Received: 23 October 2018 | Accepted: 13 September 2019

DOI: 10.1002/qre.2577

RESEARCH ARTICLE

WILEY

Software reliability analysis for safety-critical and control systems

Pramod Kumar[1] | **Lalit Kumar Singh**[2] | **Chiranjeev Kumar**[1]

[1] Department of Computer Science & Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India

[2] Department of Computer Science & Engineering, Indian Institute of Technology (Banaras Hindu University), Varanasi, Uttar Pradesh, India

Correspondence
Pramod Kumar, Department of Computer Science & Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, India.
Email: pramod.16dr000212@cse.ism.ac.in

Abstract

The transition from analog to digital safety-critical instrumentation and control (I&C) systems has introduced new challenges for software experts to deliver increased software reliability. Since the 1970s, researchers are continuing to propose software reliability models for reliability estimation of software. However, these approaches rely on the failure history for the assessment of reliability. Due to insufficient failure data, these models fail to predict the reliability of safety critical systems. This paper utilizes the Bayesian update methodology and proposes a framework for the reliability assessment of the safety-critical systems (SCSs). The proposed methodology is validated using experiments performed on real data of 12 safety-critical control systems of nuclear power plants.

KEYWORDS

Nuclear Power Plant, Safety Critical Systems, Bayesian Belief Networks, System reliability

*

1 | INTRODUCTION

Safety-critical control system (SCCS) of Nuclear Power Plant (NPP) requires accurate reliability prediction for its effective performance.[1-3] Safety-critical system (SCS) performance relies severely on the software program being used. The software undergoes rigorous testing and certification process before being deployed in SCCS NPP systems. The software program is developed with superior quality and high reliability using international standards and hence confronts rare failures. For assessing and predicting the reliability, many software reliability models (SRMs)[4-6] have been proposed in the last few decades. Various SRMs are reviewed for their potential use in quantifying software reliability and probabilities to support the reliability modeling of digital systems in SCS. Broadly, these SRMs fall into four categories: software reliability growth models, Bayesian belief network (BBN), test-based methods, and correlation methods.

ACRONYMS AND ABBREVIATIONS: BBN, Bayesian Belief Network; BFRV, Bypass Feedwater-Regulating Valve; BFV, Bypass Feedwater Valve; CPT, Conditional Probability Table; CPU, Central Processing Unit; DFWCS, Digital Feedwater Control System; FWP, Feedwater Pump; MFP, Main Feedwater Pump; MFRV, Main Feedwater-Regulating Valve; MFV, Main Feedwater Valve; PDI, Pressure Differential Indicating; PWR, Pressurized Water Reactor; SCCS, Safety-Critical Control System; SCS, Safety-Critical System; SDLC, Software Development Life Cycle; SRGM, Software Reliability Growth Model; SRM, Software Reliability Model; SRS, Software Requirement and Specification; WDT, Watchdog Timer

*

We believe that SRM for SCS must have some special desirable characteristics:

1. There should be acceptable documentation of the method and its applications that can be understood and evaluated.
2. There should be a strong justification of the assumptions made.
3. It should be possible to consider the specific operating conditions of the software.
4. It should be able to address all the uncertainties for better accuracy.
5. There should be sufficiency of model parameters.
6. The models should have proper verification and validation ways.

Several models do not have description of intended uses of the method and all assumptions.[6,7] In general, the existing SRM assumes unrealistic assumptions and use failure history to estimate the parameters of the models in order to calculate the reliability. As of our knowledge, we have found very few SRM which consider the specific operating conditions of the software.[8] Software reliability growth model (SRGM) estimates the software reliability with high accuracy, if there is sufficiency of failure data and therefore are not suitable to SCS.

Software errors are generally due to unstated, ambiguous, and unrealistic requirements or because of design errors. From this, we believe that if it is possible to formulate any model that can accommodate errors in each phase of software development life cycle (SDLC) and also can propagate them in next phase, prior to the completion of software development, it will give more accurate reliability prediction results. A perfect designed software requirement and specification (SRS) not only depends on the requirements and analysis but also on the expert's opinion, the domain knowledge, and the experience of the SRS designing team, the resources available, and many other constraints. Taking these as the inputs for the requirements phase, a perfect SRS is designed. These inputs may be considered as "safety cases" or "safety arguments" as these are the collection of evidence responsible for the proper development of final product. Evidence for safety cases or safety arguments for each phase exists. Using these evidences, we can create BBN model for each phase of life cycle model and calculate the inferred probability using the safety argument values. A proper conditional probability is to be given to evidence in order to correctly assess the reliability of each phase.

The aim of this paper is to develop a Bayesian technique for reliability analysis of SCS. The proposed technique satisfies all the above-stated six desirable characteristics and hence suitable for SCS. It provides an analytical mechanism to capture the information starting from SRS phase (first phase) to operational phase (final phase) of SDLC.

Singh et al.[8] have proposed an approach for early prediction of software reliability using stochastic modeling for feedwater system. This approach addresses the unrealistic assumption of transition probabilities of Markov Chain, made by earlier researchers. However, this model is based on the observable states of the system and there is a possibility to miss out the undesirable scenario that may happen in the future. Hence, devising a model that can embed the development characteristics of the software will give more accurate results.

Kang et al.[9] proposed an input profile-based software testing method for the quantification of the failure probability using Binomial distribution and Bayesian approach. The proposed method is capable to consider operational profile that can be produced based on process parameters. However, the method neither considers the SDLC process nor a mechanism to deal with insufficiency of data.

Eom et al.[10] proposed a verification and validation-based fault estimation method using Bayesian network to estimate the remaining faults for SCS after the SDLC is completed. Moreover, estimating remaining fault is only a reliability metric. The computation of reliability has not been discussed, based on which the regulators decide the operation of system.

Singh et al.[11] have proposed an effective method to compute the parameters of the Markov chain, which is suitable to predict the reliability of the system. The validation is done on operational profile data of 2 years. However, in case of SCSs, the operational profile data of 2 years is very less. Therefore, the approach is more conservative to be accepted for SCS.

Kumar et al.[12] have proposed a method for safety analysis of SCS with a case study of reactor core isolation cooling system of NPP. The approach provides an effective methodology to construct a Petri net model. However, to perform the reliability analysis, the throughput or rate of transitions must be known. Therefore, the uncertainties involved in the model will not give accurate results in case of SCS.

Kim et al.[13] came up with a novel approach for the estimations of the parameters for the SRGM. The authors proposed a methodology for finding the parameters using real-valued genetic algorithm instead of the approaches like

the maximum likelihood estimation, numerical methods, or least square estimation methods. For the improvement of the performance and the accuracy of the parameter estimation, two real-valued genetic operators, namely heuristic cross over and nonuniform mutation, were applied. The fitness value of a chromosome is calculated using the actual and the estimated number of failures at time t_i, which infers that proper fault history and its occurrence time is required for the chromosome fitness function calculation, which is not always feasible for the case of SCS.

Sharma et al.[14] unbolted the problem of optimal model selection by different practitioners using a limited number of model selection criteria. However, the approach has not been validated on any SCS.

Andreou and Chatzis[15] have addressed the issues of lack of research in count data regression and inefficient algorithms for the posterior calculation of hierarchical Bayesian approach. To overcome these issues, the authors proposed a doubly stochastic Poisson process for count data regression using hierarchical Bayesian model. The failure rate at each time unit was modeled as a space parameter expressed as the output variable for the Bayesian model driven by the metrics data. However, the assumptions made for the model formulation regarding N releases, x_i software metrics pertaining to the ith release, and the corresponding defect counts, with corresponding time durations $\{t_i\}_{i=1}^{n}$, cannot be justified with respect to SCS systems.

Bai[16] has developed an extended Markov Bayesian network (ENBM) model for the reliability evaluation with an operational profile and compared its performance with the Kaaniche-Kanoun model (K&K model)[17] on SPACE failure data. However, it does not consider the experts' opinion in the early phases of the SDLC.

Peng et al.[18] proposed a model fabricated with an information integration framework, two information toolkits, and three Bayesian models. However, the steps involved in the implementation of the model are not justified correctly. For example, one of the steps says "select reliability model for the product in the predevelopment stage." The basis of model selection is not mentioned. No assumptions regarding the failure are being made in order to choose the reliability models.

Xing et al.[19] presented a Bayesian method for binomial systems using earlier test data and prototype. However, the expert's knowledge is not exploited in the model. If we can exploit the expert's knowledge along with the earlier test data and the prototype, then the model could have been more effective.

Li et al.[20] studied the reliability biasness due to the assumed prior product failure distribution in the reliability estimation and proposed a model using sequential test procedure and the Bayesian theorem to overcome the issue. However, the methodology has not been validated on real-time data. And, the approach remains silent on the case of decreasing reliability growth.

Yu et al.[21] proposed a Bayesian network-based program dependence graph for fault localization. The approach overcomes the problem of inferring fault localization across nonadjacent nodes. The performance evaluation of the model was done on *Siemens suite* and *Space* datasets. However, these datasets have been widely used in fault localization. The work needs to be generalized by verifying and validating it on different varieties of datasets. The approach deals with only the correct implementation of the program, and incorrect implementation also needs to be envisaged.

Campodonico and Singpurwalla[22] used the Bayesian methodology for predicting the number of faults using logarithmic Poison model in software. The model does not consider the expert's knowledge in the earlier stages of the life cycle model.

Zou et al.[23] developed a software platform known as Software Reliability Auto-Modeling and Analysis Platform. The proposed technique is based on flow network model, which is hard to create for lengthy programs. Further, it treats the process as deterministic process and fails to capture the dynamic behavior of the system.

The paper is organized as follows: Proposed framework, along with the assumptions and requirements for the model, is discussed in Section 2. Section 3 depicts the case study of digital feedwater control system (DFWCS) along with its failure impacts. Application of the proposed framework on case study is discussed in Section 4. Result and validation are shown in Section 5. Section 6 concludes the research work.

2 | PROPOSED FRAMEWORK

We exploit Bayesian Update methodology to estimate the failure probability of the system using expert's knowledge. If the system is not newly installed or not-of-first-kind, available operational profile data shall be combined along with expert's knowledge. This estimate is called prior estimate. Prior estimate can be combined with current estimate, known as Likelihood, to forecast the failures, known as posterior distribution. The mechanism, containing these three phases, is shown in Figure 1.

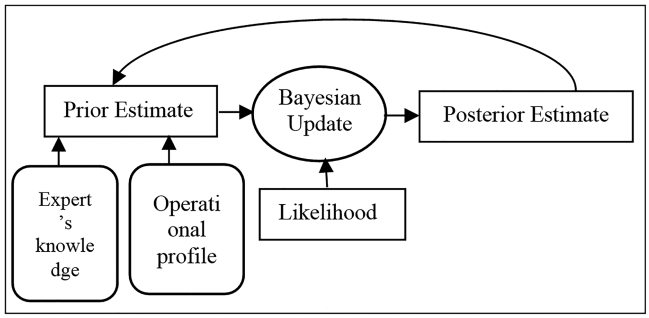

FIGURE 1 Effects of Safety-critical system failure

2.1 | BBN model construction

BBN models are directed acyclic graph used in many areas like probablistic reaoning, robust localization, and decision making.[24] The BBNs are helpful in reliability estimation due to its representation of the joint distribution and reasoning under vagueness.

The following three steps are involved in the construction of the BBN model:

1. Node identification.
2. Structure identification.
3. Filling the conditional probability table (CPT).

2.1.1 | Node identification

This step identifies the design suitability, expert's knowledge, domain knowledge and experience competence of the development team, and design constrains along with various properties of the system in each phase of the life cycle model for which the BBN model is to be created. For software reliability, the required attributes need to be considered as nodes, as applicable to the considered SDLC phase.

2.1.2 | Structure identification

This step will capture the relationships between the various properties of step 2.1.1 to determine the structure of the BBN model. The following restrictions are made on the directed arcs of the BBN:

1. Edges can only point from lower level nodes to upper level nodes. The reason for this constraint is that failure propagates from lower-level component to a higher-level component.
2. There must be a directed path from each node to the system. We will not consider that node which does not exerts any causal influence either directly or indirectly on the system.

For SDLC, each phase depends on its previous phase. Therefore, edges should be marked keeping in view of consistency and logical flow.

2.1.3 | Filling the CPT

The conditional probability corresponding to each relation must be defined in the CPT in order to complete the BBN model.

2.2 | Prior estimate

The prior estimate represents the present state of knowledge, or present description of uncertainty, about the model parameter before data is being observed. *Informative prior* and the *noninformative* prior estimation are the two methods of choosing prior estimate. In informative prior estimation, the analyst uses his knowledge along with the expert's opinion to construct the prior estimate about the substantive problem based on some other data. The noninformative prior estimation ignores the model parameter and acts as if no prior information or knowledge exists about the parameters before observing the data.

2.3 | Likelihood function

Once the data is observed, the likelihood function is constructed. It is the joint probability function of the data, viewed as function of parameters, treating the observed data as fixed quantities. Let $\boldsymbol{y} = (y_1, \ldots\ldots, y_n)$ have been obtained independently, then the likelihood ($\mathcal{L}$) function is calculated by

$$\mathcal{L}\left(\theta \mid y\right) = p\left(y_1, \ldots, y_n \mid \theta\right) = \prod_{i-1}^{n} p\left(y_1 \mid \theta\right). \tag{1}$$

The likelihood function is used to generate the maximum likelihood estimator and as a key component in Bayesian inference. The main idea is to collect and use the updated information, when and where available. The current data can be taken to derive this function.

2.4 | Posterior estimate

The posterior estimate, $p(\theta \mid y)$, can be defined as the product of the prior estimate and the likelihood function. It is calculated by applying Bayes' theorem as follows:

$$p\left(\theta \mid y\right) = \frac{p(\theta)\mathcal{L}\left(\theta \mid y\right)}{\int p(\theta)p\left(y \mid \theta\right)d\theta} \qquad = \frac{p(\theta)\mathcal{L}\left(\theta \mid y\right)}{p(y)} \propto p(\theta)\mathcal{L}\left(\theta \mid y\right), \tag{2}$$

where $p(y)$ stands for the marginal probability.

The main idea is to accommodate updated information in the prior knowledge to know much insight about the process for more accurate results.

Generally, software errors originate due to ambiguous/incomplete requirements or due to design defects in the requirement phase or the design phase respectively. Such errors propagate further in later stages of SDLC. However, a SCS system rarely meets any failure due to the standard development process being followed and rigorous testing done before it is being implemented in real-time environments. As a result, insufficient failure data is available with the reliability assessor to assess the reliability using the traditional SRGMs. In order to overcome the failure insufficiency problem for SCS reliability evaluation, we propose a Bayesian update model which captures the reliability using the expert's knowledge and the available operational data (if available) to estimate the prior distribution of the system. The prior estimate and the likelihood are combined together using the BBN model to calculate the posterior distribution in each phase of the SDLC.

The BBN model updates the prior estimate by combining the preexisting knowledge (expert's knowledge and the available operational data) with subsequent available information (likelihood) and fixes the evidence based on the observed data.

3 | CASE STUDY

Figure 2 shows the modules and components considered in the reliability model of the DFWCS and the main signals between them. Pressurized water reactor (PWR) has two primary and secondary loops. Primary loop is responsible for the generation of heat from nuclear chain reaction and transferring it to the secondary loop through the steam

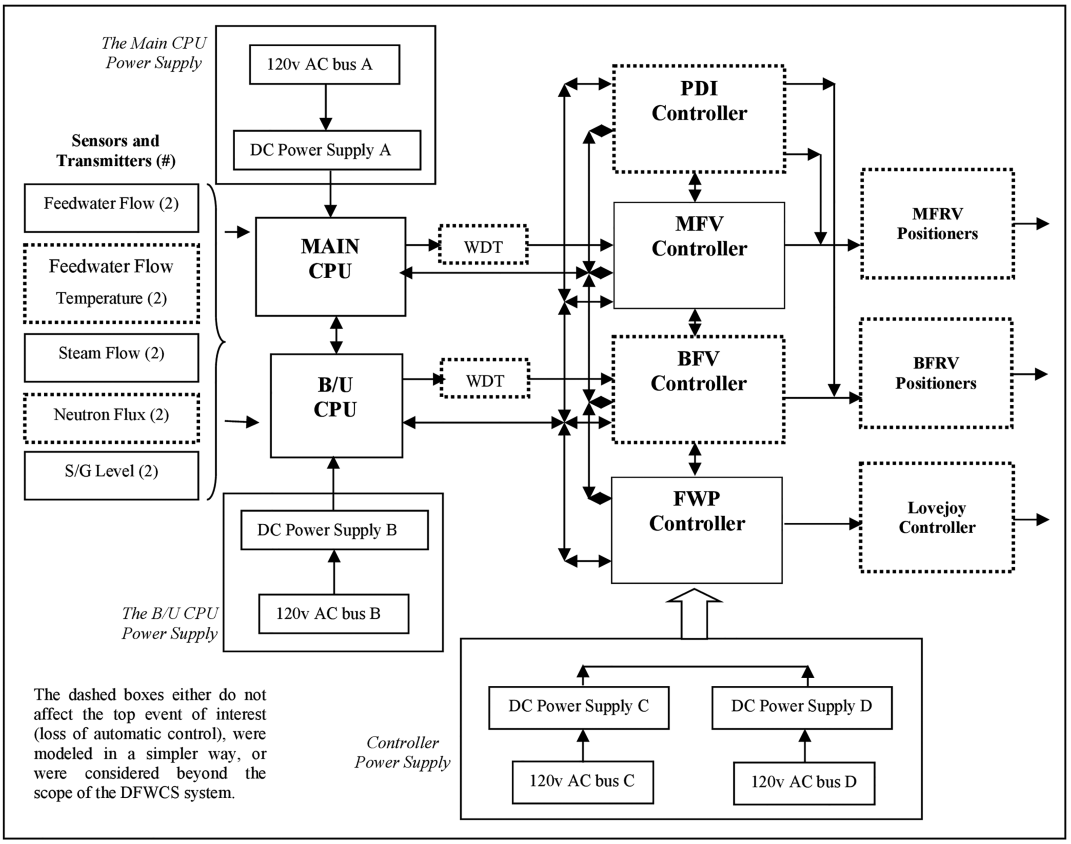

FIGURE 2 Modules of the DFWCS model

generators. The PWR has two secondary loops, each with a DFWCS. The DFWCS consists of sensors, transmitters, two CPU modules, four controller modules (main feedwater valve [MFV], bypass feedwater valve [BFV], feedwater pump [FWP], and pressure differential indicating [PDI]), and associated direct current (DC) power supplies and 120 v alternating current (AC) buses.

The DFWCS sends demand signals to the positioners of the main feedwater-regulating valve (MFRV) and the bypass feedwater-regulating valve (BFRV) and to the turbine controller of the main feedwater pump (MFP). Electrical signals are converted into pneumatic pressure by the positioners that is used to position valves. The PDI controller displays the differential pressure across the MFRV and also serves as manual control station for the MFRV and BFRV. CPU modules and the controller modules consist of a microprocessor and associated components, like analog/digital (A/D) converter and multiplexer (MUX). The feedwater controller maintains ideal water level of ± 2 in with respect to some reference point inside each steam generators. If the water level of the steam generators rises above $+30$ in or falls below -24 in, the feedwater controller is assumed to be failed.

Main and backup CPUs

The main and backup CPUs are considered the brains of the DFWCS. The control algorithms for the feedwater controller are executed on the main CPU (MC) as well as the backup CPU (BC). The FWP, BFV, and MFV use the output signal determined by the PDI through analog control signal along with the failure status signals of the MC or BC. Two CPUs are used for system redundancy. Information like CPU status, deviations, and input signal validity are exchanged by both the CPU. WDT is responsible for monitoring both the CPU functioning and in failure of any of the CPU it sends the status of its associated CPU to the controllers. Each controller uses the status information to determine which of the two demand inputs (from main or backup CPU) to send to the component associated with that controller. The main CPU is assumed to be in control, with the backup CPU operating in tracking mode.

MFV controller

The MFV controller is an interface between the CPU and the MFRV positioners. It is also a manual control station for the MFRV. The CPU provides valve-position demand signals to the MFV controller which, in turn, transmits a demand signal to the MFRV positioners. The MFV controller receives the status of the CPUs from both the CPUs and their associated WDTs.

FWP controller

The FWP controller processes the FWP demand signal when it receives pump demands and CPU status information from the CPUs and sends FWP demand to the turbine speed controller. FWP controller does not send CPU status information to the CPUs and does not have the PDI controller.

BFV controller

The BFV controller is responsible for processing the BFV demand signals. Once the BFV controller receives BFV demands and CPU status information from the CPU, it sends the BFRV demand to the BFRV controller. The BFV controller provides alarms to the plant's annunciator system and the plant computer based on failure information received from the MFV and FWP controllers through Microlink. When the plant is operating in full-power mode, the BFRV is normally closed; even if it fails open, the DFWCS is assumed to accommodate the failure.

PDI controller

PDI controller is normally on standby and does not directly undertake any control function during DFWCS automatic control. It normally displays the differential pressure across the MFRV and has a buffer for holding the outputs of the MFV and BFV controllers until the PDI controller is automatically or manually switched into the control loop.

It is interfaced with nine other SCSs. It uses heterogeneous network protocols: Media Redundancy Protocol (IEC62439-2) in sensors network, Rapid Spanning Tree protocol (IEC62439-1) data acquisition network, and Parallel Redundancy Protocol (IEC62439-3) in human machine network.

4 | APPLICATION OF PROPOSED FRAMEWORK ON CASE STUDY

4.1 | BBN model construction

The BBN model for DFWCS is shown in Figure 3. The BBN model depicts the different phases of the SDLC model along with the evidences for each phase. The evidences in this paper follows a binomial characteristic, ie, each evidence of the BBN model takes two values T or F to represent the status "selected" and "not selected," respectively. Then, for a node with n parents, 2^n conditional probabilities functions can be constructed. The notations for the evidences of the BBN model are shown in Table 1, for convenience. The BBN model also consists of the probability and the conditional probabilities for the independent nodes and the relations, respectively. The CPT for the different phases of the BBN model with respect to their evidences is shown in Table 2 . The probabilities, representing the correctness of the evidences, have been assigned based on the discussions with the respective stakeholders, peer meetings, and experience on developed projects.

4.1.1 | Calculation of the prior estimate

We have used the informative prior estimation for our case study which is qualitative in nature, depending on the discussions, meetings, experiences, and expert judgments. Given the initial probability distribution function of a Bayesian network, the joint probability distribution function for n variables,$Pr(X_1, X_2, ..., X_n)$ can be calculated as

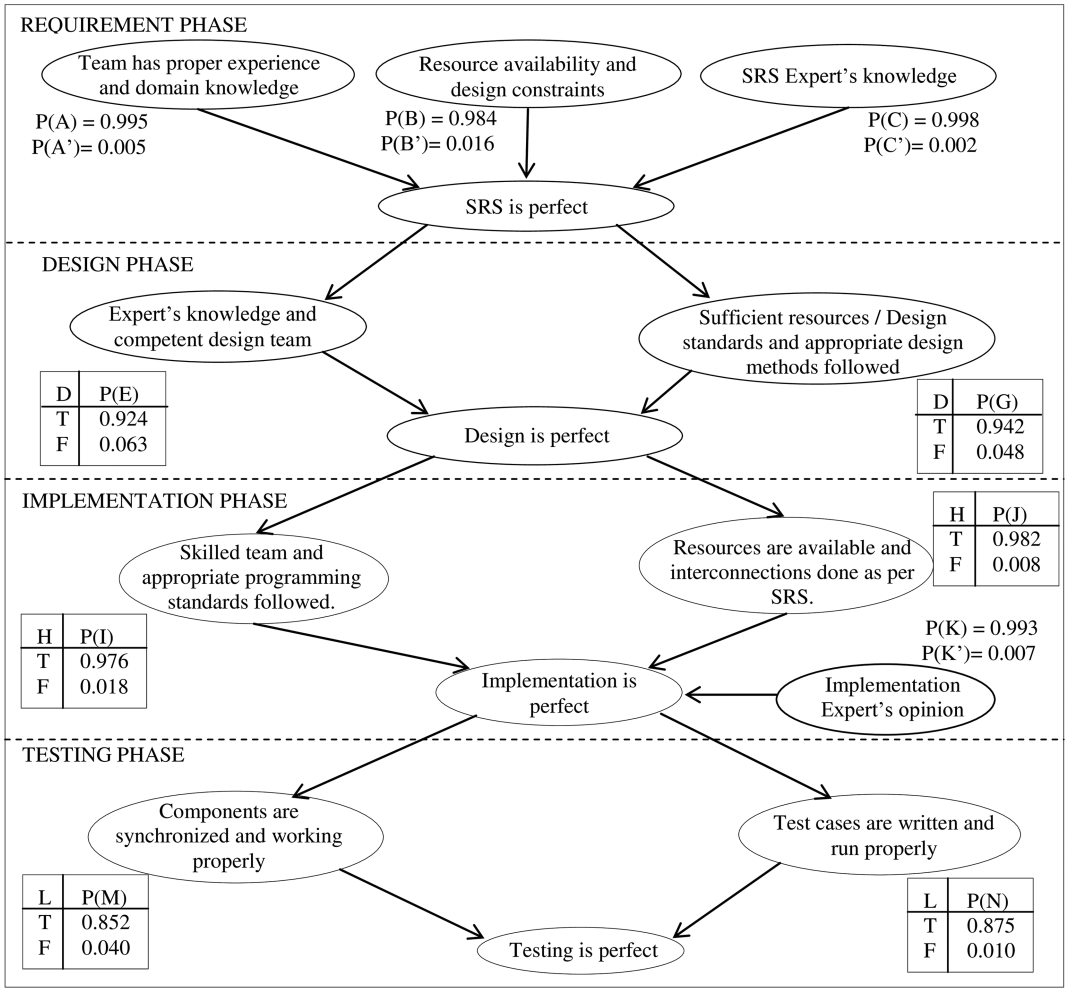

FIGURE 3 Bayesian Network Model for DFWCS model

$$Pr(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} \Pr(X_i | Parent(X_i)).$$

The reliability of the system or software depends on correctness of each phase of the system or SDLC that starts from requirements. Further, the correctness of next phase depends on the correctness of the current phase. Therefore, the BBN model can be utilized to compute the correctness of subsequent phases. The correctness of the system or software will ultimately be judged during testing phase. Hence, it is required to compute the prior correctness of the testing phase. The methodology is explained as follows.

$$\begin{aligned} \mathrm{P(D)} = {} & \mathrm{P(D, A, B, C)} + \mathrm{P(D, A, B, C')} + \mathrm{P\left(D, A, B', C\right)} + \mathrm{P\left(D, A, B', C'\right)} + \mathrm{P\left(D, A', B, C\right)} + \mathrm{P\left(D, A', B, C'\right)} \\ & + \mathrm{P\left(D, A', B', C\right)} + \mathrm{P\left(D, A', B', C'\right)}, \end{aligned} \tag{3}$$

$$\begin{aligned} = {} & \mathrm{P(D|A, B, C)*P(A, B, C)} + \mathrm{P\left(D|A, B, C'\right)*P\left(A, B, C' + P\left(D|A, B', C\right)*P\left(A, B', C\right)\right.\right.} \\ & + \mathrm{P\left(D|A, B', C'\right)*P\left(A, B', C'\right)} + \mathrm{P\left(D|A', B, C\right)*P\left(A', B, C\right)} + \mathrm{P\left(D|A', B, C'\right)*P\left(A', B, C'\right)} \\ & + \mathrm{P\left(D|A', B', C\right)*P\left(A', B', C\right)} + \mathrm{P\left(D|A', B', C'\right)*P\left(A', B', C'\right)}. \end{aligned} \tag{4}$$

4.1.2 | Prior for requirement phase

The prior probability for the Requirement Phase can be calculated as follows: Equations (3) and (4) need to be inserted here.

TABLE 1 Notations for the evidences of the BBN model

A	Team has proper experience and domain knowledge
B	Resource availability and design constraints
C	SRS expert's knowledge
D	SRS is perfect
E	Expert's knowledge and competent design team
G	Sufficient resources/design standards and appropriate design methods followed
H	Design is perfect
I	Skilled team and appropriate programming standards followed
J	Resources are available and interconnections done as per SRS.
K	Implementation expert's opinion
L	Implementation is perfect
M	Components are synchronized and working properly
N	Test cases are written and run properly
O	Testing is perfect

Using the values of CPT, given in Table 2, and the probability values of the evidences A, B, and C from Figure 3, we can calculate the prior probability for the requirement phase. Since, evidences A, B, and C are independent of each other, their joint probability is simply the product of their probabilities. So, from (4), the value of $P(D)$ can be calculated as

$$\begin{aligned} \mathrm{P(D)} = {} & (0.999*0.995*0.984*0.998) + (0.892*0.995*0.984*0.002) + (0.884*0.995*0.016*0.998) \\ & + (0.830*0.995*0.016*0.002) + (0.720*0.005*0.984*0.998) + (0.200*0.005*0.984*0.002) \\ & + (0.202*0.005*0.016*0.998) + (0.001*0.005*0.016*0.002) \Rightarrow \mathrm{P(D)} \\ & = 0.996. \end{aligned} \tag{5}$$

4.1.3 | Prior for design phase

The simplified BBN for the design phase is represented in Figure 4. Figure 4(i) is the Bayesian network model (a sub graph of the BBN model of Figure 3) for design phase. In Figure 4(ii), node D subsumes the parent nodes A, B, and C. In Figure 4(iii), nodes E and G have been clustered together to form node (E + G). The new CPT for the node (E + G) is shown in Table 3. Using the new CPT values obtained in Table 3 and the probability value of node D, the prior probability for node H can be calculated.

Similarly, the prior probabilities of the design, implementation and testing phases can be computed. The computed prior probabilities for all the four phases of the BBN model are summarized in Table 4.

4.2 | Calculation of the likelihood

The operational profile data of 3 years for DFWCS is continuously logged into a computer based system, which can be used to calculate its reliability. The calculated reliability from real data is known as likelihood. For DFWCS, the entire input domain is partitioned into subdomains, hence Brown and Lipow input domain model[25] can be applied to find its likelihood (reliability from real data) also called "the operational usage reliability," which is given by the following equation:

TABLE 2 CPT for the different relations of the BBN Model.

CPT for Requirement Phase			
A	**B**	**C**	**P(D)**
T	T	T	0.999
T	T	F	0.892
T	F	T	0.884
T	F	F	0.830
F	T	T	0.720
F	T	F	0.200
F	F	T	0.202
F	F	F	0.001
CPT for Implementation Phase			
I	**J**	**K**	**P(L)**
T	T	T	0.996
T	T	F	0.823
T	F	T	0.783
T	F	F	0.624
F	T	T	0.778
F	T	F	0.428
F	F	T	0.398
F	F	F	0.090
CPT for Design Phase			
E	**G**		**P(H)**
T	T		0.992
T	F		0.789
F	T		0.578
F	F		0.080
CPT for Testing Phase			
M	**N**		**P(O)**
T	T		0.993
T	F		0. 605
F	T		0.420
F	F		0.082

$$\mathcal{L} = 1 - \sum_{i=1}^{m} \left(\frac{f_i}{n_i}\right) P(E_i), \tag{6}$$

where
$P(E_i)$ is the probability of the specified operational input vector selected from each equivalence class.
n_i is the number of runs for test cases corresponding to each equivalence class.
f_i is the number of failed test cases from each equivalence class.
m is the number of equivalence class.
The steps executed in order to assess the likelihood based on the operational profile data is summarized below.

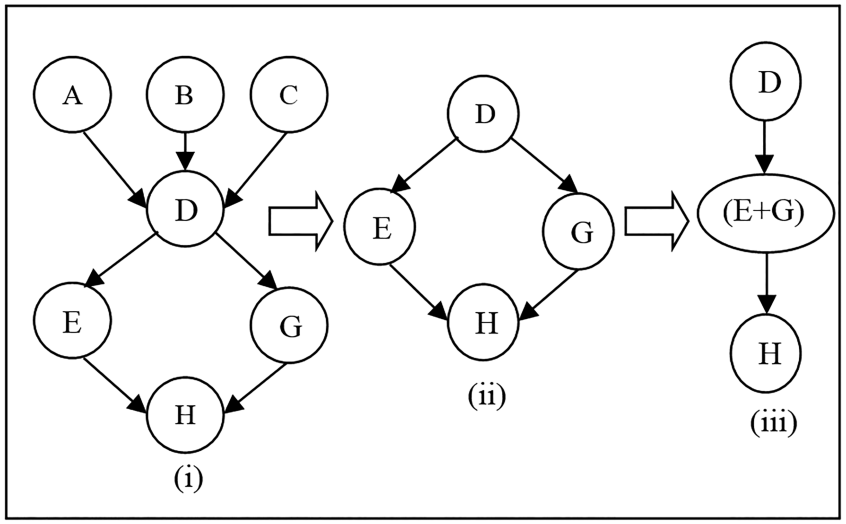

FIGURE 4 Simplified Bayesian belief network (BBN) for the design phase.

TABLE 3 CPT for the node (E+G)

	E+G			
D	**TT**	**TF**	**FT**	**FF**
T	0.870	0.053	0.071	0.004
F	0.003	0.059	0.045	0.892

TABLE 4 Prior probability for the four phases

Phase	Prior Probability
Requirement phase	0.996
Design phase	0.943
Implementation phase	0.950
Testing phase	0.900

Determine the operational profile data as follows:

1. Partitioning of the input domain into equivalence class along with assignment of appropriate operational probability to them.
2. Define the errors.
3. Select and run a set of test cases corresponding to each class.
4. Assess the reliability.

For the estimation of the likelihood of DFWCS, we take the operational data of 3 years as shown in Table 5. It should be noted that the sum of the $P(Ei)s$ is not equal to 1 because only significant equivalence classes has been considered. The likelihood ($\mathcal{L}$) of DFWCS is computed using equation (6).

TABLE 5 Likelihood calculation using Brown and Lipow model

Equivalence class	$P(E_i)$	n_i	f_i	$P(E_i)\left(\frac{f_i}{n_i}\right)$
MFV state	0.10	20	1	0.0050
BFV state	0.10	20	1	0.0050
FP state	0.20	20	3	0.0300
MFV - FP state	0.05	30	2	0.0033
BFV - FP state	0.05	30	2	0.0033

So,

$$\sum_{i=1}^{m}\left(\frac{f_i}{n_i}\right)P(E_i) = 0.0466. \qquad (from\ Table\ 5)$$

Using equation (6),

$$\therefore \mathcal{L} = 1 - 0.0466 \Rightarrow \mathcal{L} = 0.9534 \approx 0.953. \tag{7}$$

The marginal probability of "testing is perfect" node in the presence of nodes M and N can be calculated as[26]

$$\mathrm{P(O)} = \mathrm{P(M)P(O|M)} + \mathrm{P(N)P(O|N)} = (0.852{*}0.605) + (0.875{*}0.420) = 0.882 \therefore P(O) = 0.882. \tag{8}$$

4.3 | Calculation of the posterior

Bayes' theorem is used in conjunction with prior information and current data (likelihood) to provide updated reliability estimates.[26] The updated reliability value, $R^{Post}_{\mathrm{DFWCS}}$, for testing phase can be calculated using Bayes' theorem, as given in equation (2). The values of the prior probability of testing phase is taken from Table 4, the likelihood ($\mathcal{L}$) from equation (7), and the marginal probability from equation (8). The updated reliability or the posterior reliability is calculated by dividing the product of the prior estimate and the likelihood to that with the marginal probability value of testing is perfect node. So, from Table 4 and equations (7) and (8),

$$Prior\ estimate = 0.900$$

$$likelihood\ (\mathcal{L}) = 0.953$$

$$marginal\ probability = 0.882.$$

Using Bayes' theorem,

$$R^{Post}_{\mathrm{DFWCS}} = \frac{Prior\ estimate{*}likelihood}{marginal\ probability} \therefore R^{Post}_{\mathrm{DFWCS}} = 0.900{*}0.953/0.882 \Rightarrow R^{Post}_{DFWCS} = 0.972 \tag{9}$$

The new information has provided us revised reliability estimate of the DFWCS. It should be noted that, as and when new and sufficient operational profile data become available, this posterior estimate should be treated as prior estimate, which should be updated by new operational profile data. It should be a continuous process to improve the accuracy of the results.

5 | RESULTS AND VALIDATION

The prior reliability of DFWCS has been computed by development process that includes computing prior reliabilities at each phase of the system development life cycle and starting from system requirements to the system testing. Bayesian model has been used to infer the reliability at the next phase, given the reliability at the previous phase and using knowledge of required resources and expert's judgment. The reliability from real time data for DFWCS for a period of 3 years is 0.953, as given by equation (7). This reliability of the DFWCS is estimated using operational profile data given in Table 5 using Brown and Lipow input domain model[25] and is called the likelihood ($\mathcal{L}$). The reliability computed using Bayes' theorem is 0.972, as given in equation (9). This is known as posterior reliability.

Comparing the updated reliability value and likelihood reliability of the DFWCS, we get

$$\mathrm{R}^{\mathrm{Diff}}_{\mathrm{DFWCS}} = \mathrm{R}^{\mathrm{Post}}_{\mathrm{DFWCS}} - \mathcal{L},$$

where R^{Diff}_{DFWCS} is the difference between the updated reliability value also known as the posterior reliability and the operational usage reliability also known as the likelihood.

$$\Rightarrow R_{DFWCS}^{Diff} = 0.972 - 0.953 \therefore R_{DFWCS}^{Diff} = 0.019. \tag{10}$$

Therefore, error percentage can be computed as

$$\text{error}\% = \frac{R_{DFWCS}^{Diff}}{\mathcal{L}} \times 100 = \frac{0.019}{0.953} \times 100 = 1.994\% \therefore \text{Accuracy} = 100 - \text{error}\% = 100 - 1.994 \Rightarrow \textit{Accuracy} = 98.006\%. \tag{11}$$

Equation (11) shows that our method gives the reliability prediction with accuracy of 98.006%, which demonstrates the validity of our proposed software reliability prediction model.

6 | CONCLUSIONS

Software systems generally fail due to deficiency in requirements or design. Standards are extensively followed for the development of SCS or SCCS in each phase of SDLC. Therefore, failure of such systems is very rare. Number of failures and failure rate are the important parameter of reliability estimation models. Therefore, these models fail in case of SCS due to insufficiency of number of failures. Therefore, in this paper, we devised an approach to predict the reliability using development methodology of SCS using Bayesian approach. The method utilizes Bayes' theorem and the conditional probability of different evidences in each phase of SDLC for reliability estimation of the overall system. The Bayesian approach requires the estimation of the priors and the likelihood. We have used the expert opinion for estimating the reliability priors at different stages of SDLC to derive the prior reliability, which is in testing phase. The approach has been validated on 12 SCS or SCCS and demonstrated on DFWCS system of NPP using the Brown and Lipow model by comparing the estimated reliability, known as likelihood, value from the operational profile data of 3 years. The result shows that the proposed approach is capable to predict the reliability of SCS with an accuracy of 98.006%, which is quite rewardable. The approach is useful for other than SCS systems as well.

ORCID

Pramod Kumar https://orcid.org/0000-0002-5025-9956

REFERENCES

1. Singh LK, Rajput H. Dependability analysis of safety critical real-time systems by using Petri nets. *in IEEE Transactions on Control Systems Technology*. 2017;99:1-12.
2. Singh LK, Rajput H. Ensuring safety in design of safety critical computer based systems. *Annals of Nuclear Energy, Elsevier*. 2016;92:289-294.
3. Kumar V, Singh LK, Tripathi AK, Singh P. Safety analysis of safety-critical systems using state-space models. *in IEEE Software*. 2017;34 (4):38-47.
4. Singh LK, Vinod G, Tripathi AK. Design verification of instrumentation and control systems of nuclear power plants. *IEEE Trans Nucl Sci*. Apr. 2014;61(2):921-930.
5. Febrero F, Calero C, Moraga MÁ. A systematic mapping study of software reliability modeling. *Information and Software Technology, Elsevier*, Vol. 2014;56:839-849.
6. Kumar, Vinay; Singh, Lalit; Tripathi, Anil: "*Reliability analysis of safety-critical and control systems: a state-of-the-art review*", IET Software, 2017, DOI: https://doi.org/10.1049/iet-sen.2017.0053. IET Digital Library, http://digital-library.theiet.org/content/journals/10.1049/iet-sen.2017.0053
7. Singh L, Rajput H, Vinod G, Tripathi AK. Computing transition probability in Markov chain for early prediction of software reliability. *Quality & Reliability Engineering International*. April 2016;32(3):1253-1263.
8. Singh LK, Vinod G, Tripathi AK. Early prediction of software reliability: a case study with a nuclear power plant system. *IEEE Computer*. 2016;49(1):52-58.
9. Kang HG, Lim HG, Lee HJ, Kim MC, Jang SC. Input-profile-based software failure probability quantification for safety signal generation systems. *Reliability Engineering & System Safety*. 2009;94(10).
10. Eom H-s et al. V&V-based remaining fault estimation model for safety-critical software of a nuclear power plant. *Annals of Nuclear Energy*. 2013;51:38-49.

11. Singh LK, Vinod G, Tripathi AK. Approach for parameter estimation in Markov model of software reliability for early prediction: a case study. *IET Software*. 2015;9(3):65-75.
12. Kumar V, Singh LK, Tripathi AK. Transformation of deterministic models into state space models for safety analysis of safety critical systems: a case study of NPP. *Annals of Nuclear Energy, Elsevier*, Vol. 2017;105:133-143.
13. Kim T, Lee K, Baik J. An effective approach to estimating the parameters of software reliability growth models using a real-valued genetic algorithm. *The Journal of Systems and Software, Elsevier*, Vol. 2015;102:134-144.
14. Sharma K, Garg R, Nagpal CK, Garg RK. Selection of optimal software reliability growth models using a distance based approach. *IEEE Trans Rel*. June. 2010;59(2):266-276.
15. Andreas SA, Chatzis SP. Software defect prediction using doubly stochastic Poisson processes driven by stochastic belief networks. *The Journal of Systems and Software, Elsevier*. 2016;122:72-82.
16. Bai C-G. Bayesian network based software reliability prediction with an operational profile. *The Journal of Systems and Software, Elsevier*, Vol. 2005;77:103-112.
17. Kaaniche M, Kanoun K. The discrete-time hyperexponential model for software reliability growth evaluation. *Proceedings of the Third International Symposium on Software Reliability Engineering*. 1992;64-75.
18. WeiwenPeng H-ZH, Li Y, Zuo MJ, Xie M. Life cycle reliability assessment of new products—a Bayesian model updating approach. *The Reliability Engineering and System Safety*, Elsevier. 2013;112:109-119.
19. Xing Y-Y, Wu X-Y, Jiang P, Liu Q. Dynamic Bayesian evaluation method for system reliability growth based on in-time correction. *IEEE Trans Rel*. June. 2010;59(2):309-312.
20. Li D-C, Chang FM, Chen K-C. Building reliability growth model using sequential experiments and the Bayesian theorem for small datasets. *Expert Systems with Applications, Elsevier*. 2010;37:3434-3443.
21. Yu X, Liu J, Yang Z, Liu X. The Bayesian network based program dependence graph and its application to fault localization. *The Journal of Systems & Software*, Elsevier. 2017. https://doi.org/10.1016/j.jss.2017.08.025
22. Campodonico S, Singpurwalla ND. A Bayesian analysis of the logarithmic-Poisson execution time model based on expert opinion and failure data. *IEEE Trans On Soft Eng*. Sep. 1994;20(9):677-683.
23. Zou B.., et al., "*Reliability analysis of digital instrumentation and control software system*," Progress in Nuclear Energy, (2017), https://doi.org/10.1016/j.pnucene.2017.03.006
24. Kim BUK, Goodman D, Liu MLJ, Li J. Improved reliability-based decision support methodology applicable in system-level failure diagnosis and prognosis. *IEEE Transactions on Aerospace and Electronic Systems*. 2014;50:2630-2641.
25. Brown JR, Lipow M. Testing for software reliability. *Proc. Intl. Conf. Reliable Software. Los Angeles, California*. April 1975;1975:518.
26. Byers JK, Weibe HA. Pocket handbook on reliability. *USA AVS COM Technical Report*. 1975;77-16.

AUTHOR BIOGRAPHIES

Pramod Kumar is pursuing his PhD in Reliability Prediction of Safety-critical systems from the Department of Computer Science and Engineering at the Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand, India. His research interests are reliability, safety, performance and mathematical modeling. Kumar received his M. Tech. in the year 2016 in Information Technology from Birla Institute of Technology, Mesra, Ranchi, Jharkhand, India.

Lalit Kumar Singh is a scientist, level E, at the Nuclear Power Corporation of India, Department of Atomic Energy, Government of India. His research interests are software reliability, dependability, mathematical modeling, and fault tolerance. Singh received a PhD from the Indian Institute of Technology (Banaras Hindu University), Varanasi, India.

Chiranjeev Kumar is a professor in the Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines) Dhanbad, Jharkhand, India. His research interests include Wireless Networks, Software Engineering, and IoT. Kumar received his PhD from University of Allahabad, India in 2006.

How to cite this article: Kumar P, Singh LK, Kumar C. Software reliability analysis for safety-critical and control systems. *Qual Reliab Engng Int*. 2020;36:340–353. https://doi.org/10.1002/qre.2577