in Fig. 2, since at the beginning of this TIBF we do not know whether or not the intermediate event A will occur, the intermediate event A can be coded as a time-dependent covariate as follows:

$$Z_A(t) = \begin{cases} 0, & o \le t < \text{the occurence time of } A \\ 1, & \text{the occurence time of } A \le t \le \\ & \text{the end of the TIBF.} \end{cases} \quad (5)$$

If the value of a covariate is known at the start of each TIBF, it is called a *fixed-time covariate*, and in this case we would denote it as Z_i.

As indicated previously, the objective of this study is to build a statistical prediction model for a *single* event sequence, and thus some extensions of Cox model, such as the Prentice-Williams-Peterson (Prentice *et al.*, 1981) and Wei-Lin-Weissfeld (Wei *et al.*, 1989) models can not be used in this case, because they were proposed to handle recurrent event data of multiple subjects. In addition, multi-state models have some disadvantages for recurrent events data because they consider states rather than events (Hougaard, 2000). Another popular regression model used in survival analysis and reliability analysis (Wasserman, 2003) is the *accelerated failure time model*, also referred to as the *parametric regression model*, but its usage is restricted by the distributions people can assume for the time-to-failure data, i.e., we have to select an "appropriate" parametric distribution for the failure data when fitting the model, such as exponential, Weibull, Gamma distributions, etc. (Klein and Moeschberger, 2003). Compared with this model, the Cox proportional hazard model is a semi-parametric model in that it does not need to assume and thus defend any distribution for the hazard rate, which will benefit us because assuming hazard rate functions for the field data could hide some useful information although it could provide some conveniences (George, 2003). Another advantage of Cox's model is that studying interactions between variables is easy (Elsayed and Chan, 1990; Hougaard, 1999).

Now if we have a simple event sequence as shown in Fig. 1, we can predict the occurrence of failure event K by fitting the Cox model to the failure data with the events A, B and C as time-dependent covariates. However, there are two difficulties in practice if we want to apply this simple prediction model building technique.

1. In many cases, the field failure data, i.e., the failure event sequence data, may contain numerous event types, say, several hundred different event types in a log file, thus it will be quite difficult, if not impossible, to incorporate all the event types as covariates in the regression model. That is, it is hard for us to select the statistically significant covariates and interactions among these covariates from a large set of event types. For this reason, it is necessary to select the trigger events which may affect the occurrence of failure event.
2. Statistically significant covariates or interactions could be insignificant even meaningless from a physical point

of view. As stated in the Introduction, failure signatures play a very important role in service and maintenance. Thus, it is reasonable that people would predict the occurrence of failure events based on identified failure signatures. That is, in the first step, people would identify the failure signatures from both statistical and physical points of view. Next, the failure prediction model will be built based on the failure signatures. If the model is constructed in a reversed sequence, the results may be meaningless or even misleading.

To deal with the abovementioned difficulties when fitting the Cox model directly to the failure data, an effective technique for failure prediction model construction driven by failure signatures is developed.

2.2. *Steps in failure prediction model building*

A diagram of the complete procedure for failure prediction model building is illustrated in Fig. 3.

Firstly, the failure event to be predicted is defined by experts in the specific area. The failure event sequence data are collected in the field and then preprocessed. For example, if several events of the same type are recorded at a single time point, only one event will be kept.

The next step is recognizing failure signatures in the event sequence data by using the approach to be presented in Section 2.3. The frequent failure signatures identified using this approach should be checked by area experts to decide whether or not they are real signatures. If yes, go to next step. If no, then the *benign signatures* are removed from the set of fault signatures. After this step, physically significant failure signatures are identified, and thus this step serves as the variable selection process for the prediction model building.

Finally, based on the failure signatures, we can build the failure prediction model for the event sequence data. The method to build such a model will be presented in Section 2.4.

2.3. *Frequent failure signature identification*

In this subsection, we present our approach to extract frequent failure signatures from event sequence data. In practice, engineers are interested in finding failure signatures because they want to find how trigger events affect the occurrence of failure events. In other words, *trigger events* are those events that can cause the occurrence of failure events. Using system knowledge, a trigger event can be associated with a specific fault (Holloway, 1994). However, when physical relationships between events are not available, we have to identify the trigger events by using *temporal relationships* in the event sequences. In this paper, we only consider this case. An example of failure signatures, denoted as $\phi = (\{A, B, K\}, \{A, B < K\})$, can be found in Fig. 1. The latter part (*partial order*) of ϕ, $\{A, B < K\}$, means *the set of trigger events* $\mathbf{T_r} = \{A, B\}$ occur before K in the failure signature, but the order between A and B is not fixed and thus ϕ is a *parallel* failure signature, whereas $\alpha = (\{A, B, K\}, \{A < B < K\})$ is a *serial* signature because the order between trigger events A and B is fixed. All failure signatures can be recursively generated using these two basic types. Therefore, we only consider parallel and serial signatures in this paper. Hereafter we use *italic* Greek letters, such as α and ϕ, to denote failure signatures.

In practice, engineers are interested in finding patterns containing events that occur quite close together in time. The closeness of events in failure signatures is specified by experts based on experience and the physical operation principles of the system. For example, CT maintenance personnel are interested in failure signatures that occur over 30 minutes in the CT log files. In the frequent failure signature identification, we use the term *window* to define the closeness of events, that is, we only consider those failure signatures that occur in a time window of a given width w. The window width should be specified by the area experts based on experience and the physical operation principles of the system.

For a given window width of w, it is often possible to extract many failure signatures from the event sequence. Only *frequent* failure signatures, whose occurrence *frequency* is

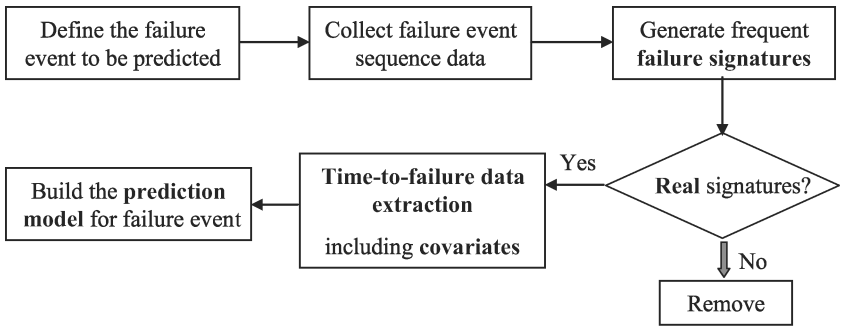

Fig. 3. The procedure for failure event prediction model building.

not smaller than a prespecified frequency threshold Δ_{fr}, are kept for further analysis. The frequency of a failure signature α is the proportion of all windows containing K that also contain the whole failure signature α. From a statistics viewpoint, we may view the frequency of a failure signature α as follows. Imagine that we arbitrarily place a window of size w on the time line, if the window covers the failure event K, then the frequency of the failure signature α is the probability that the window will also cover the whole failure signature α. A larger frequency means that the trigger events contained in fault signature α will occur with a greater probability before failure event K in the same window. Put in another way, if a failure signature is not frequent, that means that we will have a limited number of survival times related to this failure signature. From a statistical viewpoint, in this case we may not be able to estimate its effect due to the limited number of observations. In practice, the area experts also specify the frequency threshold value Δ_{fr}. The formal concepts of *failure signature*, *window* and *frequency* can be found in the Appendix.

A flow chart of the algorithm for frequent failure signature extraction is shown in Fig. 4. The ***inputs*** of this algorithm are the event sequence data $\mathbf{S}$, the failure event K, a frequency threshold Δ_{fr}, and a window width w. Notice that K, Δ_{fr} and w are specified by the area experts. The ***outputs*** are the set $\mathbf{F}$ of frequent parallel/serial failure signatures $\mathbf{F}_L$ with different length L and the corresponding frequencies. Suppose that the largest length of identified signatures is r, thus we have $\mathbf{F} = \{\mathbf{F}_2, \ldots, \mathbf{F}_r\}$, where the length of failure signatures L is $L = 2, 3, \ldots, r$.

Similar to the WINEPI approach (Mannila *et al.*, 1997), the core part of our algorithm has two alternating phases: (i) building new candidate failure signatures; and (ii) identifying the frequent failure signatures from the candidate set. The algorithm stops if no frequent failure signatures are recognized from the data. The basics of this algorithm are as follows.

Step 1. Extract event sequence data from the system log files.

Step 2. Calculate the set of event types, $\mathbf{E}$, which is used to generate the candidate failure signatures in Step 5.

Step 3. Calculate the number of windows of size w covering K.

Step 4. Decompose the event sequence into several segments in case the number of failure events in the event sequence is quite small (rare failure events), i.e., the number of windows covering K is small. Through this step, we can improve the algorithm efficiency because we only keep the sequence segments $\mathbf{S}_i$ ($i = 1, 2, \ldots, p$), in which the distance between the beginning of the segment to the first failure event K is equal to w (the end point is the event K), while other parts of $\mathbf{S}$ will be discarded.

Step 5. Generate the candidate failure signatures $\mathbf{C}_2$ with $L = 2$. The candidate signatures are in the form of $(\{E, K\}, \{E < K\})$, $E \in \mathbf{E}$, $E \neq K$ and the candidate parallel and serial signatures with $L = 2$ are the same.

Step 6. Compute *frequent* parallel and/or serial failure signatures for every $\mathbf{S}_i$, $i = 1, 2, \ldots, p$. Refer to Equation (A1) for the calculation of frequency.

Step 7. Increase the length of candidate failure signatures by one.

Step 8. Generate the candidate signatures $\mathbf{C}_{L+1}$ based on the identified frequent failure signatures $\mathbf{F}_L$. This is because of the fact that all sub-signatures of one failure signature α occur at least as α, thus we can build longer signatures from shorter ones.

Step 9. Output $\mathbf{F} = \{\mathbf{F}_2, \ldots, \mathbf{F}_r\}$, if NO further frequent failure signatures are recognized.

In Step 6, the basic idea is to slide a window of a given width along the time line and count the number of windows that cover the *candidate* failure signature α. If the frequency of α is either equal to or larger than the specified frequency threshold Δ_{fr}, we incorporate it into the set $\mathbf{F}_L$. Typically, two adjacent windows are often very similar to each other since we only move the window by one time unit along the time line at a time. Thus, in the algorithm we only need

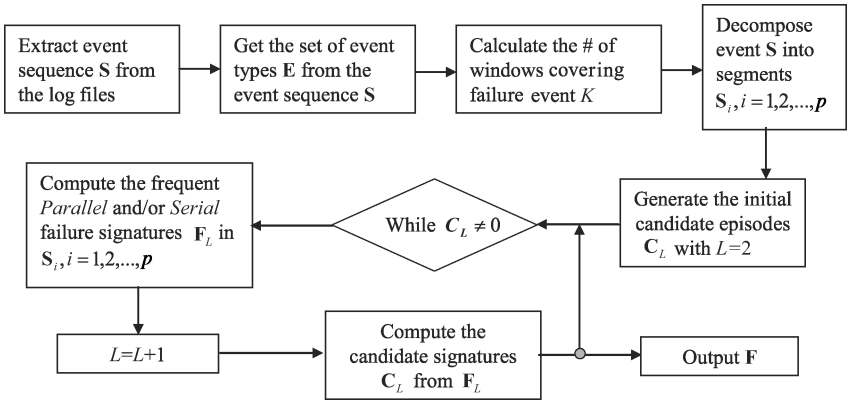

Fig. 4. Flow chart of the frequent failure signature identification algorithm.

to "update" the information in the current window. The details of Step 6 are as follows.

(a) In order to identify frequent *parallel* failure signatures, for each event in the failure signature ϕ (including trigger events set $\mathbf{T_r}$ and K), we maintain a flag $\phi.flag[j]$, $j = 1, 2, \ldots, L$, (L is the length of the failure signatures) to show whether or not the event is present in the window. When the jth event of ϕ is in the window, we have that $\phi.flag[j] = 1$. We also have a list $\phi.time[j]$, $j = 1, 2, \ldots, L$, which is used to record the occurrence time for each event in ϕ. If the ith event occurs multiple times in the window, then all the occurrence epochs are recoded in $\phi.time[j, :]$ in the order from small to large. When all flags $\phi.flag[j] = 1$, $j = 1, 2, \ldots, L$, AND the latest event in $\mathbf{T}_r$ occur before any of the failure events K in the window, the failure signature ϕ occurs entirely in the window, the occurrence indicator of ϕ ($\phi.indicator$) will change from zero to one and then the counter $\phi.counter$ will be increased by one. When sliding the window, we just update $\phi.flag[j]$, $\phi.time[j](j = 1, 2, \ldots, L)$ and $\phi.indicator$, if an event *related with* α enters into or leaves the window. If the value of $\phi.indicator$ is still one, we can increase $\phi.counter$ by one instead of checking the whole failure signature in the new window.

(b) In order to identify frequent *serial* fault signatures, the algorithm similar to the WINEPI approach is utilized. We make use of state automata to accept candidate signatures. The transition diagram of the finite automaton for the serial signature α (Fig. A1), is shown in Fig. 5. In this figure, $q_0, q_1, \ldots, q_4$ are states, and a state marked with double circles is either the initial or the final state. When the first event A of α enters the window, the corresponding automaton will be initialized and when this event A leaves the window, the automaton will be removed. If the automaton reaches its final status, which means the signature α occurs entirely in the window, we increase the number of windows which cover α by one. Similar to (a), when sliding the window, we just update the status of the corresponding automaton if an event *related to* α enters into or leaves the window. If the automaton for α is still in its final status, we can increase the occurrence number of α by one.

Whereas the WINEPI approach can be used to recognize general *episodes* (temporal patterns) in an event sequence, our proposed algorithm can identify frequent *failure signatures* more efficiently. The time complexities for calculation of the collection $\mathbf{F} = \{\mathbf{F}_2, \ldots, \mathbf{F}_r\}$ of frequent *injective* (refer to the Appendix for definition) parallel and serial failure signatures in all event sequence segments $\mathbf{S}_i$ $(i = 1, 2, \ldots, p)$ are $o(\sum_{L=2}^{r+1}[p \times |\mathbf{C}_L| \times L + pw])$ and $o(\sum_{L=2}^{r+1}[p \times w \times |\mathbf{C}_L| + pw])$, respectively, where w is the window width, L is the length of the failure signatures, $\mathbf{C}_L$ is the set of candidate failure signatures, and $|\mathbf{C}_L|$ is the number of the failure signatures in $\mathbf{C}_L$. The proof of this result can be obtained from the authors.

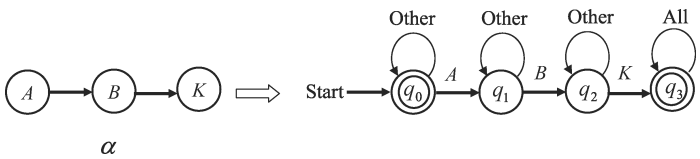

Fig. 5. The transition diagram of the finite automata based on α.

2.4. *Failure prediction using the Cox proportional hazard model*

As stated in Equation (4), the Cox proportional hazard model is used to model the hazard rate as a function of time-dependent covariates. The conditional survival function on covariate vector $\mathbf{Z}(t)$ can be expressed in terms of a baseline survival function $S_0(t)$ as follows:

$$S(t) = S_0(t)^{\exp(\boldsymbol{\beta}^T \mathbf{Z}(t))}. \tag{6}$$

The coefficient vector $\boldsymbol{\beta}$ in Equations (4) and (6) can be estimated by the maximum likelihood solution of the partial likelihood function (Klein and Moeschberger, 2003).

As we mentioned in Section 2.1, incorporating trigger events into the Cox model as time-dependent covariates, means that we can predict the failure event based on *trigger events*. Furthermore, not only the trigger events but also the relationships among these events can be derived from the recognized failure signatures. Thus, we can make full use of all the information contained in the failure signatures.

For a parallel failure signature $\phi = (\{\mathbf{T_r}, K\}, \{\mathbf{T_r} < K\})$, we can use every trigger event in $\mathbf{T_r}$ as time-dependent covariates, meanwhile, the whole $\mathbf{T_r}$ could be viewed as the interaction term among these trigger events. For example, $\phi = (\{A, B, K\}, \{A, B < K\})$ is a frequent failure signature, then other two failure signatures $\phi_1 = (\{A, K\}, \{A < K\})$ and $\phi_2 = (\{B, K\}, \{B < K\})$ should be also frequent, we could set two time-dependent covariates for α_1 and α_2 respectively as follows:

$$Z_A(t) = \begin{cases} 0, & 0 \le t < \mathrm{t}_A, \\ 1, & t_A \le t \le \text{ the end of the TIBF}, \end{cases}$$

$$Z_B(t) = \begin{cases} 0, & 0 \le t < t_B, \\ 1, & t_B \le t \le \text{ the end of the TIBF}. \end{cases}$$

where t_A and t_B are the occurrence times of event A and B respectively. For signature ϕ, we can use $Z_A(t) \times Z_B(t)$, the interaction between $Z_A(t)$ and $Z_B(t)$, in the regression model to study its effect on the occurrence of the failure event.

For a serial failure signature $\alpha = (\{\mathbf{T_r}, K\}, \{\mathbf{T_r} < K\})$, in addition to the time-dependent covariates for every trigger event, the whole set $\mathbf{T_r}$ can also be viewed as a time-dependent covariate. For example, $\alpha = (\{A, B, K\}, \{A < B < K\})$, we could also set a time-dependent covariate for

α as follows:

$$Z_{A<B}(t) = \begin{cases} 1, & t_A \le t_B \le t \le \text{the end of the TIBF}, \\ 0, & \text{otherwise.} \end{cases}$$

Through this way, we can predict the occurrence of a failure event by combining the failure signatures. After estimating the baseline survival function $S_0(t)$ and the coefficient vector $\boldsymbol{\beta}$ from the data, we can predict the occurrence of a failure event by using Equations (4) and (6).

With this method, our data can be viewed as follows. We have independent time-to-failure data $T_j, j = 1, 2, \ldots, n+1$, which correspond to the jth TIBF. δ_j is the failure event indicator for the jth TIBF, thus we have $\delta_j = 1$ for $j = 1, 2, \ldots, n$. For the last TIBF, the indicator value depends on whether or not the occurence times of failure event occurs at the end of the event sequence; if yes, $\delta_{n+1} = 1$, otherwise $\delta_{n+1} = 0$. If the last TIBF is censored, the failure event and the censoring time are independent. We also have covariate vector $\mathbf{Z}_j(t) = [Z_{j1}(t), Z_{j2}(t), \ldots, Z_{jy}(t)]^T$ for the jth TIBF, where y is the length of covariate vector. The covariates are coded as illustrated above. Using these data, we can fit the Cox model to the time-to-failure data and then predict the time interval and confidence for the occurrence of the failure event.

In the following section, a comprehensive case study and a real-world example are presented to illustrate this procedure.

3. Case studies

To show the effectiveness of our procedure as well as the performance of the Cox model with time-dependent covariates, we carried out the following case studies.

3.1. *Numerical case study*

In this case study, a simulated event sequence was used and thus we first need to generate the event sequences. The procedure is the reverse of the procedure to extract time-to-failure data from the event sequence, which is illustrated in Fig. 2. Because no distributional assumption is needed for the Cox model, we were able to generate the time intervals randomly and independently. After that, by linking all the time intervals together we created an event sequence. For a single hypothetical failure event sequence, we generated 1000 time intervals. In our study, all these 1000 time intervals were exactly observed, which means no censoring in our data. In addition to failure event K, there are three event types A, B and C in the event sequence. Event type A may occur at most once in each TIBF. The event types B and C are in the same case. To test the algorithm for frequent failure signature detection, the occurrence number of event C in the event sequence were set very small, which means that we will not get any failure signature containing event C when a comparably large threshold for the frequency is given. Some details of the simulation will now be discussed.

1. The $N = 1000$ failure times follow a Weibull distribution with parameters $\alpha = 3$, $\lambda = 50$ with shape parameter and scale parameter as 3 and 50; respectively.
2. For each TIBF, the occurrences of events A, B and C are assumed to be independent of one another. We also assumed that the events A, B and C occur within 60, 40 and 5% of all the TIBF respectively, that is, the occurrence of events A, B and C in each TIBF independently follow a Bernoulli distribution with $p = 0.6$, 0.4 and 0.05, respectively.
3. For those time intervals during which events A, B and/or C occur, we generated the occurrence times of the corresponding event according to a specified distribution. For A, the assumed distribution of the occurrence time was log normal with $\mu = 2, \sigma = 1$; for B, it was an exponential distribution with $\lambda = 10$. And uniform distribution with parameters $a = 10, b = 20$ was assumed for the occurrence times of event C. In total, $N = 1000$ sets of time-dependent covariates (events A and B) are generated.
4. Assuming that the coefficient vector $\boldsymbol{\beta}$ in Equations (4) and (6) is known, we can use the *permutational algorithm* (Abrahamowicz *et al.*, 1996; Leffondre *et al.*, 2003) to randomly pair the time-dependent covariate vector and the TIBF, according to the probability based on the partial likelihood given the values of $\boldsymbol{\beta}$. The assumed values of $\boldsymbol{\beta}$ are listed in Table 1. Because the event C is not included in the frequent failure signatures, we only considered $\boldsymbol{\beta}$ values for time-dependent covariates of events A and B. In the study, two scenarios were studied, the difference being whether or not an interaction exists between $Z_A(t)$ and $Z_B(t)$. In scenario 1, this interaction is assumed to have no effect on the hazard rate.

To apply the proposed failure prediction method, the first step is to identify the frequent failure signatures. A window width of $w = 50$ and a threshold for the frequency of $\Delta_{fr} = 5\%$ were used. Three fault failure signatures were identified $\lambda_1 = (\{A, K\}, \{A < K\})$, $\lambda_2 = (\{B, K\}, \{B < K\})$ and $\lambda = (\{A, B, K\}, \{A, B < K\})$ for both scenarios.

We ran the simulation algorithm 1000 times and obtained 1000 event sequences. For each event sequence, we obtained an estimate of $\boldsymbol{\beta}$, denoted as $\hat{\boldsymbol{\beta}}$. For each scenario studied, we calculated the mean $\bar{\hat{\boldsymbol{\beta}}}$ of the 1000 regression coefficients with the corresponding 95% confidence interval

Table 1. Assumed values of $\boldsymbol{\beta}$ for two scenarios

Covariates	*Scenario 1*	*Scenario 2*
$Z_A(t)$	0.6	0.6
$Z_B(t)$	1.2	1.2
$Z_A(t) \times Z_B(t)$	**0**	**1.6**

Table 2. Mean of estimates, corresponding to a 95% CI and relative bias for both scenarios

Scenario	*Covariates*	β	$\bar{\hat{\beta}}$	*95% CI*	$(\bar{\hat{\beta}} - \beta)./\beta$
1	$Z_A(t)$	0.6	0.6037	[0.5984, 0.6090]	0.0061
	$Z_B(t)$	1.2	1.2038	[1.1974, 1.2102]	0.0031
	$Z_A(t) \times Z_B(t)$	0	−0.0010	[−0.0090, 0.0070]	
2	$Z_A(t)$	0.6	0.6039	[0.5987, 0.6090]	0.0065
	$Z_B(t)$	1.2	1.2040	[1.1976, 1.2104]	0.0033
	$Z_A(t) \times Z_B(t)$	1.6	1.6022	[1.5936, 1.6108]	0.0014

based on normal approximation. Also, we calculated the relative bias as the ratio $(\bar{\hat{\beta}} - \beta)./\beta$. The symbol "./" represents element-wise division.

Based on the results shown in Table 2, we can see that our algorithm to build the Cox model incorporating time-dependent covariates is quite accurate. The relative bias $(\bar{\hat{\beta}} - \beta)./\beta$ is in a small range of 0.14–0.65% (except for $Z_A(t) \times Z_B(t)$ in scenario 1).

The prediction model for scenario 1 is

$$\hat{h}_{(1)}[t \mid \mathbf{Z}(t)] = \hat{h}_{(1),0}(t) \exp[\hat{\beta}_{(1),1} Z_A(t) + \hat{\beta}_{(1),2} Z_B(t)].$$

In this case, the model has two predictors which are the time-dependent covariates $Z_A(t)$ and $Z_B(t)$. Although $\lambda = (\{A, B, K\}, \{A, B < K\})$ is a frequent failure signature, the effect of the interaction between $Z_A(t)$ and $Z_B(t)$ is not significant, according to the p-value of the Wald test. That is, we can consider the occurrence of event A and B separately.

For scenario 2, the prediction model includes the effect of the interaction. The full model is

$$\begin{aligned}\hat{h}_{(2)}[t|\mathbf{Z}(t)] &= \hat{h}_{(2),0}(t) \exp[\hat{\beta}_{(2),1} Z_A(t) + \hat{\beta}_{(2),2} Z_B(t) + \hat{\beta}_{(2),3} Z_A(t) \\ &\quad \times Z_B(t)].\end{aligned}$$

The p-value of the Wald tests tell us that the interaction effect is significant, which means the effect of concurrence of event A and B is larger than the sum of effects of A and B which occurs separately, because the estimate $\beta_{(2),3}$ is positive in this case.

3.2. *Failure prediction model building for real CT log file data*

The real-world data to be analyzed are CT usage log files. A large amount of data is generally recorded over a month of monitoring. Failure event sequence data can be extracted from the log file. After data preprocessing, there were 7199 events belonging to 179 different event types that occurred in the month. This data set is available from the authors. Since we do not know the physical relationships between these event types, the technique for frequent failure signature identification presented in this paper is needed.

To maintain confidentiality, we simply denote the failure event as K, which represents a provisional malfunction of the scanner. Using our algorithm we were able to identify 50 frequent parallel failure signatures based on the window width and frequency threshold given by area experts. The results from our algorithm were compared with the failure signatures manually identified by field engineers. The comparison results are very satisfactory: essentially, all the signatures identified by the engineers were also identified by the algorithm. Among these signatures, three are viewed as useful. They are

$$\begin{aligned}\nu_1 &= (\{A, K\}, \{A < K\}),\\ \nu_2 &= (\{B, K\}, \{B < K\}),\\ \nu_3 &= (\{C, K\}, \{C < K\}).\end{aligned}$$

Also for confidentially reasons we simply use A, B and C to denote the trigger events and these events are all related to operator activities. No other failure signatures with a length larger than two were identified. Our next task was to investigate how the operator activities affect the occurrence of the malfunction of the CT scanner, that is, we constructed the prediction model based on these three failure signatures.

We set three time-dependent covariates for the three trigger events contained in the failure signatures. They were $Z_A(t)$, $Z_B(t)$ and $Z_C(t)$, which have the same form as that in Section 2.4. We do not have any fixed-time covariates for these data thus the covariate vector for the ith TIBF at time t is $\mathbf{Z}_i(t) = (Z_{iA}(t), Z_{iB}(t), Z_{iC}(t))^T$.

In the event sequence, the total number of K events is 106, thus there are 107 time intervals with the last one being censored. Using the stepwise approach, we selected significant covariates and interactions. We used the *Akaike Information Criterion* (AIC) to decide if we should add factors $Z_A(t)$, $Z_B(t)$ and $Z_C(t)$, or interactions, $Z_A(t) \times Z_B(t)$, $Z_A(t) \times Z_C(t)$ and $Z_B(t) \times Z_C(t)$ into our model. The process stops when no significant covariate and interactions are found. AIC examines the statistic:

$$AIC = -2 \log L + kp, \tag{7}$$

where L is the likelihood function, p is the number of regression parameters in the model and k is some specified constant (usually two). The parameters of the final regression model are listed in Table 3. In the table, exp(·) means the exponential function, SE(·) is the standard error of the estimate, whereas Z is the value of statistic.

Table 3. Analysis of variance table for the final model

	DOF	$\hat{\beta}$	$exp(\hat{\beta})$	$SE(\hat{\beta})$	**Z**	*p-value*
$Z_A(t)$	1	0.862	2.367	0.295	2.93	3.40×10^{-3}
$Z_B(t)$	1	1.216	3.372	0.296	4.1	4.10×10^{-5}
$Z_C(t)$	1	0.644	1.905	0.235	2.74	6.20×10^{-3}
$Z_A(t) \times Z_B(t)$	1	−1.55	0.212	0.45	−3.45	5.70×10^{-4}

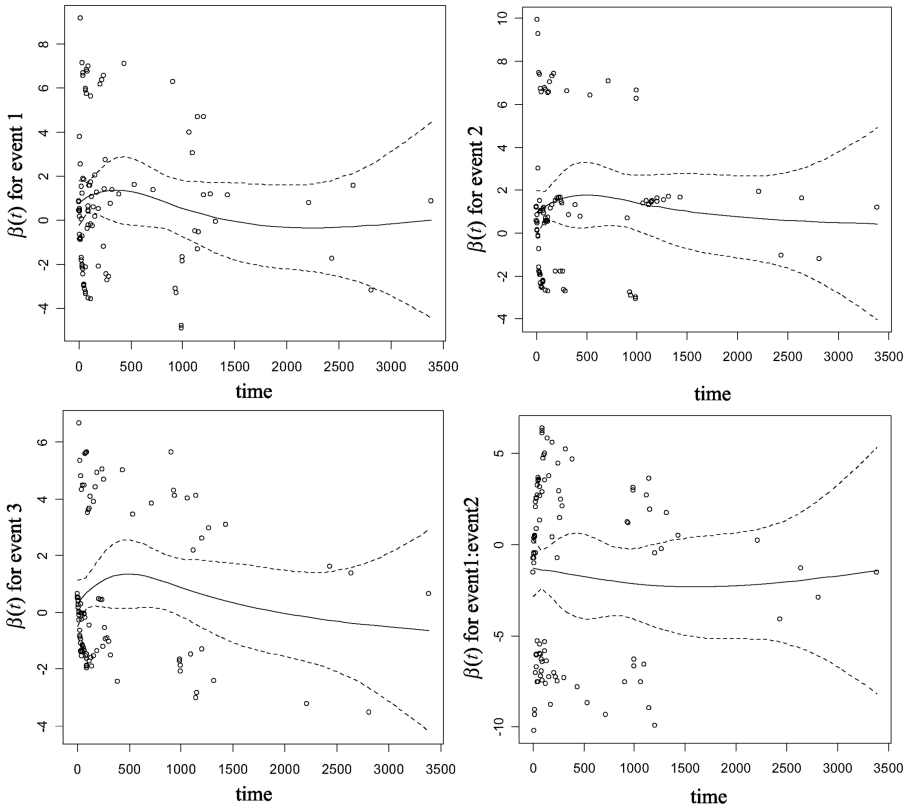

Fig. 6. Graphical checks of the proportional hazards assumption.

When fitting the Cox model to the data, we need to check the proportional assumption for every time-dependent covariate at the first place. The checking results by using the *cox* · *zph*(·) function in R are shown in Fig. 6. In the figure, the slopes of the smoothed curves for *scaled Schoenfeld residuals* are nearly zero. Also, *p*-values of the corresponding tests (not given here) show that the proportional hazard assumption holds for all three time-dependent covariates. In practice, if the proportional hazards assumption is violated for some covariate, we can stratify on that variable and fit the Cox model within each stratum for other covariates (Klein and Moeschberger, 2003).

Analogous to R^2 in linear regression analysis, a measure of goodness-of-fit for the Cox proportional hazard model was proposed by Cox and Snell (1989) to be

$$R^2 = 1 - \exp\left[\frac{2}{N}(LL_0 - LL_{\hat{\boldsymbol{\beta}}})\right] \tag{8}$$

where $LL_{\hat{\boldsymbol{\beta}}}$ is the log partial likelihood for the fitted Cox proportional hazard model, LL_0 is the log partial likelihood for model zero, and N is the number of time intervals in the event sequence. The final result of the Cox–Snell R^2 is equal to 0.204, thus the Cox model fits our data well (Verweij and Van Houwelingen, 1993). In addition, we also checked the model for outliers using *deviance residuals*, and no outliers were found.

The baseline survivor function is shown in Fig. 7. We now interpret our final hazard model. The *hazard ratio* (*HR*) between time points t_2 and t_1 is defined as

$$HR = \frac{h_0(t)\exp[\boldsymbol{\beta}^T\mathbf{Z}(t_2)]}{h_0(t)\exp[\boldsymbol{\beta}^T\mathbf{Z}(t_1)]} = \exp\{\boldsymbol{\beta}^T[\mathbf{Z}(t_2) - \mathbf{Z}(t_1)]\}.$$

Thus, based on the results shown in Table 3, we can estimate the hazard ratios. Assume that only event A occurs in the time interval $[t_1, t_2]$, then we can write $\widehat{HR} = \exp\{\hat{\boldsymbol{\beta}}^T[\mathbf{Z}(t_2) - \mathbf{Z}(t_1)]\} = \exp(\hat{\beta}_A) = 2.367$, which means the hazard rate after A occurs is 2.367 times that when event A does not occur. While both A and B occur in the interval, the hazard rate will be $\exp(\hat{\beta}_A + \hat{\beta}_B + \hat{\beta}_{AB}) = \exp(0.862 + 1.216 - 1.55) = \exp(0.528) = 1.696$ times that when events A and B do not occur. That is, other than the effects of A and B, the interaction also impacts the hazard rate. Accordingly, the survival function will change when trigger events occur. The predicted survival function can be calculated from our results.

We can use the estimated regression coefficients listed in Table 3 and baseline survival function in Fig. 7 to predict the occurrence of failure event K. For example, if all the trigger events A, B and C do not occur in the system, then the failure event K will occur in the time interval [100, 150] minutes from the start of the TIBF with a probability of $(0.5693 - 0.4776) = 0.0917 \approx 9.2\%$, which can be calculated from the estimated baseline survival

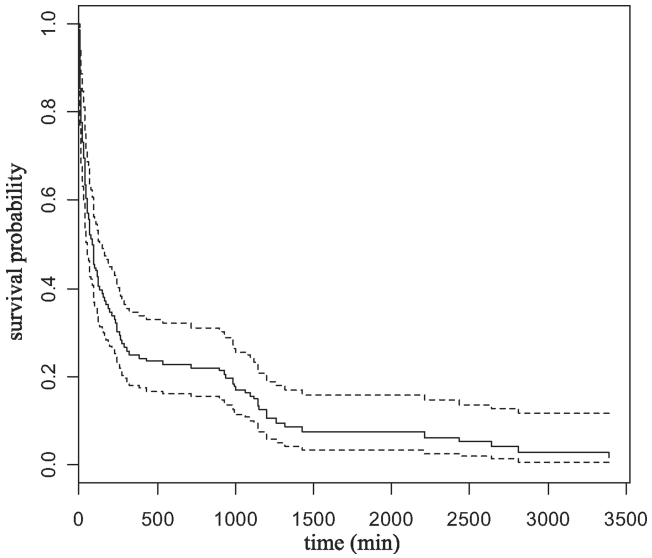

Fig. 7. Estimated baseline survival function.

function. However, if A and B do occur before the time interval, the probability for K occurring in [100, 150] will be increased to $(0.5693^{\exp(\hat{\beta}_A+\hat{\beta}_B+\hat{\beta}_{AB})} - 0.4776^{\exp(\hat{\beta}_A+\hat{\beta}_B+\hat{\beta}_{AB})}) = (0.5693^{1.696} - 0.4776^{1.696}) = 0.0991 \approx 9.9\%$.

4. Conclusions

An effective data-driven technique to predict the occurrence of failure events based on event sequence data has been presented. The Cox proportional hazard model, which has some advantages over other models and thus predominates in biomedical survival analysis, can be used to provide a rigorous statistical prediction of system failure events. However, due to the difficulties that can be encountered when fitting the Cox model directly to the event sequence data, we need to build a prediction model based on failure signatures. An algorithm to extract the frequent failure signatures has been developed and two types of failure signatures—parallel and serial signatures—can be identified efficiently through the method. Based on the recognized failure signatures, an approach to prediction model building has been developed. By coding the failure signatures as time-dependent covariates and interactions, we can fit the Cox model to the data and thus build a failure prediction model for the event sequence based on the frequent failure signatures. Finally, we illustrated the effectiveness of our approach through a numerical case study and a real-world example. In summary, the whole procedure provides a systematic methodology to analyze the failure event sequence data and can be used in the field of failure prediction.

A very interesting open issue is failure prediction for multiple event sequences. For the case in which we have several event sequences collected from different pieces of equipment, how do we predict the failure event by combining the data together? Furthermore, it will be very interesting to consider the dependence among the time intervals and include the relationship in the prediction model. The extensions of Cox models for recurrent event data of multiple subjects, such as the Prentice-Williams-Peterson (Prentice *et al.*, 1981) and Wei-Lin-Weissfeld (Wei *et al.*, 1989) models, will be investigated in a future study. The results along this direction will be reported in the future.

Acknowledgements

Financial support for this work was provided by GE Healthcare and the National Science Foundation under grant DMI-0545600. The authors thank the editor and the referees for their valuable comments and suggestions.

References

Abrahamowicz, M., Mackenzie, T. and Esdaile, J.M. (1996) Time-dependent hazard ratio: modeling and hypothesis testing with application in Lupus Nephritis. *Journal of the American Statistical Association*, **91**(436), 1432–1493.

Agrawal, R. (1996) Data mining: the quest perspective. *Australian Computer Science Communications*, **18**(2), 119–120.

Chen, Y.-L. and Provan, G. (1997) Modeling and diagnosis of timed discrete event systems—a factory automation example. *Presented at the American Control Conference*, Albuquerque, NM.

Cox, D.R. (1972) Regression models and life tables (with discussion). *Journal of the Royal Statistical Society B*, **34**, 187–220.

Cox, D.R. and Snell, E.J. (1989) *The Analysis of Binary Data*, 2nd edn., Chapman and Hall, London, UK.

Das, S.R. and Holloway, L.E. (1996) Learning of time templates from system observation, *Presented at the American Control Conference*, Seattle, WA.

Dunham, M.H. (2003) *Data Mining: Introductory and Advanced Topics*, Pearson, Upper Saddle River, NJ.

Elsayed, E.A. and Chan, C.K. (1990) Estimation of thin-oxide reliability using proportional hazards models. *IEEE Transactions on Reliability*, **39**(3), 329–335.

George, L. (2003) Biomedical survival analysis vs. reliability: comparison, crossover, and advances. *The Journal of the RAC*, **11**(4), 1–5.

Holloway, L.E. (1996) Distributed fault monitoring in manufacturing systems using concurrent discrete-event observations. *Integrated Computer-Aided Engineering*, **3**(4), 244–254.

Holloway, L.E. and Chand S. (1994) Time templates for discrete event fault monitoring in manufacturing systems. *Presented at the American Control Conference*, Baltimore, MD.

Hougaard, P. (1999) Fundamentals of survival data. *Biometrics*, **55**, 13–21.

Hougaard, P. (2000) *Analysis of Multivariate Survival Data*, Springer-Verlag, New York, NY.

Klein, J.P. and Moeschberger, M.L. (2003) *Survival Analysis: Techniques for Censored and Truncated Data*, Springer-Verlag, New York, NY.

Klemettinen, M. (1999) A knowledge discovery methodology for telecommunication network alarm databases. Ph.D. thesis, University of Helsinki, Helsinki, Finland.

Leffondre, K., Abrahamowicz, M. and Siemiatycki, J. (2003) Evaluation of Cox's model and logistic regression for matched case-control

data with time-dependent covariates: a simulation study. *Statistics in Medicine*, **22**, 3781–3794.
Mannila, H., Toivonen, H. and Verkamo, A.I. (1997) Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, **1**, 259–289.
Misra, K.B. (1992) *Reliability Analysis and Prediction: A Methodology Oriented Treatment*, Elsevier, Amsterdam, The Netherlands.
Pandalai, D.N. and Holloway, L.E. (2000) Template languages for fault monitoring of timed discrete event processes. *IEEE Transactions on Automatic Control*, **45**(5), 868–882.
Prentice, R.L., Williams, B.J. and Peterson, A.V. (1981) On the regression analysis of mulitivariate failure time data. *Biometrika*, **68**(2), 373–379.
Sampath, M., Sungupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1994) Diagnosability of discrete event systems. *Presented at the 11th International Conference on Analysis and Optimization of Systems: Discrete Event Systems*, Sophia Antipolis, France.
Srinivasan, V.S. and Jafari, M.A. (1993) Fault detection/monitoring using time petri nets. *IEEE Transitions on System, Man, and Cybernetics*, **23**(4), 1155–1162.
Valette, R., Cardoso, J. and Dubois, D. (1989) Monitoring manufacturing systems by means of petri nets with imprecise markings, in *Proceedings of the IEEE Conference on Intelligent Control*, Institute of Electrical and Electronics Engineers, Piscataway, NJ, pp. 233–238.
Verweij, P.J.M. and Van Houwelingen, H.C. (1993) Cross-validation in survival analysis. *Statistics in Medicine*, **12**, 2305–2314.
Wasserman, G.S. (2003) *Reliability Verification, Testing, and Analysis in Engineering Design*. Marcel Dekker, New York, NY.
Wei, L.J., Lin, D.Y. and Weissfeld, L. (1989) Regression analysis of multivariate incomplete failure time data by modeling marginal distributions. *Journal of the American Statistical Association*, **84**(408), 1065–1073.
Xiao, Y. and Dunham, M.H. (2001) Efficient mining of traversal patterns. *Data and Knowledge Engineering*, **39**(2), 191–214.

Appendix

Definitions of failure signature, window and frequency follows.

Failure signature: Let K denote the failure event and $\mathbf{T_r}$ the set of trigger events. A ***failure signature*** $\alpha = (\{\mathbf{T_r}, K\}, \{\mathbf{T_r} < K\})$ is the set of events $\{\mathbf{T_r}, K\}$, which occur within time intervals of a given size, and a total order $\{\mathbf{T_r} < K\}$, which represents that the trigger event set $\mathbf{T_r}$ occurs before K in the time intervals of specified size. A partial order, $\leq$, could exist on the events in $\mathbf{T_r}$:$(\mathbf{T_r}, \leq)$.

The failure signature is ***injective*** if no event type occurs twice or more in the signature. The ***length*** of a failure signature α, denoted as L, is the number of events contained in the set of events $\{\mathbf{T_r}, K\}$. Based on the definition of failure signature, we have $L \geq 2$.

Based on the partial order, $\leq$, on trigger events, there are three types of failure signatures shown in Fig. A1. $\alpha = (\{A, B, K\}, \{A < B < K\})$ is a *serial* failure signature because the order relation $\{A < B\}$ of trigger events A and B is a *total* order, while $\phi = (\{A, B, K\}, \{A, B < K\})$ is a *parallel* failure signature (notice that the order is not fixed for all $A \neq B$). In failure signatures α and ϕ, we can see failure signature $\lambda = (\{A, K\}, \{A < K\})$ is a ***sub-signature*** because it is contained in α and ϕ. The third type of failure signature, γ, is *composite*. The composite signature could be constructed based on the first two types recursively.

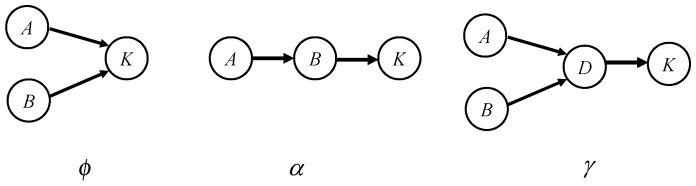

Fig. A1. The three basic types of failure signatures.

The definition of window, also used in Mannila *et al.* (1997), is presented as follows.

Window: Given an event sequence $\mathbf{S} = (T_s^{\mathbf{s}}, T_e^{\mathbf{s}}, \mathbf{s})$, a ***window*** $\mathbf{W} = (T_s^{\mathbf{w}}, T_e^{\mathbf{w}}, \mathbf{s_W})$ is a *part* of that event sequence, where $T_s^{\mathbf{w}}$ and $T_e^{\mathbf{w}}$ are the starting and ending times of the window respectively with $T_s^{\mathbf{w}} < T_e^{\mathbf{s}}$ and $T_e^{\mathbf{w}} > T_s^{\mathbf{s}}$, and $\mathbf{s_W}$ consists of those event pairs $(E_i, t_i^{\mathbf{s}})$ from $\mathbf{S}$ with $T_s^{\mathbf{w}} \leq t_i^{\mathbf{s}} < T_e^{\mathbf{w}}$.

The width of the window $\mathbf{W}$ is defined as $w = T_e^{\mathbf{w}} - T_s^{\mathbf{w}}$. Given an event sequence $\mathbf{S}$ and an integer w, we denote $\Omega(\mathbf{S}, w, K)$ the set of all windows of size w which cover failure event K on $\mathbf{S}$.

Frequency: The ***frequency*** of α in the set $\Omega(\mathbf{S}, w, K)$ of all windows of size w covering K on $\mathbf{S}$ is

$$fr(\alpha, \mathbf{S}, w, K) = \frac{\text{the number of } \mathbf{W} \text{ convering } \alpha \text{ in } \Omega(\mathbf{S}, w, K)}{\text{the number of } \mathbf{W} \text{ in } \Omega(\mathbf{S}, w, K)}. \quad \text{(A1)}$$

Given a threshold Δ_{fr} for the frequency, a given failure signature is *frequent* if $fr(\alpha, \mathbf{S}, w, K) \geq \Delta_{fr}$. Apparently, if α is frequent then all its sub-signatures are frequent.

Biographies

Zhiguo Li is a Ph.D. student in the Department of Industrial and Systems Engineering at the University of Wisconsin-Madison. His research interests focus on variation source identification in large complex systems, statistical modeling and reliability analysis of event sequence data.

Shiyu Zhou is an Assistant Professor in the Department of Industrial and Systems Engineering at the University of Wisconsin-Madison. He was awarded B.S. and M.S. degrees in Mechanical Engineering by the University of Science and Technology of China in 1993 and 1996 respectively, and a Master in Industrial Engineering and Ph.D. in Mechanical Engineering at the University of Michigan in 2000. His research interests center on in-process quality and productivity improvement methodologies obtained by integrating statistics, system and control theory, and engineering knowledge. The objective is to achieve automatic process monitoring, diagnosis, compensation, and their implementation in various manufacturing processes. His research is sponsored by the National Science Foundation, Department of Energy, NIST-ATP, and industry. He is a recipient of the CAREER Award from the National Science Foundation in 2006. He is a member of IIE, INFORMS, ASME and SME.

Suresh Choubey is a Master Black Belt—Design for Six Sigma and System Architect at GE Healthcare Global Service Technology. He

received M.S and Ph.D. degrees in Computer Science from the Center for Advanced Computer Studies, University of Louisiana at Lafayette. His research interests are RFID, Data Mining, automated methods of knowledge creation and content based image storage and retrieval systems.

Crispian Sievenpiper is a Lead System Architect at GE HealthCare Global Service Technology. He received M.S degrees in Computer Science and Statistical Science from the State University of New York. His research interests lie in the cost-effective application of system telematics and predictive service for complex machinery.