The Unnecessity of Assuming Statistically Independent Tests in Bayesian Software Reliability Assessments

Kizito Salako and Xingyu Zhao

Abstract—When assessing a software-based system, the results of Bayesian statistical inference on operational testing data can provide strong support for software reliability claims. For inference, this data (i.e., software successes and failures) is often assumed to arise in an independent, identically distributed (i.i.d.) manner. In this paper we show how conservative Bayesian approaches make this assumption unnecessary, by incorporating one’s doubts about the assumption into the assessment. We derive conservative confidence bounds on a system’s probability of failure on demand (pfdd), when operational testing reveals no failures. The generality and utility of the confidence bounds are illustrated in the assessment of a nuclear power-plant safety-protection system, under varying levels of skepticism about the i.i.d. assumption. The analysis suggests that the i.i.d. assumption can make Bayesian reliability assessments extremely optimistic – such assessments do not explicitly account for how software can be very likely to exhibit no failures during extensive operational testing despite the software’s pfdd being undesirably large.

Index Terms—Conservative Bayesian inference, CBI, dependability claims, independent software failures, operational testing, software reliability assessment, statistical testing

1 INTRODUCTION

Consider a software-based on-demand system subjected to black-box operational testing. During testing, an assessor observes the system – in particular, the software – as it responds to each demand in a random sequence of demands. By noting those demands the software correctly responds to, and those it fails on, the assessor intends to gain enough confidence to support claims of the system being sufficiently reliable.

For example, confidence in the system’s unknown probability of failure on demand (pfdd) – X say – being sufficiently small. A Bayesian approach to gaining such confidence typically requires 2 things of our assessor: i) prior to operational testing, the assessor must scrutinise all evidence related to the system’s operational readiness. Via such probing and the assessor’s domain expertise, the assessor forms beliefs about which ranges of pfdd values are most likely, and which ranges are less so. Examples of evidence include formal analyses of a codebase, the performance of a system during operation [1], the historical performance of similar systems [2], [3], and improvements in software development approaches [4]; ii) the assessor must postulate a statistical model – in essence, a family of stochastic processes, any of which could characterise the occurrence of the system’s successes and failures during operation. These processes should exhibit statistical properties consistent with the assessor’s beliefs.

For the statistical model it is often assumed that software failures and successes are the outcomes of independent, identically distributed (i.i.d.) Bernoulli trials. This is mathematically convenient and guarantees the strong law of large numbers – i.e., operational testing statistics converge almost surely to dependability measures of interest (e.g., pfdd). The i.i.d. assumption can also be reasonable on practical grounds: a typical justification is when demands are rare, and the states/properties of the software and its operational environment are effectively “reset” inbetween these rare demand occurrences. Nevertheless, we must point out that any statistical model used in reliability assessment – including one reliant on the i.i.d. assumption – is necessarily a postulate. One does not (cannot?) know with complete certainty that such model assumptions are valid in practice. The skeptical assessor allows for the possibility that the i.i.d. assumption does not hold, even if this is unlikely. By incorporating their skepticism into the assessment, our assessor can investigate whether their doubts have a significant impact on their confidence in the system. In the best case, their confidence is insensitive to significant departures from
the i.i.d. assumption. But in the worst case, ignoring the slightest failure correlations could lead to seriously misplaced confidence and dangerously optimistic reliability claims.

To aid the skeptical assessor, this work presents a new application of conservative Bayesian inference (CBI) techniques for reliability assessments. The paper’s research contributions are:

1) incorporating a formal notion of “doubting” the i.i.d. assumption into reliability assessments (Section 5);
2) a novel CBI technique that accounts for correlated operational testing outcomes (Sections 4, 5 and Appendices B, C, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TSE.2022.3233802);
3) a theorem that gives conservative posterior confidence bounds on a system’s pdf (Section 5 and Appendices B, D, in the online supplemental material).

The paper’s outline: Section 2 gives critical context, while Section 3 reviews CBI. Section 4 introduces a statistical model for (possibly) dependent system failures/successes. The conservative confidence bounds on pdf in Section 5 are applied in Section 6, and discussed in Section 7. Section 8 concludes the paper.

2 RELATED WORK

2.1 The i.i.d. Assumption in Reliability Assessments

Statistical models with the i.i.d. assumption have a long history of being used in software reliability assessment. Thayer et al.’s [5] model was one of the earliest, used in early works on random testing [6]. Regulatory bodies have recommended using the i.i.d. assumption in reliability assessments when appropriate [7].

But there are reasons to doubt the assumption. For instance, the possibility of “failure clustering”; where a system receives sequences of inputs from its operational environment that cause the system to fail. These inputs form trajectories through a system’s failure region – the subset of inputs that cause system failure. Failure regions can have topologically interesting properties that allow for failure clustering [8], [9]. These observations informed random testing approaches [10] and approaches for assessing systems with recovery block fault-tolerance [11], [12].

2.2 Weakening the i.i.d. Assumption: Statistical Models

Various statistical models that weaken the i.i.d. assumption include: Chen and Mill’s binary Markov chain [13], Goseva-Popstojanova and Trivedi’s Markov renewal process [14] (extended in [15] by Xie et al.), and Bondavalli et al.’s Markov model [16] with benign-failure states. Classical statistical inference produces “point estimates” for these models’ parameters, using only data from operational testing. Such estimates do not reflect an assessor’s uncertainty about whether the i.i.d. assumption holds or not. Indeed, fitted parameter values imply that the models either exhibit dependence, or they do not – there is no room for uncertainty here. In contrast, Bayesian inference – our preferred approach – allows for such uncertainty. In this paper, within a Bayesian framework, we model the system’s failure process as a Markov chain introduced by Klotz [17]. The Klotz model predates, is consistent with, and has (at most) as many states/parameters as, the models in [13], [14].

Another advantage of Bayesian approaches is the assessor’s beliefs (about the unknown pdf) are explicitly accounted for in the assessment; beliefs that are justified by various forms of reliability evidence. And, although models of dependent system failures/successes have been developed, none of the assessment approaches using these models provide demonstrably conservative statistical support for the skeptical assessor. Particularly when such support is justified by operational testing and other forms of reliability evidence. To the best of our knowledge, ours is the first approach to guarantee conservatism in the face of i.i.d. uncertainty.

2.3 (Conservative) Bayesian Methods for Assessments

Bayesian methods have been applied in various assessment scenarios, e.g., [2], [3] involve hierarchical models, while [18], [19], [20], [21] all use families of Beta prior distributions. More recently, conservative Bayesian inference (CBI) methods have been developed to: i) address the usual challenge of eliciting a suitable prior distribution from an assessor – a prior that captures all, and only all, of an assessor’s beliefs/views about the pdf; ii) give support for the most pessimistic reliability claims allowed by the reliability evidence. Thereby, CBI prevents dangerously optimistic claims.

Bishop et al. [22] introduced CBI, illustrating its use in assessing safety-critical systems. Povyakalo et al. [23] use CBI to obtain the smallest probability of the system surviving future demands, and Salako [24] applies CBI to assessing binary classifiers. While Flynn et al. [25], [26] apply CBI in the assessment of autonomous vehicle safety. Littlewood et al. [27] show how CBI supports dependability claims when evidence suggests a new system is an “improvement” over an older system it replaces. Salako et al. [28] extend this work, by considering more general “improvement arguments” for a wider range of assessment scenarios. Our application of CBI differs from these in 2 important ways: it allows for dependent testing outcomes during operational testing, and it incorporates skepticism of the i.i.d. assumption into assessments.

CBI methods are closely related to robust Bayesian analysis [29], [30], [31], [32], which studies how the results of Bayesian inference are impacted by uncertainty about the inputs to inference – inputs such as the prior distribution and the statistical model. In particular, Lavine [31] outlines methods that reveal how uncertainty about the statistical model (specifically, about the so-called sampling distribution) impacts inference. This uncertainty is represented by a suitably general joint prior distribution over the family of stochastic processes defined by the statistical model. Subject to constraints on this prior, algorithms give the largest and smallest values for posterior measures of interest – e.g., posterior expectations. Our use of CBI parallels the statistical techniques of Lavine, but applied to reliability assessments. Also, Lavine considers likelihoods consisting of products of
the same functional form, while we do not (in order to weaken the i.i.d. assumption).

Draper [33] also tackles the problem of statistical model uncertainty in Bayesian inference, but offers an alternative solution. If one is uncertain about a model’s assumptions – specifically, assumptions that constrain the structural/functional forms of the related family of stochastic processes – one could replace the model with an expanded model. This expanded model encompasses all of the stochastic processes defined under the original model, as well as other stochastic processes that violate the assumptions in question. A suitable prior distribution over this expanded model has to be defined. Draper argues for this Bayesian hierarchical model as a way of addressing model uncertainty. We concur, and further argue for the inference to be conservative; our results are guaranteed to be conservative, while Draper’s are not.

For uncertainty about finitely many alternative models, Pericchi et al. [34] use discrete prior distributions over these alternatives. In principle, this is a hierarchical model akin to Draper’s approach, but in more abstract terms within the robust Bayes framework. Our results lie at the intersection of Pericchi et al. and Draper’s ideas, within a reliability assessment context.

2.4 On-Demand versus Continuously Operating Software

This work focuses on assessing on-demand systems: i.e., systems that do not operate continuously, taking action only when certain operating conditions (i.e., demands) arise [35], [36]. Reliability assessments for such systems can use “discrete-time” statistical models (e.g., Bernoulli processes) with appropriate reliability measures (e.g., pfd). Contrast this with continuously operating software [37]; for assessing these systems, it is more appropriate to employ “continuous-time” statistical models (e.g., non-homogeneous Poisson processes) and consider reliability measures like failure-rates. Software reliability growth models (SRGMs) are an extensive family of stochastic processes used in predicting the future reliability of continuously operating, evolving software (e.g., software with bugs that are discovered and fixed overtime). If bug fixing is successful without introducing new software bugs then, ceteris paribus, the software becomes more reliable with each fix; i.e., reliability “grows”. Singpurwalla and Wilson [38] give an overview of early SRGMs, while Miller [39] details a unified mathematical characterisation of large SRGM subfamilies. Also, see Bergman and Xie’s review of early Bayesian SRGMs [40].

3 REVIEW: BAYESIAN RELIABILITY ASSESSMENT

Statistical inference for reliability claims comes in different flavours. The classical “frequentist” confidence statement, e.g., 95% confidence in a pfd bound b, typically means that with a sufficiently large number of i.i.d. tests, there is no more than a 5% chance that the software succeeds on all the tests despite having a pfd worse than b. While the Bayesian approach, instead, treats pfd as a random variable, with a “prior” probability distribution representing an assessor’s evidence-based beliefs about the pfd before operational testing. The assessor updates their beliefs (via Bayes Theorem) upon seeing testing evidence. This yields a “posterior” distribution. Reliability claims can be made using this posterior; claims that reflect the assessor’s updated judgements. For example, after seeing a sufficiently large number of successful tests, the assessor’s Bayesian “confidence” in a pfd bound b – i.e., their conditional probability of the pfd being less than b – is 95%.

Specifically, we recall the standard Bayesian approach to assessing an on-demand system [7], [41], [42], in which the i.i.d. assumption is adopted. An i.i.d. Bernoulli process represents the stochastic failure behaviour of the system’s software. We denote by X the system’s unknown pfd due to software failures. According to the operational profile [43], n demands are randomly submitted to the software and no failures are observed (this is the usual requirement when assessing a safety-critical system using operational acceptance testing). Let b be the required upper bound on pfd. Bayesian inference then gives an assessor’s posterior confidence in b after observing n tests without failure:

\[ P(X \leq b | n \text{ demands without failure}) = \frac{P(X \leq b, n \text{ demands without failure})}{P(n \text{ demands without failure})} \]  

In practice, Bayesian reliability assessments require that one specifies a prior distribution representing one’s beliefs about the possible values of pfd. CBI relaxes this by requiring only a partial specification of the prior distribution, when such specifications can be justified by evidence obtained prior to operational testing. Such partial specifications – so-called prior knowledge (PK) – take various forms. Most notably, the form of confidence bounds; e.g., being 90% confident that the pfd is no greater than 10^−3, partly because IEC 61508 Safety Integrity Level 3 [44] was a strict requirement in the system’s development. When an assessor articulates their beliefs as PKs, there is an infinitely large set of all prior distributions that satisfy these PKs. CBI then determines the worst support the priors in D can give for a reliability claim; e.g., the smallest posterior confidence (1) an assessor can have

\[ \inf_D P(X \leq b | n \text{ demands without failure}) \]  

The solution of (2) identifies a prior with posterior confidence that is the infimum value; no other prior that satisfies the PKs can give a smaller value for posterior confidence (1). In this sense, CBI results are conservative. This “worst case” prior encodes within it the most conservative assessor beliefs consistent with the PKs.

4 A STATISTICAL MODEL OF TESTING OUTCOMES

The Klotz model [17] is a stationary random process consisting of possibly dependent Bernoulli trials. As a model of a system’s failure process it generalises the i.i.d. Bernoulli process used in reliability assessments. During operational testing, a random sequence of demands is submitted to the system. On each demand, the system either successfully

\[ P(\text{failure-free operation} | n \text{ demands without failure}) \]
handles the demand or fails. So, we have a sequence of random variables $T_1, \ldots, T_n$, each taking the values 0 or 1, corresponding to success or failure, respectively. In this paper we follow the usual convention of upper-case letters for random variables and lower-case for their realisations.

The Klotz model is characterised by a “frequency” parameter $x$ and a “dependence” parameter $\lambda$. Here, $x$ is the system’s pdf while $\lambda$ is the probability that a failure is followed by another failure. So, $P(T_i = 1) = P(T_{i-1} = 1) = x$ and $P(T_i = 1 | T_{i-1} = 0) = \frac{(1-\lambda)x}{\lambda}$ for $i = 2, \ldots, n$ (see Appendix A in the online supplemental material), which yields Fig. 1. The transitions in Fig. 1 need to lie between zero and one, so

$$0 \leq x < 1, \quad \max\{0, (2x - 1)/x\} \leq \lambda \leq 1.$$  \hfill (3)

Inequalities (3) define a subset $R$ of the unit square (Fig. 2a).

Suppose the system succeeds on all $n$ demands during operational testing. For $(x, \lambda) \in R$, the Klotz likelihood function gives the probability of observing this sequence of successes

$$L(x, \lambda; n) := (1 - x)\left(1 - \frac{(1-\lambda)x}{1-x}\right)^{n-1}.$$  \hfill (4)

Different values of $(x, \lambda)$ can alter the dependence structure and functional form of the Klotz likelihood: i) $x = \lambda$ is the special case of independent testing outcomes, with $L(x, x; n) = (1 - x)^n$; ii) when $\lambda > x$, successes (and failures) tend to cluster during testing – i.e., positive dependence. In the extreme, $\lambda = 1$ and we have $L(x, 1; n) = (1 - x)^n$; iii) lastly, when $\lambda < x$, successes and failures tend to alternate more often – i.e., negative dependence. In the extreme, $x = \frac{1}{2}$ and we have $L\left(\frac{1}{2}, \lambda; n\right) = 0$.

The assessor’s uncertainty about $x$ and $\lambda$ is captured by a joint prior distribution of $(X, \Lambda)$ over $R$. The assessor’s posterior confidence (1) is

$$P(X \leq b \mid n \text{ demands without failure}) = \frac{\mathbb{E}[L(X, \Lambda; n)\mathbf{1}_{X \in [0,b]}]}{\mathbb{E}[L(X, \Lambda; n)]},$$  \hfill (5)

where the indicator function $\mathbf{1}_{x \in S}$ equals 1 if $x \in S$, and is 0 otherwise.

In Section 5, via CBI (i.e., constrained nonlinear optimisation of (5) in the vein of (2)), we derive conservative posterior confidence bounds on pdf.

5 CONSERVATIVE CONFIDENCE (PFD) BOUNDS

We now present conservative posterior confidence bounds on pdf for the skeptical assessor. We begin by formalising the assessor’s beliefs as a collection of 4 constraints called “prior knowledge”. These PKs only weakly specify the joint prior distribution of $(X, \Lambda)$. First, the assessor’s prior distribution of $X$ is continuous.

Prior Knowledge 1 (continuous prior distribution of $X$). A density function $f(x)$ gives the assessor’s prior confidence in pdf bound $u$ – i.e., $P(X \leq u) = \int_0^u f(x) \, dx$ for all $u \in [0, 1]$.

Second, the assessor does not rule out negatively or positively correlated testing outcomes (see Fig. 2a).

Prior Knowledge 2 (confidence in negative correlation). $\phi_1 \times 100\%$ confident that the outcomes of successive tests are negatively correlated, i.e., $P(\Lambda < X) = \phi_1$.

Prior Knowledge 3 (confidence in positive correlation). $\phi_2 \times 100\%$ confident that the outcomes of successive tests are positively correlated, i.e., $P(\Lambda > X) = \phi_2$.

Third, the assessor is relatively confident that the i.i.d. assumption holds, and bound $b$ is satisfied, but not so confident as to make testing unnecessary. A straightforward extension of the theorem presented here accounts for less confidence.

Prior Knowledge 4 (confidence in bound and independence). For a target system pdf $b$, $\phi_3 \times 100\%$ confident that $P(X \leq b \mid n \text{ demands without failure})$ is solved by the prior in Fig. 2b, since the infimum equals the value of $P(X \leq b \mid n \text{ demands without failure})$ for this prior. The infimum takes the form $\frac{1}{\Pi_{i=0}^n Q_i}$, where $Q$ is

$$\int_b^1 \left( (1-x)^n \mathbf{1}_{x \in [0,c_1 \cap \ldots \cap c_n]} + (1-x)\mathbf{1}_{x \in [c_1, c_2]} \right) f(x) \, dx \left/ \int_0^1 \left( (1-x)^n \mathbf{1}_{x \in [0,c_1 \cap \ldots \cap c_n]} + (1-2x)^{n-1} \mathbf{1}_{x \in [c_1, c_2]} \right) f(x) \, dx \right.,$$  \hfill (6)

and the pdfs $c_1, c_2, c_1, c_2$ are the unique values of $r, s, v, w$, respectively, that solve

$$\arg\min_{0 \leq r<s \leq b} \left| g_r(r) - g_r(s) \right|, \quad \arg\min_{b<v<w \leq 1} \left| g_v(w) - g_w(w) \right| \quad \text{s.t.} \quad g_l(0) \leq g_l(r), \quad g_l(b) \leq g_l(s), \quad g_s(b) \leq g_s(v), \quad g_s(1) \leq g_s(w), \quad \int_r^s f(x) \, dx = \phi_1, \quad \int_v^w f(x) \, dx = \phi_2,$$

where $g_l(r) = \frac{\mathbb{E}[L(X, \Lambda; n)\mathbf{1}_{X \in [r,b]}]}{\mathbb{E}[L(X, \Lambda; n)]}$.
for \( g_l : [0,1/2] \to [0,1] \) and \( g_u : [0,1] \to [0,1] \) defined as
\[
g_l(x) = (L(x,x; n) - L(x,0;n)) \mathbb{1}_{x \in [0,1/2]},
g_u(x) = (L(x,1;n) - L(x,x; n)) \mathbb{1}_{x \in [0,1]}.
\] (7)

Numerical estimates for \( c^1, c^2, c^1, c^2 \) may be computed using “root-finding” algorithms such as that in Appendix D.

If the assessor has no doubts about the i.i.d. assumption, this CBI result is given by traditional Bayesian inference. That is,

**Corollary** Let \( Q \) be given by (6) in the theorem, when \( \phi_1 = \phi_2 = 0 \). Then the infimum is given by traditional Bayesian inference, using the prior density \( f(x) \). That is,
\[
\inf_{P} P(X \leq b\mid n \text{ demands without failure}) = \frac{\int_0^b (1 - x)^n f(x) \, dx}{\int_0^b (1 - x)^n f(x) \, dx}.
\] (8)

However, if the assessor begins operational testing with beliefs captured by PKs 1, 2, 3 and 4 – i.e., beliefs that only partially specify a joint prior distribution of \((X, \Lambda)\) – then the CBI theorem identifies a joint prior that is consistent with these beliefs, is unique up to zero probability events, and that gives the smallest upper-bound confidence in the pdf upper-bound \( b \) (see Fig. 2b).

This CBI prior assigns probability density only to the indicated thick black line segments in \( R \); the rest of \( R \) has zero probability\(^5\) (Appendix B). For example, prior to testing, the conservative assessor expects negatively, or positively, dependent test outcomes with probabilities \( \phi_1 = P(c^1_l \leq X \leq c^2_l, \Lambda = 0) = \int_{c^1_l}^{c^2_l} f(x) \, dx \) and \( \phi_2 = P(c^1_u \leq X \leq c^2_u, \Lambda = 1) = \int_{c^1_u}^{c^2_u} f(x) \, dx \), respectively. Or consider the probabilities of the 2 shaded rectangular regions in Fig. 2c. The non-zero contributions to these probabilities come from the thick black line segments that intersect these regions; i.e., \( P(x_1 \leq X \leq x_2, \lambda_1 \leq \Lambda \leq \lambda_2) = \int_{x_1}^{x_2} f(x) \, dx \) and \( P(x_3 \leq X \leq x_4, \lambda_3 \leq \Lambda \leq 1) = \int_{x_3}^{x_4} f(x) \, dx \). The CBI prior Fig. 2b encodes conservative beliefs. In particular, the pdf values \( c^1, c^2, c^1, c^2 \) indicate where (in \( R \)) doubts in the i.i.d. assumption should be placed for conservative posterior confidence. The \( R \) locations between \((c^1,0)\) and \((c^2,0)\) along the \( \lambda = 0 \) edge represent statistical dependence that is unlikely to produce failure-free testing if the unknown pdf actually satisfies the bound. And if the pdf does not satisfy the bound, the locations between \((c^1,1)\) and \((c^2,1)\) along \( \lambda = 1 \) represent dependence likely to produce failure-free testing. Note that the assessor has been unable to rule out these extreme beliefs prior to testing, since these beliefs are consistent with the PKs.

These beliefs depend on the number \( n \) of required failure-free tests; the \( c^l \)'s become smaller as \( n \) becomes bigger. Because, the \( c^l \)'s are defined with respect to the stationary points of the theorem’s \( "g" \) functions – themselves dependent on \( s \). Fig. 3a plots how the \( x \) values for these stationary points tend to zero as \( n \) increases. Fig. 3b shows the consequence of this – \( c^l \) decreases towards zero and \( c^u \) decreases towards a non-zero value dependent on \( f(x) \). Fig. 3c tells a similar story – \( c^l \) decreases towards the bound \( 10^{-4} \), and \( c^u \) towards a non-zero value dependent on \( f(x) \). This asymptotic behaviour is reasonable: the greater the amount of failure-free operation required during testing, the smaller the pdf is expected to be for a system that performs this well (even when one is doubtful of the tests being i.i.d.).

Informally, beliefs in i.i.d. tests “lie on the boundary” of the set of conservative beliefs that express doubts about i.i.d. tests, and confidence from beliefs in i.i.d. tests (8) is well-approximated by conservative confidence from the theorem. Indeed, by the dominated convergence theorem [45], (6) tends to \( \frac{\int_0^b (1-x)^n f(x) \, dx}{\int_0^b (1-x)^n f(x) \, dx} \) as \( \phi_1 \) and \( \phi_2 \) tend to 0 (since \( c^l, c^1 \) tend to \( c^l, c^2 \) respectively).

6.1 Practical Context and Guidance
The theorem gives conservative confidence in a pdf bound \( b \), when an on-demand system is subjected to black-box operational testing. A bound such as \( b = 10^{-4} \); a target pdf used in the assessment of the Sizewell-B nuclear power plant safety...
In particular, the theorem can be used during acceptance testing to check the robustness of confidence (8) to doubts about the i.i.d. assumption; we illustrate how to do this in the rest of this section. When applying statistical techniques like the theorem, one may follow the guidance from Littlewood and Strigini [49] and Lyu et al. [37] on performing statistical testing. See Parnas et al. [48] for additional discussion on evaluating safety-critical software, including random test-case (e.g., demand) selection. For nuclear safety applications in particular, [7], [50] give guidance on reliability assessment using statistical techniques.

More generally, the theorem can be applied in any black-box testing phase where failure-free operation can be used to gain confidence (i.e., (8)) in the software. Such testing would typically involve subjecting (some part of) the software to a large number of randomly generated demands in a simulated environment. For example, in accordance with integrity level 4 (see IEEE 1012:2016 [51]), one may apply the theorem during component, or integration, testing phases for safety-critical software. For nuclear safety-critical software testing phases, see also IEC 60880 [52].

6.2 Examples: Prior Beliefs and Confidence in a Bound

The analyses in the rest of this section show how: i) confidence based on the i.i.d. assumption can be very optimistic as failure-free tests accumulate; ii) some forms of doubt about the i.i.d. assumption significantly impact confidence, while other forms do not; iii) surprisingly, failure-free testing can eventually undermine confidence in the system satisfying the bound.

The PK1 density $f(x)$ can be from any family of continuous distributions over the interval [0,1]. Beta densities are often used in practice [7], [18], [20]. Consider 3 alternative $f(x)$; the Beta distributions in Fig. 4a with parameters/properties in Table 1. Let these represent prior beliefs of 3 assessors that differ in how confident they are that the system satisfies the bound. Suppose the assessors are a little skeptical of the tests being i.i.d. (e.g., $\phi_1 = \phi_2 = 0.05$). Fig. 4b shows how posterior confidence evolves as operational testing evidence mounts. For each Beta prior, the posterior confidence (8) under an i.i.d. assumption is plotted against the CBI posterior confidence $\frac{1}{1-Q}$ for our skeptic. In all cases, confidence from assuming independence is initially comparable to conservative confidence – the relevant pair of curves for each Beta prior almost overlap initially.

However, as more failure-free testing is observed – i.e., as $n$ grows – “i.i.d.”-based confidence grows and tends towards certainty. While conservative confidence grows more slowly, reaches a maximum, and then tends towards zero. Appendix C, available in the online supplemental material, proves this zero limiting behaviour will occur whenever an assessor allows for the possibility of positively correlated tests (i.e., $\phi_0 > 0$). In the limit of large $n$, “i.i.d.”-based confidence can be the most optimistic confidence can be, while still remaining consistent with an assessor’s informed views/beliefs about the unknown $pfd$.

If the evidence available to an assessor before testing justifies being very confident in the bound, then the initial closeness between the “i.i.d.”-based posterior confidence and the confidence given by CBI can continue for longer before ultimately diverging. In Fig. 4b the pair of curves for the Beta(0.1,1000) prior – i.e., for the very confident assessor – stay closer together for longer, compared to the curves for the least confident assessor with prior Beta(2,20000). The greater the prior confidence in the bound, the greater the posterior confidence when testing begins (i.e., small $n$).

6.3 Sensitivity Analysis: How Failure-Free Tests and Skepticism About i.i.d. Tests Impact Confidence

The nature of an assessor’s skepticism determines whether their conservative posterior confidence ultimately grows or shrinks during operational testing. At one extreme, some forms of doubt have no noticeable impact on confidence when no failures are observed during testing. The possibility
(however likely) of negatively correlated tests has no apparent effect on conservative confidence. For example, an assessor might intentionally seek to “stress” the software during testing, by randomly including a disproportionate number of test demands that are thought will likely cause the software’s failure. If stressful demands are adequately interspersed with significantly less stressful ones, one might expect the testing outcomes (i.e., the software’s successes/failures) to exhibit some negative dependence (i.e., non-zero $f_1$) – so a failure is quickly followed by successes, then another failure relatively soon afterwards, etc. However, when no failures occur, Figs. 4c and 4d show that the “rise and fall” of CBI confidence in Fig. 4b is unaffected by varying one’s prior confidence in negative dependence. Intuitively, the more successes occur, the less likely these are from a system undergoing negatively dependent tests.

At the other extreme are positively correlated tests. Figs. 4e and 4f both show that the smaller $\phi_2$ is, the closer conservative confidence gets to the confidence under the i.i.d. assumption. When $\phi_2 = 0$, conservative confidence grows to certainty as the number of successes grows (Appendix C). While the larger $\phi_2$ is, the more conservative confidence becomes. Here, confidence in positive correlations (i.e., large $\phi_2$) may be due to pessimistic reasons for the failure-free tests – i.e., “success clustering” can occur even if the software is unreliable. The tests could be unrepresentatively “easy” for the software to correctly respond to, or the test oracle is incorrect so failures go undetected [53], [54].

So, failure-free testing can undermine one’s confidence in a system’s $pfd$. Such conservatism is not unique to CBI – even with classical inference, confidence bounds can be quite conservative initially, becoming optimistic (compared to the CBI bounds) after many tests. Indeed, using the Klotz likelihood (4) when $\lambda = 1$, the probability of succeeding on all $n$ tests (despite the $pfd$ being worse than $10^{-4}$) is at most $(1 - 10^{-4})^{n} = 0.9999$. That is, a system with a $pfd$ worse than $10^{-4}$ may be almost certain to succeed on all tests if the tests are strongly positively correlated. This bleak result holds for all $n$; so, increasing the number of failure-free tests does not increase one’s confidence in the system.

**TABLE 1**

| $\alpha$ | $\beta$ | $E[X] = \frac{\alpha}{\alpha + \beta}$ | $P(X \leq 10^{-4})$ |
|----------|---------|----------------------------------------|---------------------|
| 2        | 20000   | 0.0001                                 | 0.6                 |
| 1        | 10000   | 0.0001                                 | 0.63                |
| 0.1      | 1000    | 0.0001                                 | 0.83                |

Fig. 4. Sensitivity analyses showing which forms of PK have the biggest impact on (conservative) posterior confidence $c$. 

**TABLE 1**

A Summary of 3 Beta Prior Distributions of $pfd$
7 Discussion

7.1 Skepticism About Model Assumptions
Software reliability assessments should be conservative: to wit, only when test results stand up to the most critical scrutiny can confidence be justifiably placed in the system satisfying a \( pfd \) bound. Conservative assessments require a skeptical assessor. In Bayesian terms, our assessor holds conservative beliefs about what the evidence implies for a system’s reliability, and about the validity of statistical modelling assumptions.

This paper illustrates a general, incremental approach to dealing with doubts about any statistical model assumptions: we offer a demonstrably conservative form of Draper’s ideas [33]. For a model property one is doubtful of, one can check the sensitivity of claims based on the model by using a slightly more general model (that has the original model as a special case and weakens the property in question) for inference. “Slight” model generalisations keep models from becoming unnecessarily complex, ensure generalisations cover all scenarios covered by the original models, and minimise eliciting increasingly complex prior distributions.

This incremental approach is a “win-win”. If the i.i.d. assumption is not too optimistic, “i.i.d.”-based confidence doesn’t depart significantly from the CBI theorem’s conservative confidence based on the Klotz model. If the i.i.d. assumption is too optimistic, sensitivity analysis using the theorem can reveal this – in such circumstances, caution is warranted when relying on “i.i.d.”-based reliability claims. And if, in turn, one has doubts about the Klotz model, then a generalisation of the Klotz model can be used to check if “Klotz model”-based confidence is sensitive to doubts.

7.2 Limitations and Future Work
Let us highlight some Klotz model shortcomings. It does not distinguish between different success/failure types. Future work might consider using the models of [16], [55], [56] with CBI, to check the robustness of “Klotz model”-based claims. These models account for the cumulative impact of benign failures.

The Klotz model uses the relative sizes of \( x \) and \( \lambda \) to characterise all pairs of successive Bernoulli trials as being identically positively, negatively, or zero correlated. Consequently, the Klotz model is unable to express non-stationary dependence, such as may be due to software updates that remove, or inadvertently add, faults to the software. The model cannot capture dependence across time either, such as periodic correlations over relatively short, or relatively long, runs of demands. Such periodicity can arise if demands that cause software failure are more likely at certain times when (or certain locations where) operating conditions tend to be more stressful (e.g., for software ensuring aircraft safety, unfavourable weather along a flight path may be more likely at certain times of the year, or more likely along certain flight paths). CBI models with time-dependent correlations are worth exploring.

On the use of \( pfd \), we make the following comment. When the failure process is stationary, \( pfd \) make sense. The probability of the system failing on the \( n \)th demand is the same for all \( n \). But for time-dependent failure probabilities, there are more suitable dependability measures – such as the probability of future failure-free operation. Strigini [56] makes a related point in a classical inference context. Even in the present context of posterior confidence bounds on \( pfd \), it’s worth investigating whether other dependability measures are more/less robust to i.i.d. doubts.

We have formalised (via PK1) and illustrated how any Bayesian reliability assessment using a continuous prior, \( f(x) \), can conservatively incorporate doubts about “i.i.d.”. However, when eliciting \( f(x) \) proves too challenging, future work could extend the theorem to work with partially specified \( f(x) \). For example, prior to testing, one might justifiably have some confidence in the software containing no faults [57]. In effect, \( f(x) \) becomes discontinuous, with a non-zero probability of the \( pfd \) being zero. For such scenarios, preliminary results suggest significantly better agreement between “i.i.d.”-based and “Klotz model”-based posterior confidence, even with a very large number of successful tests.

In general, the “elicitation challenge” remains an open problem for Bayesian approaches. In light of this, best practice approaches should be followed when eliciting PKs [58], [59], while sensitivity analyses (as illustrated in Section 6) is crucial and practical for checking the robustness of confidence to PK changes.

This work has not considered model selection or validation. One might envisage applying CBI to conservatively gain confidence in the i.i.d. assumption, or using Bayes factors and CBI to conservatively determine which modelling assumptions lead to more trustworthy predictions about future system reliability.

The theorem could be extended to account for failures during testing, e.g., when assessing machine learning applications, or extended to support conservative claims for software modules in a fault-tolerant configuration (e.g., extending Singh et al. [19]).

8 Conclusion
When assessing software using operational testing it is natural to ask, “is it appropriate to assume the software’s failures and successes arise in an i.i.d. manner?” For many practical scenarios there are well-known reasons to doubt the i.i.d. assumption. A few statistical models which weaken the assumption have been proposed for use in reliability assessments [13], [14], [16], [55]. However, none of these proposals allow an assessor to remain unsure about whether i.i.d. holds or not, nor allow the assessor to see the impact of their uncertainty on their confidence in a \( pfd \) bound. This, despite it often being the case that an assessor may have good reason to believe in i.i.d., but not enough reason to be certain that it holds. Furthermore, these proposals do not directly support conservative reliability claims.

Using conservative Bayesian techniques – in particular, CBI – we show how doubts about i.i.d. can be formally included in software reliability assessments (see Sections 5 and 6). In this way, we obviate the need imposed by the previous proposals – the need for assessors to either assume/conclude that the software test outcomes are i.i.d., or assume/conclude that they aren’t. Instead, our method allows a skeptical assessor’s confidence in a target \( pfd \), and their confidence in (and doubts about) i.i.d., to grow or shrink in response to seeing the software operate without failure.
Moreover, CBI’s conservative confidence bounds continuously invite our assessor to be skeptical, and to question whether seemingly favourable reliability evidence from testing does, in fact, corroborate actual reliability. For example, while failure-free operation is generally an indicator of a desirable pfd, our results highlight why this might be (at best) a sanguine view in some situations. There may be undesirable reasons for why failure-free operation is occurring; reasons that ultimately undermine one’s confidence in the pfd being sufficiently small (see Section 6). CBI, by weakening the i.i.d. assumption and producing conservative confidence bounds, can call into question the representativeness of failure-free operation (as indicating a reliable system). When this happens, it’s incumbent on the assessor to rule out potential problems during testing that “masquerade” as failure-free operation, and to incorporate these efforts into any further use of CBI.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers whose comments were very helpful in improving the presentation. We are also grateful to Bev Littlewood for giving very helpful feedback on an initial draft of this paper.

REFERENCES

[1] T. E. Wierman, S. T. Beck, M. B. Calley, S. A. Eide, C. D. Gentillon, and W. E. Kohn, “Reliability study: Combustion engineering reactor protection system, 1984–1998,” Idaho National Engineering and Environmental Laboratory, U. S. Nuclear Regulatory Commission Washington, DC, Tech. Rep. NUREG/CR-5300, 2001. [Online]. Available: https://www.nrc.gov/docs/ML0220/ML022050271.pdf
[2] C. Bunea, T. Chariats, R. M. Cooke, and G. Becker, “Two-stage Bayesian models—Application to ZEDB project,” Rel. Eng. Syst. Saf., vol. 90, no. 2, pp. 123–130, 2005.
[3] K. Pörn, “The two-stage Bayesian method used for the T-book application,” Rel. Eng. Syst. Saf., vol. 51, no. 2, pp. 169–179, 1996.
[4] N. E. Fenton and M. Neil, “Software metrics: Successes, failures and new directions,” J. Syst. Softw., vol. 47, no. 2, pp. 149–157, 1999.
[5] R. A. Thayer, M. Lipow, and E. C. Nelson, Software Reliability. Amsterdam, Netherlands: North-Holland, 1978.
[6] J. W. Duran and S. C. Ntafos, “An evaluation of random testing,” IEEE Trans. Softw. Eng., vol. SE-10, no. 4, pp. 438–444, Jul. 1984.
[7] C. Atwood et al., “Handbook of parameter estimation for probabilistic risk assessment,” U. S. Nuclear Regulatory Commission, Washington, DC, Report NUREG/CR-6823, 2003.
[8] P. E. Ammann and J. C. Knight, “Data diversity: An approach to software fault tolerance,” IEEE Trans. Comput., vol. 37, no. 4, pp. 418–425, Apr. 1988.
[9] P. Bishop, “The variation of software survival time for different operational input profiles (or why you can wait a long time for a big bug to fail),” in Proc. IEEE 23rd Int. Symp. Fault-Tolerant Comput., 1993, pp. 98–107.
[10] R. Huang, W. Sun, Y. Xu, H. Chen, D. Towey, and X. Xia, “A survey on adaptive random testing,” IEEE Trans. Softw. Eng., vol. 47, no. 10, pp. 2052–2083, Oct. 2021.
[11] A. Cesnki, “Reliability analysis of recovery blocks with nested clusters of failure points,” IEEE Trans. Rel., vol. 42, no. 1, pp. 12–18, Mar. 1993.
[12] L. A. Tomek, J. K. Muppala, and K. S. Trivedi, “Modeling correlation in software recovery blocks,” IEEE Trans. Softw. Eng., vol. 19, no. 11, pp. 1071–1086, Nov. 1993.
[13] S. Chen and S. Mills, “A binary Markov process model for random testing,” IEEE Trans. Softw. Eng., vol. 22, no. 3, pp. 218–223, Mar. 1996.
[14] K. Tchepo-Popstojanova and K. S. Trivedi, “Failure correlation in software reliability models,” IEEE Trans. Rel., vol. 49, no. 1, pp. 37–48, Mar. 2000.
[15] Y. Dai, M. Xie, and K. Poh, “Modeling and analysis of correlated software failures of multiple types,” IEEE Trans. Rel., vol. 54, no. 1, pp. 100–106, Mar. 2005.
[16] A. Bondavalli, S. Chiarradonna, F. Di Giandomenico, and L. Strigini, “Dependency models for iterative software considering correlation between successive inputs,” in Proc. IEEE Int. Comput. Perform. Depend. Symp., 1995, pp. 13–21.
[17] J. Klotz, “Statistical inference in bernoulli trials with dependence,” Ann. Statist., vol. 1, no. 2, pp. 373–379, 1973.
[18] K. W. Miller et al., “Estimating the probability of failure when testing reveals no failures,” IEEE Trans. Softw. Eng., vol. 18, no. 1, pp. 33–43, Jan. 1992.
[19] H. Singh, V. Corteseca, B. Cukic, E. GuneI, and V. Bharadwaj, “A Bayesian approach to reliability prediction and assessment of component based systems,” in Proc. IEEE 12th Int. Symp. Softw. Rel. Eng., 2001, pp. 12–21.
[20] B. Littlewood, P. Popov, and L. Strigini, “Assessing the reliability of diverse fault-tolerant software-based systems,” Saf. Sci., vol. 40, no. 9, pp. 781–796, 2002.
[21] P. Popov, “Bayesian reliability assessment of legacy safety-critical systems upgraded with fault-tolerant off-the-shelf software,” Rel. Eng. Syst. Saf., vol. 117, pp. 98–113, 2013.
[22] P. Bishop, R. Bloomfield, B. Littlewood, A. Povyakalo, and D. Wright, “Toward a formalism for conservative claims about the dependability of software-based systems,” IEEE Trans. Softw. Eng., vol. 37, no. 5, pp. 708–717, Sep./Oct. 2011.
[23] L. Strigini and A. Povyakalo, “Software fault-freeness and reliability predictions,” in Proc. Int. Conf. Comput. Rel. Saf. Secure, 2013, pp. 106–117.
[24] K. Salako, “Loss-size and reliability trade-offs amongst diverse redundant binary classifiers,” in Proc. Int. Conf. Quantitative Eval. Syst., 2020, pp. 96–114.
[25] X. Zhao, V. Robu, D. Flynn, K. Salako, and L. Strigini, “Assessing the safety and reliability of autonomous vehicles from road testing,” in Proc. IEEE/IFIP 51st Annul. Intn. Conf. Dependable Syst. Netw., 2021, pp. 451–462.
[26] J. O. Berger, “An overview of robust Bayesian analysis,” Test, vol. 3, no. 1, pp. 5–124, 1994.
[27] J. O. Berger, “Robust Bayesian analysis: Sensitivity to the prior,” J. Statist. Plan. Inference, vol. 25, pp. 303–328, 1990.
[28] M. Lavine, “Sensitivity in Bayesian statistics: The prior and the likelihood,” J. Amer. Statist. Assoc., vol. 86, no. 414, pp. 396–399, 1991.
[29] J. Berger and E. Moreno, “Bayesian robustness in dimensionless models: Prior independence,” J. Statist. Plan. Inference, vol. 40, no. 2, pp. 161–176, 1994.
[30] D. Draper, “Assessment and propagation of model uncertainty,” J. Roy. Statist. Society. Ser. B. (Methodol.), vol. 57, no. 1, pp. 45–97, 1995.
[31] L. R. Pericchi and M. E. Perez, “Posterior robustness with more informative prior,” J. Statist. Plan. Inference, vol. 40, no. 2, pp. 279–294, 1994.
[32] PD IEC TR 63161:2022, “Assignment of safety integrity require-ments: Basic rationale,” IEC, Geneva, Switzerland, Standard, 2022.
[33] M. Rausand, Reliability of Safety-Critical Systems: Theory and Applications. Hoboken, NJ, USA: Wiley, 2014.
[34] M. R. Lyu, Ed., Handbook of Software Reliability Engineering. New York, NY, USA: McGraw-Hill, 1996.
[35] N. D. Singpurwalla and S. P. Wilson, “Software reliability modeling,” Int. Statist. Rev. / Revue Int. de Statistique, vol. 62, no. 3, pp. 289–317, 1994. [Online]. Available: http://www.jstor.org/stable/1403763
[36] D. R. Miller, “Exponential order statistic models of software reliability growth,” IEEE Trans. Softw. Eng., vol. SE-12, no. 1, pp. 12–24, Jan. 1986.
[40] B. Bergman and M. Xie, “On Bayesian software reliability modelling,” J. Statist. Plan. Inference, vol. 29, no. 1, pp. 33–41, 1991. [Online]. Available: https://www.sciencedirect.com/science/article/pii/037837899290119D

[41] B. Littlewood and L. Strigini, “Validation of ultra-high dependability for software-based systems,” Commun. ACM, vol. 36, pp. 69–80, 1993.

[42] B. Littlewood and D. Wright, “Some conservative stopping rules for the operational testing of safety critical software,” IEEE Trans. Softw. Eng., vol. 23, no. 11, pp. 673–683, Nov. 1997.

[43] J. D. Musa, “Operational profiles in software-reliability engineering,” IEEE Softw., vol. 10, no. 2, pp. 14–32, Mar. 1993.

[44] International Electrotechnical Commission, “[Functional safety of electrical/electronic/programmable electronic safety related systems,” IEC, Geneva, Switzerland, Standard, Tech. Rep. IEC 61508:2010, 2010.

[45] R. L. Schilling, Measures, Integrals and Martingales. Cambridge, U.K.: Cambridge Univ. Press, 2005.

[46] D. M. Hunns and N. Wainwright, “Software-based protection for Sizewell B: The regulator’s perspective,” in Proc. IEEE Int. Conf. Elect. Control Aspects Sizewell B. Power Station, 1992, pp. 198–203.

[47] NuSAC study group on the safety of operational computer systems, “The Use of Computers in Safety-critical Applications,” Health and Safety Commission, London, U.K., 1998.

[48] B. Littlewood and L. Strigini, “Validation of ultra-high dependability for software-based systems,” Commun. ACM, vol. 33, no. 6, pp. 636–648, 1990.

[49] L. Strigini and B. Littlewood, “Guidelines for statistical testing,” City University London, Project Report PASCON/WO6-CCN2/TN12, 1997.

[50] The International Regulator Task Force on Safety Critical Software, “Licensing of safety critical software for nuclear reactors. common position of international nuclear regulators and author-ised technical support organisations,” 2022. [Online]. Available: https://www.orntower.org.uk/software.pdf

[51] IEEE 1012:2016, “IEEE Standard for System, Software, and Hardware Verification and Validation,” IEEE, NewYork, USA, Standard, 2016.

[52] IEC 60880, “Nuclear power plants-Instrumentation and control systems important to safety. software aspects for computer-based systems performing category A functions,” IEC, Geneva, Switzerland, Standard, 2006.

[53] B. Littlewood and D. Wright, “The use of multilegged arguments to increase confidence in safety claims for software-based systems: A study based on a BBN analysis of an idealized example,” IEEE Trans. Softw. Eng., vol. 33, no. 5, pp. 347–365, May 2007.

[54] E. T. Barr, M. Harman, P. McMinn, M. Shahbaz, and S. Yoo, “The oracle problem in software testing: A survey,” IEEE Trans. Softw. Eng., vol. 41, no. 5, pp. 507–525, May 2015.

[55] A. Bondavalli, S. Chiariadonna, F. Di Giandomenico, and S. La Torre, “Modelling the effects of input correlation in iterative software,” Rel. Eng. Syst. Saf., vol. 57, no. 3, pp. 189–202, 1997.

[56] L. Strigini, “On testing process control software for reliability assessment: The effects of correlation between successive failures,” Softw. Testing Verification Rel., vol. 6, no. 1, pp. 33–48, 1996.

[57] B. Littlewood and J. Rushby, “Reasoning about the reliability of diverse two-channel systems in which one channel is “possibly perfect”,” IEEE Trans. Softw. Eng., vol. 38, no. 5, pp. 1178–1194, Sep./Oct. 2012.

[58] PRA Working Group, “A review of NRC staff uses of probabilistic risk assessment,” Nuclear Regulatory Commission U. S., Tech. Rep. NUREG-1489, 1990. [Online]. Available: https://www.nrc.gov/docs/ML0635/ML063540593.pdf

[59] A. O’Hagan et al., “Uncertain judgements: Eliciting experts’ probabilities,” Statist. Pract. New York, NY, USA: Wiley, 2006.

[60] W. Rudin, Principles of Mathematical Analysis, 3rd ed., New York, NY, USA: McGraw-Hill, 1976.

[61] V. Bryant, Metric Spaces: Iteration and Application. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[62] C. D. Aliprantis and K. C. Border, Infinite Dimensional Analysis: A Hitchhiker’s Guide, 2nd ed. Berlin, Germany: Springer, 1999.

Kizito Salako received the double honours (1st-class) degree in mathematics and statistics from the University of Lagos, the master of advanced study in mathematics degree from the University of Cambridge (where he was both a Shell Centenary scholar and a Commonwealth scholar), and the PhD degree in computer science from City, University of London. He is a lecturer with the Department of Computer Science, City, University of London. He is passionate about applications of probability theory, Bayesian statistics, geometry and machine-learning, when simulating, assessing and forecasting the (failure) behaviour of software-based systems. His research produces statistical techniques that support conservative dependability claims for safety-critical systems. He also builds simulations of large-scale, complex, interdependent, critical infrastructure, in order to forecast the occurrence and impact of cascading failures, cyber-attacks and the efficacy of mitigation strategies.

Xingyu Zhao received the bachelor’s and master’s degrees from Beihang University, and the PhD degree in computer science from the Centre for Software Reliability, City, University of London, in 2017. He is a lecturer with the Department of Computer Science, University of Liverpool. His research expertise covers probabilistic verification of autonomous systems, Bayesian inference with partial/vague prior knowledge, reliability assessment and safety assurance, and trustworthy AI. He has published more than 30 papers in interdisciplinary fields of software engineering, trustworthy AI, and system safety and reliability. Beyond publications, he has secured funding from U.K. EPSRC, U.K. DSTL and Innovate U.K. as a co-Investigator.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.