Author's accepted manuscript, to appear in the proc. of ASE2020. DOI: https://doi.org/10.1145/3324884.3416565

Interval Change-Point Detection for Runtime Probabilistic Model Checking

Xingyu Zhao
Heriot-Watt University
Edinburgh, U.K.
xingyu.zhao@hw.ac.uk

Radu Calinescu
University of York
York, U.K.
radu.calinescu@york.ac.uk

Simos Gerasimou
University of York
York, U.K.
simos.gerasimou@york.ac.uk

Valentin Robu
Heriot-Watt University
Edinburgh, U.K.
v.robu@hw.ac.uk

David Flynn
Heriot-Watt University
Edinburgh, U.K.
d.flynn@hw.ac.uk

ABSTRACT

Recent probabilistic model checking techniques can verify reliability and performance properties of software systems affected by parametric uncertainty. This involves modelling the system behaviour using *interval Markov chains*, i.e., Markov models with transition probabilities or rates specified as intervals. These intervals can be updated continually using Bayesian estimators with imprecise priors, enabling the verification of the system properties of interest at runtime. However, Bayesian estimators are slow to react to sudden changes in the actual value of the estimated parameters, yielding inaccurate intervals and leading to poor verification results after such changes. To address this limitation, we introduce an efficient interval change-point detection method, and we integrate it with a state-of-the-art Bayesian estimator with imprecise priors. Our experimental results show that the resulting end-to-end Bayesian approach to change-point detection and estimation of interval Markov chain parameters handles effectively a wide range of sudden changes in parameter values, and supports runtime probabilistic model checking under parametric uncertainty.

CCS CONCEPTS

• **Software and its engineering → Software verification and validation**; • **Computer systems organization → Reliability**; • **Mathematics of computing → Bayesian computation**; **Metropolis-Hastings algorithm**.

KEYWORDS

Change-point detection, interval Markov chains, Bayesian inference, imprecise probability, probabilistic model checking, runtime verification, interval model checking

ASE '20, September 21–25, 2020, Virtual Event, Australia

ACM ISBN 978-1-4503-6768-4/20/09…$15.00
https://doi.org/10.1145/3324884.3416565

ACM Reference Format:
Xingyu Zhao, Radu Calinescu, Simos Gerasimou, Valentin Robu, and David Flynn. 2020. Interval Change-Point Detection for Runtime Probabilistic Model Checking. In *35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20), September 21–25, 2020, Virtual Event, Australia.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3324884.3416565

1 INTRODUCTION

Detecting sudden changes in the parameters of a software or software-controlled system has a plethora of important applications. For a system undergoing development, such changes may correspond to defects being introduced in the code base, and knowing the time when the changes occurred can help identify and remove these defects. For a running system, sudden parameter changes may correspond to faults in a system component such as a disk drive storing a system database, or a sensor used by a robot. Alternatively, they may be due to violations of service-level agreements by third-party components such as a public cloud service, or to cyberattacks, or to environmental changes such as a sudden increase in the rate of requests received by a web server. In all these scenarios, detecting the change supports the identification of its cause, the analysis of its impact on the system, and (if needed) the mitigation of this impact.

Given these benefits, numerous *change-point detection* (CPD) methods have been developed to estimate the time of such sudden changes and the new values of the affected parameters. These methods have been successfully used in domains ranging from software engineering [14, 19] to medicine [29] and finance [36], and are described in multiple surveys, e.g., [1, 12, 32, 34].

Our paper introduces an *interval Change-Point Detection* (iCPD) method that complements existing CPD methods by efficiently solving an important and previously unexplored variant of the problem. The key distinguishing features of iCPD (and thus the main contributions of our paper) are summarised below:

(1) To the best of our knowledge, iCPD is the first method that tackles the detection of sudden changes in systems affected by parametric uncertainty and verified through the probabilistic model checking of *interval Markov chains*. Interval Markov chains are Markov models whose transition probabilities (for discrete-time Markov chains) and transition rates (for continuous-time Markov chains) are expressed as intervals due to the epistemic

uncertainty affecting the corresponding system parameters. Examples of such parameters include the success probability of a web service invocation and the measurement rate of a sensor, as incomplete knowledge often means that these can only be estimated correctly using intervals such as $[0.92, 0.95]$ and $[6\mathrm{s}^{-1}, 7.5\mathrm{s}^{-1}]$, respectively.[1]

(2) iCPD is integrated with a recently developed *Bayesian estimator with imprecise priors* [35], and computes new intervals of priors for this estimator after each sudden change detected in the monitored transition parameter (i.e., probability or rate) of a Markov chain. Bayesian estimators with imprecise priors (also called *robust Bayesian estimators*) [6] associate intervals with the uncertain parameters of a system; they use intervals as priors, and their posteriors are also intervals.

(3) Its integration with a robust Bayesian estimator enables iCPD to decide in constant time whether a new system observation (corresponding to a state transition in the Markov chain) should trigger the full CPD analysis, which is computationally more expensive. This lightweight decision mechanism makes iCPD particularly suited for online use, and is missing from traditional CPD methods.

(4) Used in conjunction, iCPD and the robust Bayesian estimator [35] form an end-to-end Bayesian approach to change-point detection and estimation of interval Markov chain parameters, and support the effective runtime probabilistic model checking of systems affected by parametric uncertainty.

We structure the rest of the paper as follows. Section 2 provides the required background on probabilistic model checking and robust Bayesian estimators. Our iCPD change-point detection method and its integration with a robust Bayesian estimator are described in Section 3, and their effectiveness is evaluated in Section 4. Finally, Section 5 compares iCPD to existing CPD methods, and Section 6 summarises the paper and suggests areas of future work.

2 PRELIMINARIES

2.1 Probabilistic model checking

Probabilistic model checking (PMC) is a formal technique for verifying the correctness, reliability, and performance of systems characterised by stochastic behaviour [3, 27], where this behaviour is modelled by Markov chains. Formally, a Markov chain is a tuple $M = (S, s_0, \delta)$, where S is a finite set of states, $s_0 \in S$ is the initial state, and δ is a state-transition function defined as:

- $\delta : S \times S \to [0, 1]$ for discrete-time Markov chains (DTMCs), with $\delta(s_i, s_j) = p_{ij}$ giving the probability of transition between states $s_i, s_j \in S$, and $\sum_{s_j \in S} \delta(s_i, s_j) = 1$;
- $\delta : S \times S \to \mathbb{R}_{\geq 0}$ for continuous-time Markov chains (CTMCs), with $\delta(s_i, s_j) = r_{ij}$ giving the rate of transition between states $s_i, s_j \in S$.

PMC supports the verification of discrete-time properties (e.g., successful completion probability of a protocol) using DTCMs, and of continuous-time properties (e.g., expected execution time of an application, or energy consumption of a device) using CTMCs. To this end, the states of Markov models are labelled with atomic propositions that hold in those states, and the properties to verify are expressed in temporal logics over these atomic propositions, e.g., probabilistic temporal tree logic (PCTL) [8, 24] for DMTCs and continuous stochastic logic (CSL) [2, 4] for CTMCs. Efficient PMC algorithms are available, and are implemented by widely used probabilistic model checkers such as PRISM [28] and Storm [15].

Recent advances in PMC [9, 13, 30, 37] support the verification of *interval Markov chains* [26]. In these models, the transition probabilities of DTMCs and the transition rates of CTMCs can be specified as intervals, enabling the representation of parametric uncertainty for the modelled systems. Accordingly, the verification of interval Markov chains with automated tools such as Prism-PSY [37] and FACT [9] yields value intervals for the reliability and performance properties of the verified system.

2.2 Robust Bayesian estimation of interval Markov chain parameters

Building on the theory of imprecise probability with sets of priors (IPSP) [33], recent research has introduced a robust Bayesian estimator [35] for the transition parameters of interval Markov chains. This IPSP estimator supports the use of imprecise prior knowledge in the Bayesian learning process, and provides bounded estimates on transition parameters. Due to space constraints, we only detail the operation of the IPSP estimator for a generic transition probability p_{ij} between states s_i and s_j of an interval DTMC (iDTMC); the changes needed to use the estimator for an interval CTMC (iCTMC) transition rate are mentioned at the end of the section.

Given a DTMC (S, s_0, δ) and a state $s_i \in S$, the outgoing transitions from s_i follow a *multinomial distribution* with parameters given by the probabilities $\delta(s_i, \cdot)$ of these transitions. As such, if we observe n_{ij} transitions from s_i to a state s_j out of n_i outgoing transitions from s_i (a scenario labelled 'data' in the equations below), the binomial likelihood for this scenario is (by omitting the combinatorial factor that will be cancelled in the Bayes formula):

$$Pr(\text{data} \mid p_{ij}) = p_{ij}^{n_{ij}} (1 - p_{ij})^{n_i - n_{ij}}, \tag{1}$$

where $p_{ij} = \delta(s_i, s_j)$. As usual, for mathematical convenience in Bayesian inference, the IPSP estimator uses a *conjugate* prior distribution for the above likelihood function, i.e., a beta distribution of p_{ij}, thus ensuring that the posterior is also beta distributed. However, unlike point estimators that use the posterior mean as an approximation for the value of p_{ij} (e.g., [18]), the IPSP estimator can, more realistically, operate with *imprecise prior knowledge* obtained, for instance, from a group of experts or derived from noisy historical data.

The IPSP estimator uses the canonical parameterisation $Beta\big(n_i^{(0)}, p_{ij}^{(0)}\big)$ of the beta distribution,[2] which allows an intuitive interpretation of $p_{ij}^{(0)}$ as the "best prior guess", and of $n_i^{(0)}$ as the size of an imaginary sample on which the prior estimation $p_{ij}^{(0)}$ is based [33]. However, instead of point values for the two prior parameters, the

[1] The use of point estimates for uncertain parameters is rarely justified [6]; for Markov models, it can lead to highly inaccurate verification results that may endorse invalid software engineering decisions [9].

[2] i.e., a parameterisation in which the shape parameters α and β of the common parameterisation $Beta(\alpha, \beta)$ are replaced by $n_i^{(0)} = \alpha + \beta$ and $p_{ij}^{(0)} = \alpha/(\alpha + \beta)$, where the superscript '(0)' reflects the fact that these values represent the knowledge before any observation is available.

IPSP estimator operates with *prior intervals*

$$[\underline{n_i}^{(0)}, \overline{n_i}^{(0)}], \qquad [\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}] \tag{2}$$

for $n_i^{(0)}$ and $p_{ij}^{(0)}$, respectively. Accordingly, the lower and upper bounds for $p_{ij}^{(n_i)}$, the posterior of interest after n_i observations of outgoing transitions from state s_i of the iDTMC, can be calculated in constant time as [33]:

$$\begin{aligned}
\underline{p_{ij}}^{(n_i)} &= \begin{cases} \dfrac{\overline{n_i}^{(0)}\underline{p_{ij}}^{(0)}+n_{ij}}{\overline{n_i}^{(0)}+n_i}, & \text{if} \quad \dfrac{n_{ij}}{n_i} \geq \underline{p_{ij}}^{(0)} \\ \dfrac{\underline{n_i}^{(0)}\underline{p_{ij}}^{(0)}+n_{ij}}{\underline{n_i}^{(0)}+n_i}, & \text{otherwise} \end{cases} \\
\overline{p_{ij}}^{(n_i)} &= \begin{cases} \dfrac{\overline{n_i}^{(0)}\overline{p_{ij}}^{(0)}+n_{ij}}{\overline{n_i}^{(0)}+n_i}, & \text{if} \quad \dfrac{n_{ij}}{n_i} \leq \overline{p_{ij}}^{(0)} \\ \dfrac{\underline{n_i}^{(0)}\overline{p_{ij}}^{(0)}+n_{ij}}{\underline{n_i}^{(0)}+n_i}, & \text{otherwise} \end{cases}
\end{aligned} \tag{3}$$

The IPSP robust estimator [35] is generic to Bayesian inference in canonical exponential families [33], where the Gamma-Poisson setup is typically applied (cf. Proposition 5.4 and Example 5.5 in [7, pp.266–277]). Thus, IPSP can be applied to iCTMC transition rates by replacing the beta priors with gamma priors.

3 INTERVAL CHANGE-POINT DETECTION

3.1 Problem definition

Our iCPD method is applicable to systems modelled by interval Markov chains whose transition parameters are associated with system parameters affected by sudden changes. Given such a system and an interval Markov chain that models its behaviour, we assume that:

(1) The system is monitored and all the events that correspond to state transitions within the Markov chain (and, for iCTMCs, the timing of these events) are recorded. Examples of such events include the invocation of a method, the receipt of a database query, and the failure or repair of a server.
(2) The intervals for the transition probabilities or rates of the interval Markov chains are continually updated using the IPSP robust estimator from Section 2.2.

The information required to detect sudden changes in the probability p_{ij} of transitioning between states s_i and s_j of an iDTMC consists of the sequence of observations $o_1, o_2, \ldots, o_{n_i}$ of all events corresponding to the outgoing transitions from state s_i, where

$$\forall k = 1, 2, \ldots, n_i, \; o_k = \begin{cases} 1, \text{ if the } k\text{-th transition is to state } s_j \\ 0, \text{ otherwise} \end{cases} \tag{4}$$

The *monitoring window*, defined as $w = n_i$ for an iDTMC, may be fixed (in which case only the most recent n_i observations are used) or may include all observations since the monitoring began.

Given this information, the *interval change-point detection problem for an iDTMC transition probability* p_{ij} is to determine:

(1) whether the value of p_{ij} experienced a sudden change within the monitoring window w;
(2) if the answer to (1) is positive, the time step from $\{1, 2, \ldots, n_i\}$ when the change occurred;
(3) if the answer to (1) is positive, new prior intervals (2) for the robust IPSP estimator of p_{ij}.

The *interval change-point detection problem for an iCTMC transition rate* r_{ij} is defined similarly. In this case, the system monitor needs to also record the sojourn times $t_1, t_2, \ldots, t_{n_i}$ that the system spent in state s_i prior to undertaking each of the n_i outgoing transitions given by observations (4), and the monitoring window is defined as $w = \sum_{k=1}^{n_i} t_k$.

3.2 The iCPD method

Fig. 1 shows a generic window of $w = n_i$ observations (4) for an iDTMC, where the shaded circles correspond to transitions from state s_i to state s_j, and the empty circles correspond to transitions from s_i to other states than s_j. Assuming that the transition probability p_{ij} undergoes a sudden change from $p_{ij} = a$ to $p_{ij} = b$ within this time window; we use a random variable x to denote the unknown change point in the sequence, and $N(x) = \sum_{k=1}^{x} o_k$ and $M(x) = \sum_{k=x+1}^{n_i} o_k$ to denote the number of transitions from s_i to s_j before and after the time step x, respectively. An analogous notation can be defined for a transition rate r_{ij} of an iCTMC changing suddenly from $r_{ij} = a$ to $r_{ij} = b$ within the time window; in this case, we have $N(x) = \sum_{k=1}^{x} o_k t_k$ and $M(x) = \sum_{k=x+1}^{n_i} o_k t_k$. Given the parametric uncertainty associated with systems modelled by interval Markov chains, not only the change time x but also the precise values of a and b are unknown. The following proposition defines a Bayesian estimator for these values (this is a formalisation of a known result, e.g., presented only for DTMCs in [19]).

Proposition 3.1. *Given a prior joint distribution $f(a, b, x)$ for the unknown values a and b of an interval Markov chain parameter, and for the unknown time x when the parameter value changes from a to b, the (marginal) estimates for the three unknowns are given by the posterior joint distribution*

$$\frac{a^{N(x)}(1-a)^{x-N(x)} b^{M(x)}(1-b)^{w-x-M(x)} f(a,b,x)}{\iiint_D a^{N(x)}(1-a)^{x-N(x)} b^{M(x)}(1-b)^{w-x-M(x)} f(a,b,x) \, \mathrm{d}a \, \mathrm{d}b \, \mathrm{d}x} \tag{5}$$

for an iDTMC, where $D = [0,1] \times [0,1] \times \{1, 2, \ldots, w\}$, and

$$\frac{a^{N(x)} e^{-ax} b^{M(x)} e^{-b(w-x)} f(a,b,x)}{\iiint_D a^{N(x)} e^{-ax} b^{M(x)} e^{-b(w-x)} f(a,b,x) \, \mathrm{d}a \, \mathrm{d}b \, \mathrm{d}x} \tag{6}$$

for an iCTMC, where $D = [0, \infty) \times [0, \infty) \times [0, w]$.[3]

Proof. As in (1), we can write the likelihood function associated with the iDTMC 'data' from Fig. 1 as

$$\begin{aligned} L(a, b, x; \text{data}) &= Pr(\text{data until } x \mid a, b, x) \cdot Pr(\text{data after } x \mid a, b, x) \\ &= \left(a^{N(x)}(1-a)^{x-N(x)}\right) \cdot \left(b^{M(x)}(1-b)^{w-x-M(x)}\right) \end{aligned} \tag{7}$$

and the likelihood function associated with analogous observations for an iCTMC as

$$\begin{aligned} L(a, b, x; \text{data}) &= Pr(\text{data until } x \mid a, b, x) \cdot Pr(\text{data after } x \mid a, b, x) \\ &= \left(a^{N(x)} e^{-ax}\right) \cdot \left(b^{M(x)} e^{-b(w-x)}\right). \end{aligned} \tag{8}$$

[3] We use Lebesgue-Stieltjes integration in (5), (6) and throughout the rest of the paper, to cover in a compact way both discrete and continuous prior distributions f, as these integrals naturally reduce to sums for discrete distributions.

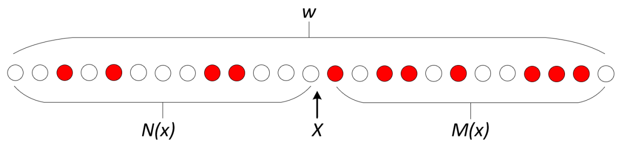

Figure 1: Sequence of w successive transitions from state s_i of a DTMC; shaded circles denote transitions to state s_j.

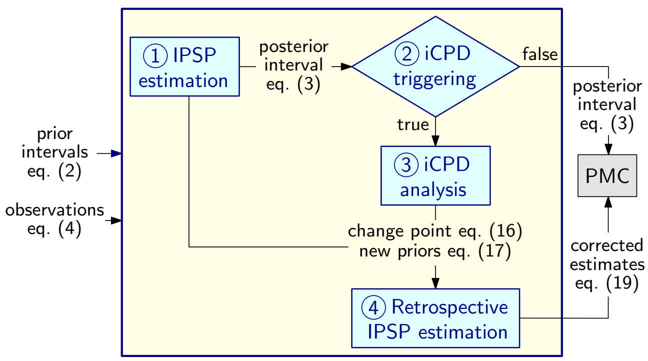

Figure 2: Robust Bayesian change-point detection and estimation for interval Markov chain parameters (shown for an iDTMC transition probability) supports runtime probabilistic model checking (PMC) under parametric uncertainty.

Applying the Bayes theorem to the prior joint distribution $f(a, b, x)$ and to these likelihood functions yields the result from (5) and (6), respectively. □

As we explain later in this section, our iCPD method:

- avoids the difficulty and bias of choosing a *multivariate* prior distribution $f(a, b, x)$ (which is hard to be elicited from human experts in practice and implicitly introduces unjustified assumptions) by reducing the number of unknowns;
- determines the posterior estimates of interest with lower computational cost than the traditional way of using Gibbs samplings for (5) and (6) (e.g., [19]);
- uses a lightweight mechanism to only trigger the CPD analysis under certain conditions, reducing the CPD overheads further.

iCPD operates in conjunction with the IPSP robust Bayesian estimator described in Section 2.2, as an end-to-end Bayesian approach to CPD and estimation of interval Markov chain parameters. This approach is shown in Fig. 2, and its four steps are detailed next.

1 IPSP estimation – IPSP runs as the online robust Bayesian estimator for the uncertain transition parameters of the interval Markov chain under verification. As described in Sec. 2.2, this provides bounded estimates for the transition parameter, e.g., (3) for iDTMCs.

2 iCPD triggering – The width of the interval defined by the IPSP bounds from step 1 is monitored, and increases in this width are used to trigger a change-point detection analysis. This trigger exploits the ability of IPSP to detect *prior-data conflicts* [20], i.e., conflicts between prior beliefs and the observed data in Bayesian inference. For an iDTMC transition probability for instance, the interval width can be written [33] as a sum of two terms:[4]

$$\overline{p_{ij}}^{(n_i)} - \underline{p_{ij}}^{(n_i)} = \frac{\overline{n_i}^{(0)}\left(\overline{p_{ij}}^{(0)} - \underline{p_{ij}}^{(0)}\right)}{\overline{n_i}^{(0)} + n_i} + \frac{\Delta^{(n_i)} n_i \left(\overline{n_i}^{(0)} - \underline{n_i}^{(0)}\right)}{\left(\overline{n_i}^{(0)} + n_i\right)\left(\underline{n_i}^{(0)} + n_i\right)} \tag{9}$$

where

$$\Delta^{(n_i)} = \Delta\left(\frac{n_{ij}}{n_i}; \overline{p_{ij}}^{(0)}, \underline{p_{ij}}^{(0)}\right) = \begin{cases} \frac{n_{ij}}{n_i} - \overline{p_{ij}}^{(0)}, & \text{if } \frac{n_{ij}}{n_i} > \overline{p_{ij}}^{(0)} \\ \underline{p_{ij}}^{(0)} - \frac{n_{ij}}{n_i}, & \text{if } \frac{n_{ij}}{n_i} < \underline{p_{ij}}^{(0)} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

represents [33] the *degree of prior-data conflict* after observation n_i. The first term in (9) decreases as n_i grows (and becomes negligible for $n_i \gg 1$), whereas the second term becomes suddenly positive when a prior-data conflict $\frac{n_{ij}}{n_i} \notin \left[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}\right]$ occurs, leading to an increase in the interval width (also cf. Remark 4.2 in [33]).

Note that the computational cost of our iCPD triggering is negligible, as the IPSP interval width can be computed in constant time. Additionally, as we will show in Section 4, the sensitivity of the iCPD trigger can be configured by adjusting the IPSP estimator parameters and the size of the sliding window for the observations (4).

3 iCPD analysis – This step is executed infrequently, i.e., only when the iCPD trigger from step 2 is exercised. To further lower the iCPD overheads, we reduce the number of unknowns from the multivariate Bayesian estimator in Proposition 3.1 by replacing a and b with their maximum likelihood estimations (MLEs) calculated using the observations made until and after the (unknown) change point x, respectively. The justification for using these MLEs is twofold. First, since no prior-data conflict exists until x, the observations collected until x reflect well the true value of a. Second, after the sudden change at time x, we have no prior knowledge about the new value of the transition parameter, so using the MLE for b is the best strategy available. The next result formalises the effect of using the two MLEs for an iDTMC transition probability.[5]

Proposition 3.2. *Using the MLEs for a and b in the multivariate Bayesian estimator from Proposition 3.1, and assuming no prior knowledge about the change point x reduces the posterior* (5) *to:*

$$f_X(x \mid data) = \frac{L_X(x; data)}{\sum_{x=1}^{n_i} L_X(x; data)}, \text{ where } L_X(x; data) = \tag{11}$$
$$\frac{N(x)^{N(x)}(x - N(x))^{x - N(x)} M(x)^{M(x)} (n_i - x - M(x))^{n_i - x - M(x)}}{x^x (n_i - x)^{n_i - x}}.$$

Proof. For an iDTMC transition probability, the MLEs for a and b are $\hat{a} = \frac{N(x)}{x}$ and $\hat{b} = \frac{M(x)}{w-x}$, so the likelihood (7) becomes:

$$L_X(x; \text{data}) = Pr(\text{data} \mid x) =$$
$$\left(\frac{N(x)}{x}\right)^{N(x)} \left(1 - \frac{N(x)}{x}\right)^{x - N(x)} \left(\frac{M(x)}{n_i - x}\right)^{M(x)} \left(1 - \frac{M(x)}{n_i - x}\right)^{n_i - x - M(x)} \tag{12}$$

Additionally, with no prior knowledge of where the change point x is, we need to assume a discrete uniform distribution $f_X(x)$ for it:

$$\forall x \in \{1, 2, \ldots, n_i\},\ f_X(x) = 1/n_i. \tag{13}$$

[4] The calculations for an iCTMC transition rate are entirely similar, and not included in the paper due to space constraints.
[5] The adjustments needed to replicate this result for an iCTMC transition rate are summarised at the end of the description for this step.

The posterior (11) can now be obtained by using the likelihood (12) (after simple algebraic manipulations) and the prior (13) in (5), with the Lebesgue-Stieltjes integration rewritten as a summation. □

To estimate the univariate posterior distribution (11), iCPD uses the Markov chain Monte Carlo (MCMC) method with the Metropolis-Hastings algorithm (whose presentation is out of the scope here, but is available in [17]). This yields an MCMC sample sequence

$$\langle x_1, x_2, \ldots, x_k \rangle \tag{14}$$

for the change point x, and from this iCPD computes a sample sequence

$$\langle b_1, b_2, \ldots, b_k \rangle = \left\langle \frac{M(x_1)}{n_i - x_1}, \frac{M(x_2)}{n_i - x_2}, \ldots, \frac{M(x_k)}{n_i - x_k} \right\rangle \tag{15}$$

for the unknown transition probability b. Note that computing the latter sequence in this way is far more efficient than using Gibbs sampling (if we treat b as an unknown) to obtain b_i, $1 \leq i \leq k$, from a conditional probability distribution given x_i.

Given the MCMC sample sequences (14) and (15), iCPD computes the change point as

$$\hat{x} = \frac{\sum_{i=1}^{k} x_i}{k} \tag{16}$$

and calculates the following new IPSP prior intervals (2):

$$\begin{aligned}\left[\underline{n_i}^{(0)}, \overline{n_i}^{(0)}\right] &= [w - \max\{x_1, x_2, \ldots, x_k\}, w - \min\{x_1, x_2, \ldots, x_k\}] \\ \left[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}\right] &= [\min\{b_1, b_2, \ldots, b_k\}, \max\{b_1, b_2, \ldots, b_k\}]\end{aligned} \tag{17}$$

These calculations can be preceded by an elimination of any outliers that might be present in the sequences (14) and (15).

iCPD performs calculations similar to (12)–(17) for an iCTMC transition rate. These calculations treat a and b as rates, start from the likelihood function (8) to compute the univariate likelihood function

$$Pr(\text{data} \mid x) = \left(\frac{N(x)}{x}\right)^{N(x)} e^{-x} \cdot \left(\frac{M(x)}{w - x}\right)^{M(x)} e^{-M(x)}, \tag{18}$$

and yield prior rate intervals $\left[\underline{r_{ij}}^{(0)}, \overline{r_{ij}}^{(0)}\right]$ instead of the prior probability intervals $\left[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}\right]$ in (17).

4 Retrospective IPSP estimation – The IPSP estimates

$$\left[\underline{p_{ij}}^{(\hat{x})}, \overline{p_{ij}}^{(\hat{x})}\right], \left[\underline{p_{ij}}^{(\hat{x}+1)}, \overline{p_{ij}}^{(\hat{x}+1)}\right], \ldots, \left[\underline{p_{ij}}^{n_i}, \overline{p_{ij}}^{n_i}\right] \tag{19}$$

after the change point (16) are retrospectively corrected.[6] To this end, the IPSP estimator is reapplied to the observations $o_{\hat{x}+1}$, $o_{\hat{x}+2}$, …, o_{n_i} using the new priors (17), and the corrected interval estimates are used to retrospectively revise and re-analyse the interval Markov chain under verification. This is often essential in order to establish the impact that the sudden change has already had on the modelled system, e.g., due to additional energy that may have been used or (as we will show in Section 4) loss of throughput between the change point and its detection time.

[6] We only describe this step for an iDTMC transition probability, as its application to an iCTMC transition rate is entirely similar.

4 EVALUATION

4.1 Research questions

We evaluated our iCPD approach by performing extensive experiments to address the following research questions.

RQ1 (Accuracy): How accurately does iCPD detect points of change in various scenarios? We carried out experiments to assess the ability of iCPD to detect a wide range of changes.

RQ2 (Configurability): How easy is to configure iCPD to operate with different trade-offs? We assessed the ease of calibrating iCPD to operate with data window sizes and triggering sensitivities that support different needs in terms of detection speed and/or trade-offs between false positives and false negatives.

RQ3 (Efficiency:) What are the computational overheads of iCPD ? Since we devised iCPD for the runtime detection of change points in the transition parameters of interval Markov chains, we measured its overheads across a range of configurations and parameter-change scenarios.

RQ4 (Verification support): How effectively does iCPD support the accurate PMC of system-level properties at runtime? We examined the effect of using iCPD to continually update interval Markov chain parameters, supporting the runtime PMC of key system properties.

4.2 Evaluation methodology

Our experimental setup comprises a wide range of scenarios in which each scenario corresponds to a sequence of observations $o_1, o_2, \ldots, o_{n_i}$ for the unknown transition parameter (probability p_{ij} or rate r_{ij} for iDTMC and iCTMC, respectively) instrumented with changes representing four classes of change patterns commonly studied in related research [10, 22]. In particular, we used the following patterns: (i) *Step* where a sudden change causes the parameter value to increase instantaneously; (ii) *Square* that extends the *Step* pattern with another sudden change indicating a recovery of the parameter to its original value; (iii) *Ramp* that represents a gradual or steep change to the parameter value; and (iv) *Fixed* where the parameter value remains constant for the duration of the scenario (to assess whether iCPD produces false positives).

To answer RQ1–RQ3, for each scenario we firstly define an *unknown* change point x (as ground truth) and the *unknown* actual (ground truth) transition parameter values a and b before and after the change point x, respectively. All this information is *unknown* to iCPD. Then, for iDTMCs we generate the sequence of observations $o_1, o_2, \ldots, o_{n_i}$ by sampling from two Binomial distributions whose probabilities are a and b, respectively. Similarly, the sequence of observations for iCTMCs is sampled from two exponential distributions whose mean time is set to $\frac{1}{a}$ and $\frac{1}{b}$, respectively. Finally, we run the scenario by providing to iCPD the observations in order as described in Section 3.2.

We answer RQ4 using the CTMC model of an embedded system from the area of autonomous underwater vehicles (AUV) adapted from [11, 23]. The AUV is deployed on an oceanic surveillance mission, and the successful mission completion requires the continual verification of the CTMC model to ensure compliance with a pair of reliability and performance requirements. The AUV is equipped with several on-board sensors whose operating rates are unknown and could potentially vary due to sensor failure or degradation in operating rate. We combine our iCPD with the probabilistic model checker Prism-PSY [37] and evaluate the extent to which iCPD can detect abrupt changes in sensor rates and update the transition parameters of the CTMC, thus supporting the prompt system reconfiguration due to the violation of mission requirements.

All experiments were run on a Windows 10 Pro 64 bit machine with Intel 1.80GHz 4 cores i7-8550U CPU and 16GB RAM. The code, Markov models, data used for the experimental evaluation and the full experimental results are publicly available at https://github.com/x-y-zhao/iCPD.

4.3 Results and discussion

RQ1 (Accuracy). Fig. 3 shows the estimated iCPD transition probability interval for the unknown transition probability p_{ij} over nine iDTMC scenarios instrumented with the following change patterns: (A) big step, (B) medium step, (C) small step, (D) square representing a normal recovery, (E) square representing a quick recovery, (F) steep ramp, (G) gradual ramp, (H) early big step and (I) fixed. Table 1 presents the configurations used for each scenario including the change pattern and the (unknown) actual values for both the change point x and the transition probability p_{ij}, along with iCPD-related analysis results, i.e., the estimated change point $\hat{x}$, the trigger point x^t and the total computational cost (both representing the timing information of iCPD's runtime analysis). Fig. 4 shows the approximated posterior distributions of x and b derived from MCMC sampling corresponding to the scenarios from Fig. 3.

In scenario A (Fig. 3A, row A in Table 1), i.e., the big step p_{ij} change from $a = 0.3$ to $b = 0.7$ at change point $x = 500$, we observe that the interval width of the IPSP estimator (the solid green and purple lines) continually decreases as new observations are made signifying no prior-data conflict. At observation o_{690} (the vertical blue line), iCPD detects the change that occurred at o_{500} causing a prior-data conflict (10). This detection triggers the CPD analysis (Step 3 in Section 3.2) which utilises the information up to the triggering point to approximate the posterior distribution of x (Fig. 4A#1) concluding that the change happened at $\hat{x} = 504$ (the yellow vertical line), which is very close to the actual change point $x = 500$. Similarly, iCPD approximates the posterior distribution of b (Fig. 4A#2) and uses (17) to estimate the new IPSP prior intervals $\left[\underline{n_i}^{(0)}, \overline{n_i}^{(0)}\right] = [165, 322]$, $\left[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}\right] = [0.508, 0.740]$ which capture the unknown transition probability $b = 0.7$. Starting from the estimated change point $\hat{x} = 504$ and the new IPSP prior intervals, iCPD uses observations $o_{504}, \ldots, o_{690}$ to execute the retrospective IPSP estimation (Step 4 in Section 3.2) and refine the IPSP interval (*dashed* lines labelled "retrospected" and showing retrospectively corrected lower and upper bounds in Fig 3A). For completeness, the *dotted* lines show the transition probability interval estimated by the IPSP estimator without the CPD capability (cf. Section 2.2). Clearly, the change is not detected and a significant number of new observations is needed before the estimated interval encloses the updated transition probability.

The scenario shown in Fig. 3B corresponds to a medium step change of the unknown transition probability p_{ij} from 0.3 to 0.5. iCPD detects the change and triggers the iCPD analysis (x^t = 976). As expected, the change is detected later than the big step change in scenario A, because the term n_{ij}/n_i needs more data (and consequently more time) after the change point to accumulate to a level that leads to a positive degree of prior-data conflict, i.e. $\Delta^{(n_i)} > 0$ in (10). Similarly to scenario A, the estimated change point $\hat{x} = 492$ is very close to the actual change point $x = 500$ (Fig. 4B#1) and the new IPSP prior interval $\left[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}\right] = [0.492, 0.527]$ captures the new unknown transition probability $b = 0.5$ (Fig. 4B#2).

In contrast to the previous step changes, the small step change in scenario C (Fig. 3C) does not trigger the iCPD analysis. This behaviour occurs because as new observations are made, the term n_{ij}/n_i is asymptotic to the upper bound of the IPSP estimated interval $\overline{p_{ij}}^{(0)} = 0.4$ – but very unlikely to exceed it (because n_{ij}/n_i also averages the data collected before x which is concentrated on $a = 0.3$). Hence, no prior-data conflict is detected in (10).

In scenario D (Fig. 3D), iCPD correctly detected both changes corresponding to the normal recovery pattern triggering the iCPD analysis at $x^t = 623$ and $x^t = 1358$ and accurately estimating change points $\hat{x} = 499$ (Fig. 4D#1) and $\hat{x} = 896$ (Fig. 4D#3), and the new IPSP prior intervals (Fig. 4D#2 and Fig. 4D#4)). Although scenario E (Fig. 3E) follows a similar square pattern, the quick recovery prevents iCPD from identifying the prior-data conflict and triggering the analysis. We evaluate in RQ2 how adjusting the iCPD hyperparameters, cf. (3) and (9), can increase the iCPD "sensitivity" and enable the detection of small and ephemeral changes.

The steep and gradual ramp scenarios in Figs. 3F and 3G, respectively, demonstrate the competency of iCPD to cope adequately with this class of changes. More specifically, iCPD performs particularly well in the steep ramp scenario triggering the iCPD analysis at $x^t = 750$ and calculating close approximations of the change point $\hat{x} = 545$ (Fig. 4F#1) and new IPSP prior intervals (Fig. 4F#2). The gradual ramp scenario is more difficult due to the (unknown) smaller slope which entails more time to reach the final value and complete the change. Since ramp changes (especially those with a gradual structure) do not by definition constitute sudden changes [1, 32], the observed behaviour is expected and does not violate the statistical model underpinning iCPD. Extending iCPD with support for more accurate analysis of ramp changes is planned for future work.

In scenario H (Fig. 3H), we examined how the timing of a change affects the iCPD accuracy by introducing early in the execution ($x = 100$), the same big step change as in scenario A. The estimated change point $\hat{x} = 93$ (Fig. 4H1) is fairly accurate. Due to the significantly smaller number of observations available during the iCPD analysis step ($x^t = 131$), the new IPSP prior interval is unsurprisingly wider compared to that from scenario A. Even though the change occurs early, the end of the scenario finds the IPSP estimator alone (i.e., without change-point detection capability) incapable to adapt its interval to include the new parameter value (*dotted* lines in Fig. 3H). This result once more shows the usefulness of iCPD.

Finally, we consider the scenario where the unknown transition parameter value is fixed throughout the scenario (Fig. 3I). As expected, the IPSP interval keeps decreasing and no iCPD analysis is triggered. Enforcing iCPD to run at the end of the scenario resulted in a nearly uniform posterior distribution of x (Fig. 4I#1) signifying the "high uncertainty" for the location of the change point. Hence, we have evidence that iCPD is not affected by false positives.

Table 1: Configurations and iCPD results over nine iDTMC scenarios. In all scenarios, the IPSP prior parameters are set to $[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}] = [0.2, 0.4]$ and $[\underline{n}^{(0)}, \overline{n}^{(0)}] = [10, 300]$. The scenario ID is associated with the corresponding subfigure in Fig 3 (Notations – x: change point; x^t: iCPD trigger point; $\hat{x}$: iCPD estimated change point; k/l: ratio of accepted samples over the total number of MCMC trials; $\langle x_1, \ldots, x_k \rangle$: MCMC sample sequence of the change point; $\langle b_1, \ldots, b_k \rangle$: sample sequence of transition probability b; N/A: iCPD not triggered).

ID	Pattern	(unk.) actual values			iCPD analysis					
#		a	b	x	x^t	$\hat{x}$	k/l	$\langle x_1, \ldots, x_k \rangle$	$\langle b_1, \ldots, b_k \rangle$	time (ms)
A	Step (big)	0.3	0.7	500	690	504	1733/5k	Fig. 4A#1	Fig. 4A#2	445
B	Step (medium)	0.3	0.5	500	976	492	2578/5k	Fig. 4B#1	Fig. 4B#2	414
C	Step (small)	0.3	0.4	500	N/A					
D	Square (normal recovery)	0.3	0.7	500, 900	623, 1358	499, 896	1571/5k, 4169/10k	Fig. 4D#1,3	Fig. 4D#2,4	392, 810
E	Square (quick recovery)	0.3	0.7	500, 600	N/A					
F	Ramp (steep)	0.3	0.7	$[500, 600]$	750	544	3987/10k	Fig. 4F#1	Fig. 4F#2	790
G	Ramp (gradual)	0.3	0.7	$[500, 900]$	1003	651	4046/10k	Fig. 4G#1	Fig. 4G#2	862
H	Step (big and early)	0.3	0.7	100	131	93	2082/5k	Fig. 4H#1	Fig. 4H#2	402
I	Fixed	0.3	0.3	N/A	800 (man.)	N/A	3386/5k	Fig. 4I#1	Fig. 4I#2	398

We provide a similar set of scenarios for iCTMCs on our project webpage at https://github.com/x-y-zhao/iCPD that demonstrate very similar results. Given these experimental results, we have sufficient evidence to conclude that iCPD can support the detection of sudden changes in the transition parameters of iDTMCs and iCTMC under several change patterns and steer the estimation of new IPSP prior intervals that enclose the new parameter value.

RQ2 (Configurability). The sensitivity of the iCPD triggering Eq. (9) can be configured either by modifying the prior parameters of the IPSP estimator in (2) or by assigning different importance levels to observations $o_1, o_2, \ldots, o_w$ based on temporal conditions. To answer this research question and assess whether iCPD can support the detection of small and/or transient changes, we evaluated how iCPD performs under different sensitivity configurations.

First, we introduced a sliding time window based on which iCPD discards old observations and operates using only observations falling within this time window. We evaluated iCPD enhanced with a time window in scenario C that includes a small step change (cf. row C in Table 1) and for which the standard iCPD did not detect a conflict (cf. Fig 3C). Using a time window of size 500, this iCPD variant detected the prior-data conflict at $x^t = 924$ triggering the iCPD analysis and calculating new IPSP prior intervals that include the updated transition probability (Fig 5A). By considering the 500 most recent observations, the term n_{ij}/n_i in (10) is influenced more by these observations leading to a positive degree of prior-data conflict. Similar reasoning has been applied in [10] which assigns ageing information to observations.

Second, we specified narrower IPSP prior intervals by setting $[\underline{p_{ij}}^{(0)}, \overline{p_{ij}}^{(0)}] = [0.25, 0.35]$ and $[\underline{n}^{(0)}, \overline{n}^{(0)}] = [5, 150]$ and evaluated iCPD in scenario E (row E in Table 1) that corresponds to a quick recovery. Compared to the iCPD configuration used in RQ1, under which both changes in scenario E were missed (Fig. 3), the narrower IPSP prior intervals enabled iCPD to detect the first change despite its short duration (occurring in observations o_{500}–o_{600}) and accurately estimate the change point ($\hat{x}=498$). Given the sparse observations about the new unknown transition parameter after the first change $p_{ij} = 0.7$, the new IPSP prior intervals are unsurprisingly wider than usual; this finding aligns with our results in scenario H, cf. Fig. 3H. The wide IPSP prior intervals and sparse observations are the primary reasons causing iCPD to miss the second change (i.e., the recovery). CPD using sparse information is widely acknowledged as a very challenging problem [1, 32]. Investigating how iCPD can handle more accurately changes of this type is part of our future work.

Finally, we evaluated an iCPD variant that employs both a time window and narrower IPSP prior intervals using scenario I (cf. Fig 3). Although in this scenario the value of the unknown transition parameter is fixed (and since no prior-data conflict occurs no iCPD analysis should be triggered), this iCPD variant is proven very sensitive (Fig. 5C). More specifically, this iCPD variant is susceptible to small fluctuations in term n_{ij}/n_i in (10) leading to four false positives, i.e., incorrectly triggering the iCPD analysis. In Fig. 5C, we also show the width of the IPSP interval and circle the times when these false alarms were triggered. Since the IPSP interval almost always enclosed the unknown transition probability, these false positives had only a minor effect on the overall iCPD behaviour.

These findings clearly indicate that increasing the sensitivity of iCPD can enable the detection of small and ephemeral changes. Nevertheless, this should be handled with caution to achieve a balanced trade-off between false positives and false negatives.

RQ3 (Efficiency). We answer this research question by analysing the overheads over the steps comprising the iCPD approach (Section 3.2). The IPSP estimation and the iCPD triggering steps use the closed-form formulas in (3) and (9) whose computational costs are constant and negligible. Similarly, the retrospective IPSP estimation step applies the closed-form formula of the IPSP estimator in (3) to the observations $o_{\hat{x}}$, $o_{\hat{x}+1}$, etc. As we have shown in RQ1 and RQ2 (cf. Table 1 and Fig. 3), the length of each sequence of observations, given by $x^t - \hat{x}$, is typically small. Hence, the final iCPD step has also insignificant overheads.

Only the iCPD analysis step requires further overheads investigation due to using MCMC [17] to generate the sample sequence

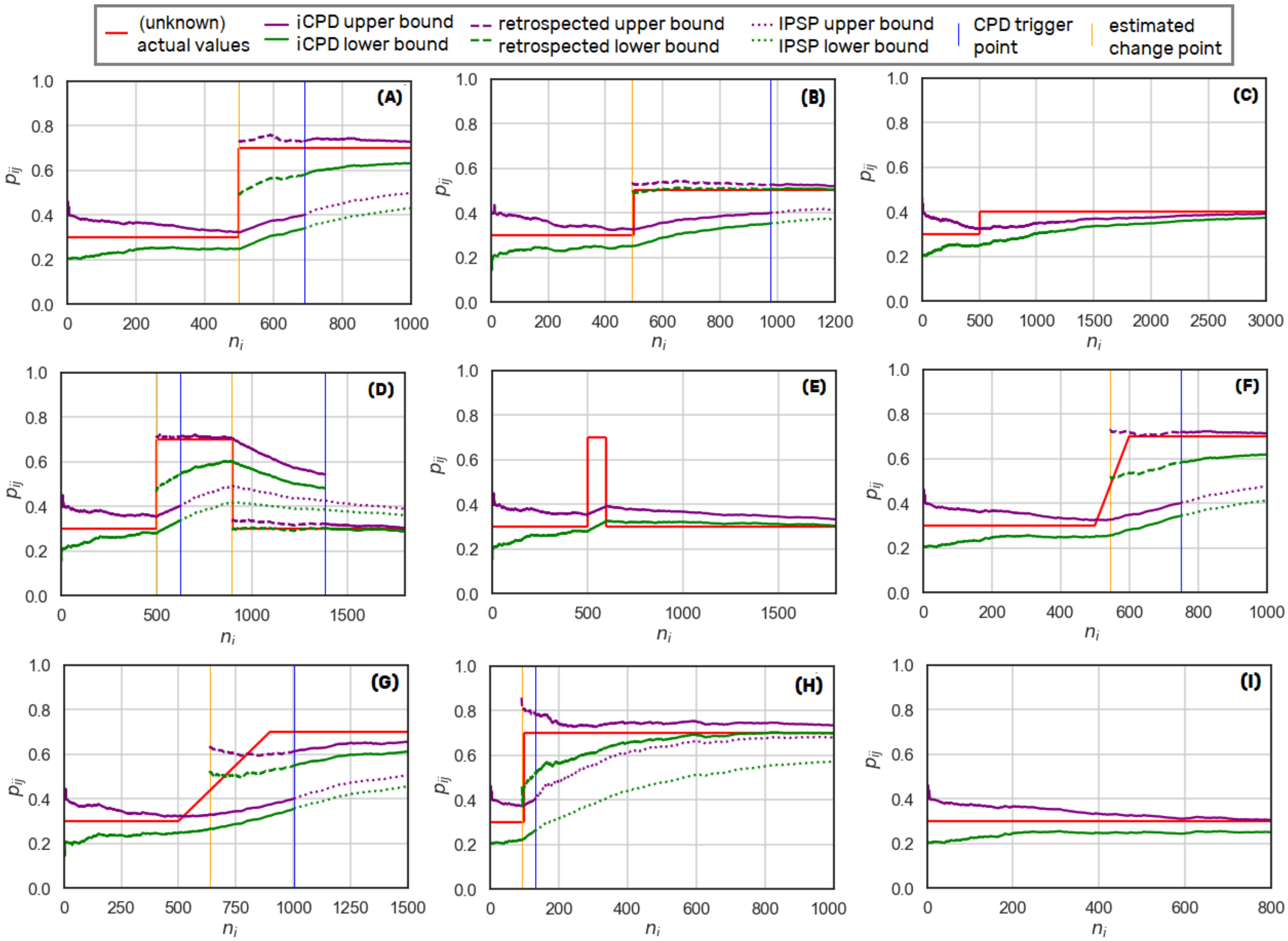

Figure 3: iCPD results over nine iDTMC scenarios with the configuration shown in Table 1 instrumented with the change patterns: (A) big step, (B) medium step, (C) small step, (D) square – normal recovery, (E) square – quick recovery, (F) steep ramp, (G) gradual ramp, (H) early big step and (I) fixed.

in (14). To quantify the overheads of this step, we replicated scenarios A and B (cf. Table 1) and varied the number of conducted MCMC trials $\in [0, 15000]$, while tuning the MCMC proposal distribution to keep the MCMC acceptance rate around 0.20–0.24 which is the MCMC diagnostics step [16] (Fig. 6 (right)). As shown in Fig. 6 (left), the computation time increases linearly with the number of MCMC trials and consumes more than 1 second only when the sampling size exceeds 13000. The rightmost column in Table 1 shows the time consumed by iCPD to execute its steps when a change has been detected (with a maximum of 10000 MCMC samples per change). Irrespective of the scenario and change pattern, the overheads are always below 1 second.

Similarly to [19], we assessed the accuracy of iCPD by measuring the estimation error E of the change point given by $E = \frac{|ec-rc|}{rc}$, where ec and rc are the estimated and real change points, respectively. In both scenarios, the E is relatively stable and smaller than 0.02 after 2000 MCMC trials; Fig. 6 (middle). As expected, the more MCMC trials conducted the more precise the iCPD estimation. Since in all our scenarios the minimum number of MCMC trials was 5000, we have evidence that iCPD has a fairly small change-point estimation error.

RQ4 (Verification support). We combined iCPD with the probabilistic model checker Prism-PSY [37] to establish if this integrated solution can support the effective PMC of the iCTMC of an autonomous underwater vehicle (AUV) by accurately detecting change points and estimating the updated transition rate intervals.

This iCTMC model, which has been used in related research [23, 31], models an AUV equipped with $n \geq 1$ sensors that can make observations of an oceanic parameter (e.g., salinity). The n sensors can be switched on and off individually (e.g., to save battery power when not used). When sensor i is switched on, it makes observations of the oceanic parameter with unknown and variable operating rate r_i. During operation, the AUV must adapt to changes

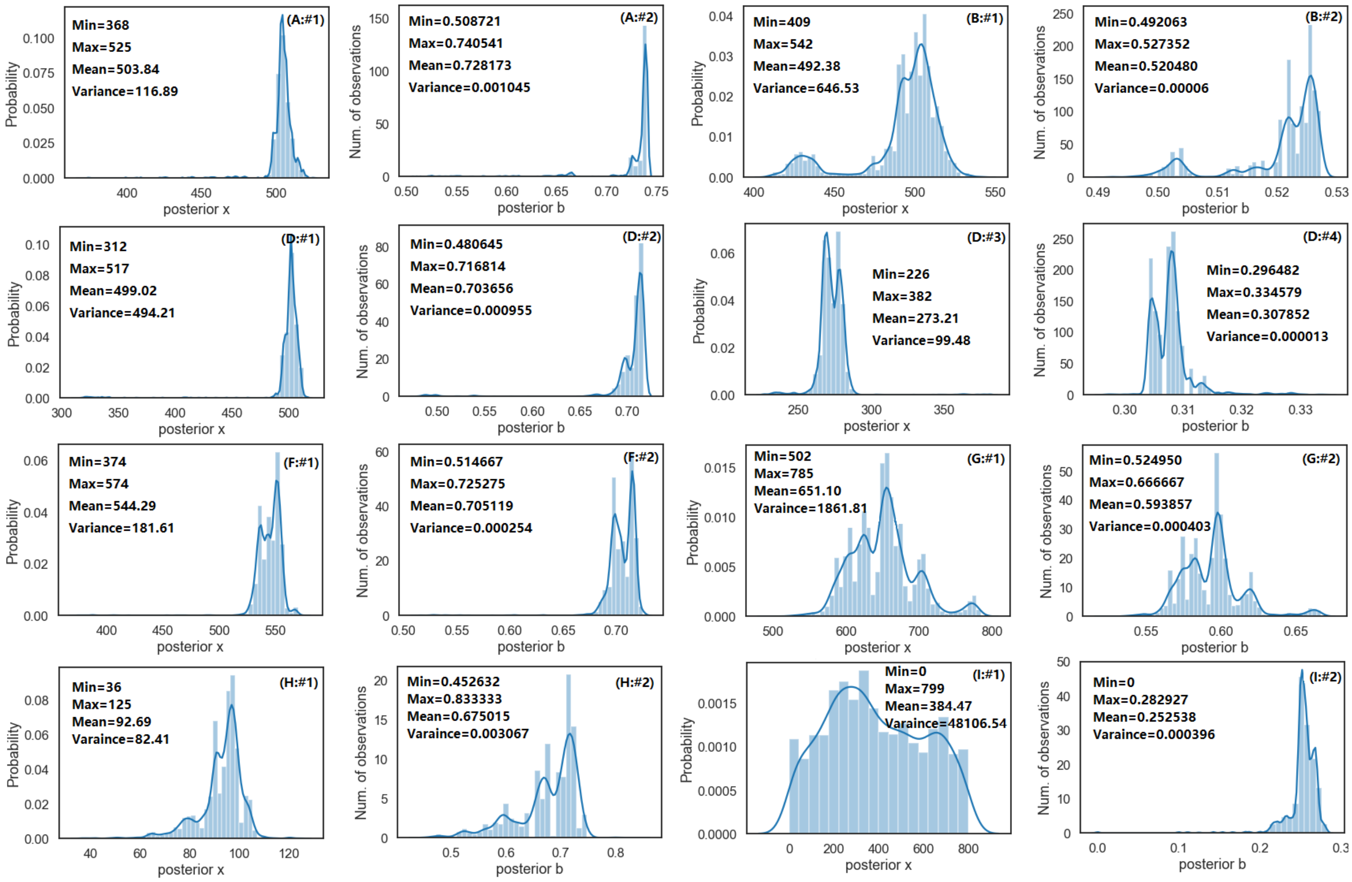

Figure 4: x and b posterior distributions corresponding to scenarios in Table 1 approximated by histograms of MCMC samples.

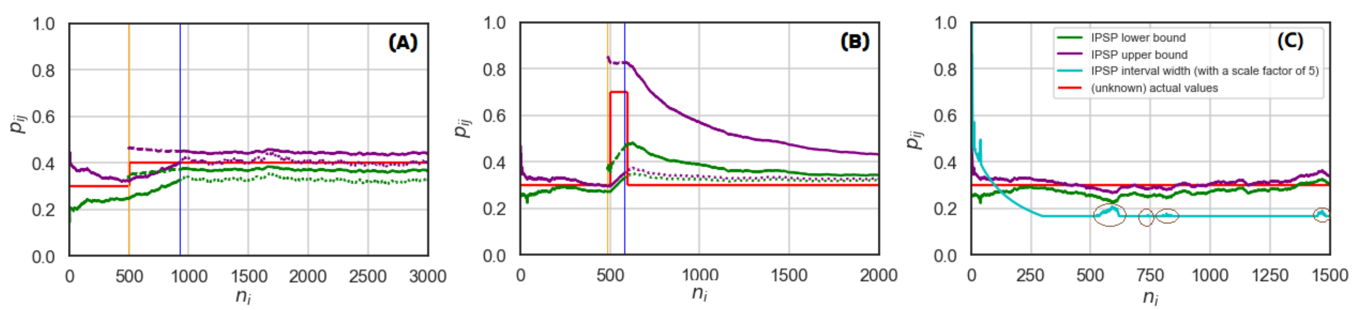

Figure 5: iCPD results with different sensitivity configurations. iCPD with time window (left), with narrower IPSP prior intervals (middle), and with both time window and narrower IPSP prior intervals (right) used in scenarios C, E, and I, respectively.

in the operating rate r_i of its sensors due to failure or service degradation by adjusting its speed and sensor configurations so that the following requirements are satisfied at all times[7]:

R1: An active sensor must make at least 20 observations of sufficient accuracy per 10 surveyed metres.

R2: The energy consumed by each sensor should not exceed 120 Joules per 10 surveyed metres.

[7]We only use the set of requirements needed to demonstrate the combined use of iCPD and Prism-PSY. For the complete set of system-level requirements, see [23].

Fig. 7 (left) shows the estimated IPSP transition rate interval that accurately captures the actual operating rate of the i-th sensor. After 300 seconds of operation, the AUV experiences a sudden service degradation of its currently active i-th sensor that reduced the sensor's operating rate from 6Hz to 2Hz. iCPD detects this change at $x^t = 358$ and triggers the iCPD analysis leading to the correct estimation of the change point $\hat{x} = 299$ and the accurate calculation of the revised IPSP prior interval that encloses the updated sensor operating rate. The revised transition rate interval is employed for

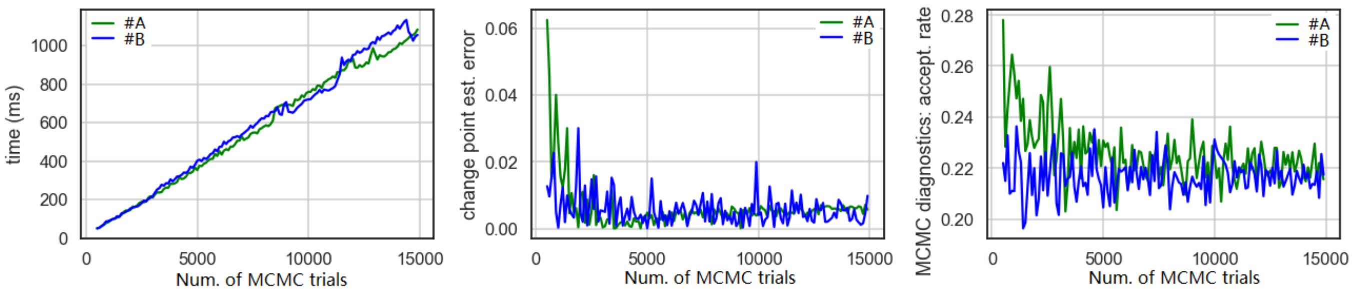

Figure 6: iCPD overheads in scenarios A and B showing computation time (left), change point estimation error (middle), and MCMC diagnostics acceptance rate [17] (right) over the number of conducted MCMC trials.

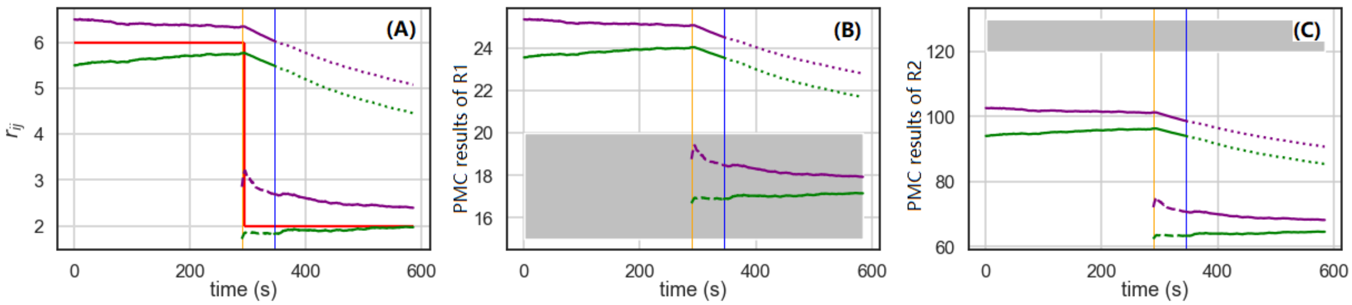

Figure 7: In an AUV surveillance mission with a sudden change of sensor rate (from 6Hz to 2Hz), iCPD results of the sensor rate (left) and PMC results of mission requirements R1 (middle) and R2 (right) (reusing the legends of Fig. 3 with shaded areas showing property violation).

the verification of requirements R1 and R2 over the iCTMC using Prism-PSY. Although requirement R2 is always met (Fig. 7 (right)), the reduced sensor rate results in the violation of R1, depicted by the shaded area in Fig. 7 (middle), requiring the AUV controller to select another sensor configuration that meets both R1 and R2 for all active sensors. To simplify the presentation, we do not show this reconfiguration task; for further information about this task, see [23]. iCPD can also establish the period during which requirement R1 was violated (i.e., $x^t - \hat{x}$) enabling the AUV controller to identify the region in which insufficient observations have been made, thus instructing the AUV to revisit this region and make additional observations to meet R1 for the region.

4.4 Threats to validity

Construct validity threats may arise due to simplifications and assumptions made when designing the evaluation methodology and instrumenting the investigated scenarios with the changes described in Section 4.2. To mitigate this threat, we devised changes that conform to four classes of change patterns (i.e., Step, Square, Ramp, Fixed) widely studied in the literature [10, 19, 22]. The iCTMC model of the AUV system used in to answer RQ4 has also been used in related research [11, 23, 31].

Internal validity threats may correspond to bias in establishing cause-effect relationships in our experiments. We limit them by examining instantiations of the four classes of change patterns for multiple values of the unknown transition parameters a and b (cf. Proposition 3.1), and for multiple pattern configurations concerning the change duration (Table 1). We reduce further the risk of biased results due to using a fine-tuned iCPD by comparing it against a CPD-agnostic IPSP estimator, showing that although the intervals of both estimators typically become narrower with new observations, the CPD-agnostic estimator is in most scenarios unable to adapt its interval to include the new parameter value. We also performed experiments with varied numbers of MCMC trials, showing that more MCMC trials lead to better estimates of the change point $\hat{x}$, thus conforming to the Bayesian practice [17]. Finally, we enable replication by making all experimental results publicly available on our project webpage https://github.com/x-y-zhao/iCPD.

External validity threats might be due to difficulties in obtaining the sequence of observations $o_1, ..., o_{n_i}$ as defined in (4). We limit this threat by devising iCPD to work in running systems enhanced with closed-loop controls (e.g., MAPEK-K [25]), which include a monitor component that continually monitors the system and records data about its behaviour. Another threat might occur if the sensitivity level of iCPD is inapplicable for the target system. We mitigated this by demonstrating how configuring the iCPD hyper-parameters (e.g., IPSP estimator prior intervals, monitoring window w size) enables achieving the desired trade-off between false positives and false negatives. Finally, to further reduce the risk that iCPD might be difficult to use in practice, we validated it both using iDTMCs and iCTMCs, and showed in RQ4 how it can be integrated with the probabilistic model checker Prism-PSY to verify key system properties (cf. Fig. 2). Nevertheless, additional experiments are needed to establish the applicability and feasibility of iCPD in interval Markov chains other than those used in our evaluation.

5 RELATED WORK

CPD analysis has been widely studied and successfully applied in many areas, including software engineering [14], climate change [5] and medicine [29]. From the numerous types of CPD solutions developed by this research (likelihood ratio methods, kernel-based methods, etc. – see [1] for a survey), iCPD falls into the category of *hierarchical Bayesian models* (HBMs) [12], i.e., CPD methods that rely on MCMC techniques to calculate change-point posteriors.

However, existing HBM methods employ MCMC each time a new data point becomes available, which is too computationally expensive for online analysis. This is also true about [19], which – to the best of our knowledge – is the only other project that has tackled CPD for (discrete-time) Markov chain parameters. In contrast, iCPD use a lightweight trigger to decide when this MCMC-based CPD analysis is needed, and can therefore be much more efficiently used for online CPD.

Additionally, current HBM methods (including [19]) compute point estimates for parameters affected by sudden changes. As emphasised in [6], point estimates of (uncertain) parameters can rarely be justified in practice. Unlike these methods, which it complements, iCPD focuses on interval CPD, computing new intervals of priors that support the robust Bayesian estimation of the uncertain parameters after the detected change point.

Bayesian methods to learn the transition parameters of Markov chains at runtime have been proposed in [18, 21], and a lightweight adaptive filter is introduced in [22] to reduce noise and provide smooth estimates. However, those approaches do not consider change-points explicitly, and need a relatively long time to make accurate estimates after sudden changes. In addition, these approaches yield point estimates that can be affected by unquantified and potentially significant errors. The work in [9] is the first to synthesise bounds for unknown transition probabilities of DTMCs, based on the frequentist theory of simultaneous confidence intervals. The only Bayesian approach to computing bounded estimates for PMC that we are aware of is [35]. Our iCPD builds on this robust Bayesian estimator and, to the best of our knowledge, is the first that provides CPD analysis for PMC with interval Markov models.

6 CONCLUSION & FUTURE WORK

We introduced iCPD, an end-to-end Bayesian approach to change-point detection and estimation of interval Markov chain parameters. Our experimental evaluation comprising scenarios instrumented with changes from four widely studied classes of change patterns showed that (i) iCPD can detect different types of changes accurately and efficiently; (ii) adjusting the iCPD hyper-parameters can enable achieving different trade-offs between false alarms and missed changes; and (iii) iCPD supports the effective runtime PMC of systems affected by parametric uncertainty. As future work, we plan to extend iCPD with support for other patterns of change (e.g., waves, triangles), and to investigate principled mechanisms of eliciting the IPSP priors.

ACKNOWLEDGMENTS

This work is supported by the UK EPSRC (through the Offshore Robotics for Certification of Assets [EP/R026173/1] and its PRF project COVE) and the Assuring Autonomy International Programme.

REFERENCES

[1] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51, 2 (2017), 339–367. https://doi.org/10.1007/s10115-016-0987-z
[2] Adnan Aziz, Kumud Sanwal, Vigyan Singhal, and Robert Brayton. 1996. Verifying continuous time Markov chains. In *Computer Aided Verification (LNCS)*, Rajeev Alur and Thomas A. Henzinger (Eds.), Vol. 1102. Springer Berlin Heidelberg, 269–276. https://doi.org/10.1007/3-540-61474-5_75
[3] Christel Baier and Joost-Pieter Katoen. 2008. *Principles of model checking*. MIT press.
[4] Christel Baier, Joost-Pieter Katoen, and Holger Hermanns. 1999. Approximative Symbolic Model Checking of Continuous-Time Markov Chains. In *CONCUR'99 Concurrency Theory (LNCS)*, Jos C. M. Baeten and Sjouke Mauw (Eds.), Vol. 1664. Springer, Berlin, Heidelberg, 146–161. https://doi.org/10.1007/3-540-48320-9_12
[5] Claudie Beaulieu, Jie Chen, and Jorge L Sarmiento. 2012. Change-point analysis as a tool to detect abrupt climate variations. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370, 1962 (2012), 1228–1249.
[6] James O Berger, Elías Moreno, Luis Raul Pericchi, M Jesús Bayarri, José M Bernardo, Juan A Cano, Julián De la Horra, Jacinto Martín, David Ríos-Insúa, Bruno Betrò, et al. 1994. An overview of robust Bayesian analysis. *Test* 3, 1 (1994), 5–124. https://doi.org/10.1007/BF02562676
[7] Jose M. Bernardo and Adrian F. M. Smith. 1994. *Bayesian theory*. Wiley.
[8] Andrea Bianco and Luca de Alfaro. 1995. Model checking of probabilistic and nondeterministic systems. In *Foundations of Software Technology and Theoretical Computer Science (LNCS)*, P. S. Thiagarajan (Ed.), Vol. 1026. Springer Berlin Heidelberg, 499–513. https://doi.org/10.1007/3-540-60692-0_70
[9] R. Calinescu, C. Ghezzi, K. Johnson, Pezzé M., Y. Rafiq, and G. Tamburrelli. 2016. Formal verification with confidence intervals to establish quality of service properties of software systems. *IEEE Transactions on Reliability* 65, 1 (2016), 107–125. https://doi.org/10.1109/TR.2015.2452931
[10] Radu Calinescu, Yasmin Rafiq, Kenneth Johnson, and Mehmet Emin Bakır. 2014. Adaptive model learning for continual verification of non-functional properties. In *Proc. of the 5th ACM/SPEC Int. Conference on Performance Engineering (ICPE '14)*. ACM, New York, NY, USA, 87–98. https://doi.org/10.1145/2568088.2568094
[11] R. Calinescu, D. Weyns, S. Gerasimou, M. U. Iftikhar, I. Habli, and T. Kelly. 2018. Engineering trustworthy self-adaptive software with dynamic assurance cases. *IEEE Transactions on Software Engineering* 44, 11 (Nov. 2018), 1039–1069. https://doi.org/10.1109/TSE.2017.2738640
[12] Bradley P. Carlin, Alan E. Gelfand, and Adrian F. M. Smith. 1992. Hierarchical Bayesian analysis of change-point problems. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 41, 2 (1992), 389–405. https://doi.org/10.2307/2347570
[13] Souymodip Chakraborty and Joost-Pieter Katoen. 2015. Model checking of open interval Markov chains. In *Analytical and Stochastic Modelling Techniques and Applications (LNCS)*, Marco Gribaudo, Daniele Manini, and Anne Remke (Eds.), Vol. 9081. Springer, Cham, 30–42. https://doi.org/10.1007/978-3-319-18579-8_3
[14] David Daly, William Brown, Henrik Ingo, Jim O'Leary, and David Bradford. 2020. The use of change point detection to identify software performance regressions in a continuous integration system. In *Proc. of the ACM/SPEC International Conference on Performance Engineering (ICPE '20)*. ACM, New York, NY, USA, 67–75. https://doi.org/10.1145/3358960.3375791
[15] Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. 2017. A Storm is coming: A modern probabilistic model checker. In *Computer Aided Verification (LNCS)*, Rupak Majumdar and Viktor Kunčak (Eds.), Vol. 10427. Springer, Cham, 592–600. https://doi.org/10.1007/978-3-319-63390-9_31
[16] Therese M. Donovan and Ruth M. Mickey. 2019. MCMC diagnostic approaches. In *Bayesian Statistics for Beginners*. Oxford University Press, Oxford. https://doi.org/10.1093/oso/9780198841296.003.0014
[17] Therese M. Donovan and Ruth M. Mickey. 2019. The white house problem revisited: MCMC with the Metropolis–Hastings algorithm. In *Bayesian Statistics for Beginners*. Oxford University Press, Oxford. https://doi.org/10.1093/oso/9780198841296.003.0015
[18] Ilenia Epifani, Carlo Ghezzi, Raffaela Mirandola, and Giordano Tamburrelli. 2009. Model evolution by run-time parameter adaptation. In *Proc. of the 31st International Conference on Software Engineering (ICSE '09)*. IEEE, Washington, DC, USA, 111–121. https://doi.org/10.1109/ICSE.2009.5070513
[19] Ilenia Epifani, Carlo Ghezzi, and Giordano Tamburrelli. 2010. Change-point detection for black-box services. In *Proc. of the 18th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE '10)*. ACM, New York,

NY, USA, 227–236. https://doi.org/10.1145/1882291.1882326
[20] Michael Evans and Hadas Moshonov. 2006. Checking for prior-data conflict. *Bayesian Analysis* 1, 4 (2006), 893–914. https://doi.org/10.1214/06-BA129
[21] Antonio Filieri, Carlo Ghezzi, and Giordano Tamburrelli. 2012. A formal approach to adaptive software: Continuous assurance of non-functional requirements. *Formal Aspects of Computing* 24, 2 (2012), 163–186. https://doi.org/10.1007/s00165-011-0207-2
[22] Antonio Filieri, Lars Grunske, and Alberto Leva. 2015. Lightweight adaptive filtering for efficient learning and updating of probabilistic models. In *Proc. of the 37th International Conference on Software Engineering (ICSE '15)*. IEEE, Florence, Italy, 200–211. https://doi.org/10.1109/ICSE.2015.41
[23] Simos Gerasimou, Radu Calinescu, Stepan Shevtsov, and Danny Weyns. 2017. UNDERSEA: an exemplar for engineering self-adaptive unmanned underwater vehicles. In *IEEE/ACM 12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems*. Buenos Aires, Argentina, 83–89. https://doi.org/10.1109/SEAMS.2017.19
[24] Hans Hansson and Bengt Jonsson. 1994. A logic for reasoning about time and reliability. *Formal Aspects of Computing* 6, 5 (Sept. 1994), 512–535. https://doi.org/10.1007/BF01211866
[25] Jeffrey O Kephart and David M Chess. 2003. The vision of autonomic computing. *Computer* 36, 1 (2003), 41–50.
[26] Igor O. Kozine and Lev V. Utkin. 2002. Interval-valued finite Markov chains. *Reliable Computing* 8, 2 (2002), 97–113. https://doi.org/10.1023/A:1014745904458
[27] Marta Kwiatkowska, Gethin Norman, and David Parker. 2004. Modelling and verification of probabilistic systems. *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems. CRM Monograph Series* 23 (2004), 93–215.
[28] Marta Kwiatkowska, Gethin Norman, and David Parker. 2011. PRISM 4.0: Verification of probabilistic real-time systems. In *Computer Aided Verification (LNCS)*, Ganesh Gopalakrishnan and Shaz Qadeer (Eds.), Vol. 6806. Springer, Berlin, Heidelberg, 585–591. https://doi.org/10.1007/978-3-642-22110-1_47
[29] Siqi Liu, Adam Wright, and Milos Hauskrecht. 2018. Change-point detection method for clinical decision support system rule monitoring. *Artificial Intelligence in Medicine* 91 (2018), 49 – 56. https://doi.org/10.1016/j.artmed.2018.06.003
[30] Koushik Sen, Mahesh Viswanathan, and Gul Agha. 2006. Model-checking Markov chains in the presence of uncertainties. In *Tools and Algorithms for the Construction and Analysis of Systems (LNCS)*, Holger Hermanns and Jens Palsberg (Eds.), Vol. 3920. Springer, Berlin, Heidelberg, 394–410. https://doi.org/10.1007/11691372_26
[31] Stepan Shevtsov and Danny Weyns. 2016. Keep It SIMPLEX: Satisfying multiple goals with guarantees in control-based self-adaptive systems. In *Proc. of the 2016 24th ACM SIGSOFT Int. Symp. on Foundations of Software Engineering (FSE 2016)*. ACM, New York, NY, USA, 229–241. https://doi.org/10.1145/2950290.2950301
[32] Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing* 167 (2020). https://doi.org/10.1016/j.sigpro.2019.107299
[33] Gero Walter and Thomas Augustin. 2009. Imprecision and prior-data conflict in generalized Bayesian inference. *Journal of Statistical Theory and Practice* 3, 1 (2009), 255–271. https://doi.org/10.1080/15598608.2009.10411924
[34] S Zacks. 1983. Survey of classical and Bayesian approaches to the change-point problem: fixed sample and sequential procedures of testing and estimation. In *Recent advances in statistics*. Elsevier, 245–269. https://doi.org/10.1016/B978-0-12-589320-6.50016-2
[35] Xingyu Zhao, Valentin Robu, David Flynn, Fateme Dinmohammadi, Michael Fisher, and Matt Webster. 2019. Probabilistic model checking of robots deployed in extreme environments. In *Proc. of the 33rd AAAI Conference on Artificial Intelligence*, Vol. 33. Honolulu, Hawaii, USA, 8076–8084. https://doi.org/10.1609/aaai.v33i01.33018066
[36] Xiaoqian Zhu, Yongjia Xie, Jianping Li, and Dengsheng Wu. 2015. Change point detection for subprime crisis in American banking: From the perspective of risk dependence. *International Review of Economics & Finance* 38 (2015), 18 – 28. https://doi.org/10.1016/j.iref.2014.12.011
[37] Milan Češka, Petr Pilař, Nicola Paoletti, Luboš Brim, and Marta Kwiatkowska. 2016. PRISM-PSY: Precise GPU-accelerated parameter synthesis for stochastic systems. In *Tools and Algorithms for the Construction and Analysis of Systems (LNCS)*, Marsha Chechik and Jean-François Raskin (Eds.), Vol. 9636. Springer, Berlin, Heidelberg, 367–384. https://doi.org/10.1007/978-3-662-49674-9_21