Radboud Repository
sharing science

Active Learning of Runtime Monitors Under Uncertainty

Junges, J.S.L.; Seshia, S.A.; Torfah, H.
2025, Article in monograph or in proceedings (Kosmatov, N. (ed.), Integrated Formal Methods: 19th International Conference, IFM 2024, Manchester, UK, November 13–15, 2024, Proceedings, pp. 297-306)
Doi link to publisher: https://doi.org/10.1007/978-3-031-76554-4_18

Version of the following full text: Publisher's version
Published under the terms of article 25fa of the Dutch copyright act. Please follow this link for the
Terms of Use: https://repository.ubn.ru.nl/page/termsofuse
Downloaded from: https://hdl.handle.net/2066/317539
Download date: 2026-02-28

Note:
To cite this publication please use the final published version (if applicable).



Active Learning of Runtime Monitors Under Uncertainty

Sebastian Junges[1], Sanjit A. Seshia[2], and Hazem Torfah[3(✉)]

[1] Radboud University, Nijmegen, Netherlands
sebastian.junges@ru.nl
[2] University of California at Berkeley, Berkeley, USA
sseshia@berkeley.edu
[3] Chalmers University of Technology and Gothenburg University, Gothenburg, Sweden
hazemto@chalmers.se

Abstract. We investigate the problem of active learning of runtime monitors for cyber-physical systems (CPS) under uncertainty. In CPS, runtime monitors need to make decisions with only partial information about the system state and cannot always rely on having a precise environment model. As a result, the learning process and resulting monitors must be able to handle this type of uncertainty. We present a framework for the active learning of monitors and discuss the challenges in implementing oracles for membership and equivalence queries. We particularly apply the framework to a setting where uncertainty models are defined by Markov decision processes. We present initial results demonstrating the efficacy of our approach in learning accurate monitors using a set of benchmarks from the domain of autonomous systems.

1 Introduction

Runtime monitors are crucial for maintaining the safety of cyber-physical systems (CPS) throughout their lifetime [3,4,14,20]. They play a vital role in detecting irregularities during runtime [12], and in evaluating the performance of the system [5,6]. Runtime monitors can also assist in maintaining situational awareness, especially in systems such as autonomous CPS [17]. Depending on the state of the system and its environment, monitors must determine if an autonomous system is within its operational design domain. If not, certain fail-safe plans need to be executed to keep the system safe. The key challenge in the construction of such monitors is to ensure that they are resilient to uncertainty in the information received about the environment. Monitors in a CPS perceive information about the environment from sensors, which are generally imperfect and may deliver data that does not accurately reflect the actual state of the environment.

In this paper, we investigate active learning of runtime monitors for CPS under uncertainty. We present a formalization of the problem and introduce a general framework for learning based on this formalization. We particularly discuss the challenges in implementing oracles for answering membership and

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2025
N. Kosmatov and L. Kovács (Eds.): IFM 2024, LNCS 15234, pp. 297–306, 2025.
https://doi.org/10.1007/978-3-031-76554-4_18

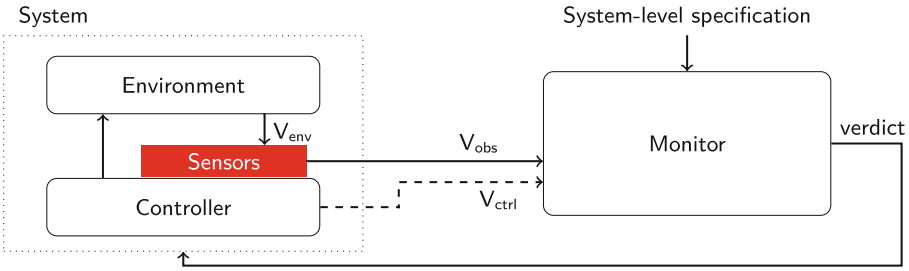


Fig. 1. A general monitoring architecture in cyber-physical systems.

equivalence queries and elaborate on the role of probabilistic and statistical verification methods in providing solutions for these problems.

Classically, runtime monitors are synthesized directly from (formal) temporal specifications [4]. The specifications are typically defined on the system level, i.e., over the joint state of system and environment. However, the monitor may only have partial observability into the current state, as shown in Fig. 1. A system-level specification is defined over the set of variables $\mathsf{V}_{\mathsf{sys}} = \mathsf{V}_{\mathsf{env}} \cup \mathsf{V}_{\mathsf{ctrl}}$, capturing both the state of the environment and control. A monitor, on the other hand, is defined over variables $\mathsf{V}_{\mathsf{obs}}$, capturing information about the environment as received via a set of sensors. Notice that the relation between both variables may be complex, in particular, there generally is no (deterministic) mapping between assignments to $\mathsf{V}_{\mathsf{sys}}$ and $\mathsf{V}_{\mathsf{obs}}$.

Monitoring for systems with uncertainty has been studied in a variety of settings, spanning from model-based [11,15,16,18] to simulations-based approaches [7,9,19,21]. Our formalization of the monitoring problem under uncertainty generalizes over both. In our experiments, we primarily, however, work in the context of the former. The models provide a ground truth, but computing the safety can only be done via, e.g., model checking on a large model and for a single sequence of observations at a time. Given key requirements on (1) promptness, i.e., monitors must be able to issue a verdict promptly, (2) complexity, i.e., monitors themselves are safety-critical software, which means that they must be sufficiently simple, and (3) efficiency, i.e., ideally using constant memory, it is unrealistic to deploy such algorithms on a device that must be monitored.

This paper presents a work-in-progress on using supervised, active learning to extract an artifact implementing a monitor that adheres to the requirements above and that is resilient to uncertainty in the observed data. In the scope of active learning, the model-based monitor approaches yield membership oracles. We often do not have access to efficient equivalence oracles. Instead, we use statistical conformance checks. The main contribution of this paper is a statistical sound approach to do so, based on Chernoff bounds and the Bonferroni correction method. The second contribution is an empirical evaluation of this approach, using membership oracles provided by previous work on monitoring Markov decision processes (MDP) [11]. Our preliminary evaluation demonstrates the applicability of this approach.

2 Motivating Example

Consider a CPS composed of an autonomous vehicle, the *ego*, and an environment that comprises other vehicles driving on the road. At all times the ego vehicle needs to satisfy a specification φ that requires it to keep a distance larger than m meters from any other vehicle ahead of it. A monitor is integrated into the ego car, continuously checking whether this distance is maintained. The monitor has no information about the concrete positions of other cars on the road and relies in its decision making on an image-based detection module that is able to provide the monitor with information about the distances of other objects on the road. Relating this to Fig. 1, the ground truth of the positions of other vehicles on the road is part of $\mathsf{V}_{\mathsf{env}}$, and not directly observable by the monitor, whereas the distances computed by the object detection module is part of the observable feature space $\mathsf{V}_{\mathsf{obs}}$. The accuracy of a monitor for checking φ will therefore depend on the accuracy of the output of the detection module and a good monitor will have to be robust for any imperfections in the received data.

3 The Monitor Generation Problem

Before formally defining our problem, we expand on a few aspects regarding the sources of uncertainty in cyber-physical systems (CPS). Generally, systems are subject to at least two sources of uncertainty: process uncertainty that makes the evolution of states uncertain, and measurement uncertainty that yields different observations even in the same state. In this work, we consider that these types of uncertainty can be captured with a distribution. A third notion of uncertainty is that the monitor may often not have access to a precise model of the system dynamics, e.g., due to lack of data. This third notion of uncertainty is a challenge to create monitors, but not part of the monitoring problem itself. We discuss handling this type of uncertainty in our experiments.

Based on the intuition above, we can formalize the problem as follows. We use the following notation: For a countable set X, let $\mathit{Distr}(X) \subset (X \to [0,1])$ define the set of distributions over X. A distribution $d \in \mathit{Distr}(X)$ particularly satisfies, $\sum_{x \in X} d(x) = 1$. We further, define the sets Σ_{sys} and Σ_{obs} to denote the set of valuations over the variable $\mathsf{V}_{\mathsf{sys}}$ and $\mathsf{V}_{\mathsf{obs}}$, respectively. We state the problem and then describe the ingredients.

Problem Statement. Given a distribution $d \in \mathit{Distr}(\Sigma^*_{\mathsf{sys}})$, a mapping $\mu \colon \Sigma^*_{\mathsf{sys}} \to \mathit{Distr}(\Sigma^*_{\mathsf{obs}})$, a risk measure $r \colon \Sigma^*_{\mathsf{sys}} \to \mathbb{R}$, and a constant λ, learn a monitor M such that

$$M(\sigma_{\mathsf{obs}}) \Leftrightarrow \sum_{\sigma_{\mathsf{sys}} \in \Sigma^*_{\mathsf{sys}}} d(\sigma_{\mathsf{sys}}) \cdot \mu(\sigma_{\mathsf{sys}})(\sigma_{\mathsf{obs}}) \cdot r(\sigma_{\mathsf{sys}}) > \lambda$$

In our problem formulation, we assume that a system-level behavior, i.e., traces σ_{sys}, occurs according to some distribution $d \in \mathit{Distr}(\Sigma^*_{\mathsf{sys}})$. The mismatch between the system-level behavior and observations, as well as the stochastic errors in sensors is captured by the probabilistic function μ. The risk function r

defines a quantitative specification, that associates each system-level trace with a risk. Note that a qualitative specification can be represented by a measure r that returns either 0 or 1, stating whether a system-level trace satisfies (lowest risk) or violates (highest risk) the specification, respectively. A monitor then accepts an observation trace σ_{obs} if the total risk, computed as the sum of weighted risks (weighted through the probability of occurrence measured in terms d and μ) of system-level traces that may induce σ_{obs}, is larger than the threshold λ.

In our motivating example, the risk r can be described as a function that produces higher risk the closer the ego car approaches another car. The measure μ defines the error of the detection module in capturing the true distances to other cars on the road. Lastly, d is a distribution over the actions of other cars on the road. In most cases, such distribution will be unknown and will have to be approximated, for example, by performing a worst-case analysis on the non-deterministic behavior of the environment, as we show later in our experiments.

4 Active Learning Under Uncertainty

Active learning describes a process where a concept is learned by actively querying an entity about this concept. In this paper, we are interested in an active learning setting, where learning a monitor, assumes access to an entity, an *oracle*, that can answer two types of queries: *membership* and *equivalence* queries. Membership queries are used to check whether a certain observation trace is accepted by a monitor. Equivalence queries check the correctness of a learned monitor with respect to the system-level specification. If an equivalence query fails, the counterexamples computed by the oracle are used to refine the training dataset and to guide the next round of membership queries.

4.1 Membership Queries

In membership queries, the goal is to check whether a certain observation trace $\sigma_{\mathsf{obs}} \in \Sigma^*_{\mathsf{obs}}$ is an element of the language of the goal monitor. Following the problem definition above, solving a membership query for a trace σ_{obs} boils down to checking the validity of the expression

$$\sum_{\sigma_{\mathsf{sys}} \in \Sigma^*_{\mathsf{sys}}} d(\sigma_{\mathsf{sys}}) \cdot \mu(\sigma_{\mathsf{sys}})(\sigma_{\mathsf{obs}}) \cdot r(\sigma_{\mathsf{sys}}) > \lambda \tag{1}$$

The evaluation of the above expression depends on the underlying uncertainty model, and how we can approximate the measures d and μ. In our experiments, we use an example showing how to approximate these measures for uncertainty models represented by Markov decision processes.

4.2 Equivalence Queries

Ideally, equivalence queries are solved by a model checker that checks the learned monitor against the system-level specification. This is often not feasible, as it can

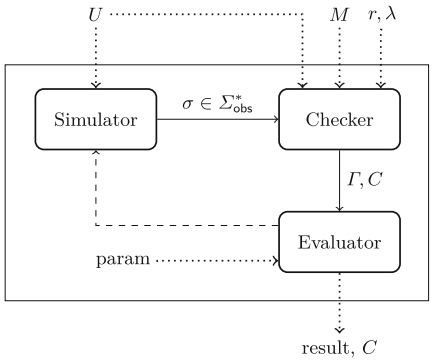


Fig. 2. A statistical verification approach implementing equivalence queries.

be intractable to compute or because we do not have a precise model. Instead, we will rely on conformance-based approaches that repeatedly pose membership queries and use statistical verification techniques [1] to obtain a statistical guarantee on the correctness of the monitor.

In statistical verification, we assume access to an executable system, which can be of stochastic nature, and a system-level specification. Using a (Monte Carlo) simulator, the system is executed repeatedly. Executions are checked against the specification, and a statistical result is computed aggregating the individual results.

Our statistical verification approach is depicted in Fig. 2. Given a system U (which can also be a black box), a learned monitor M, a risk function r, and a risk threshold λ, the statistical verification approach samples observation traces σ from U and checks whether the verdict of M over σ matches the result of a membership query defined in Eq. (1). After several sampling and checking rounds, the set of verdicts Γ is forwarded to an evaluator that computes a statistical verdict on the correctness of M. The evaluator can be implemented using statistical methods, such as hypothesis testing or estimation methods [1,10,13]. If M passes the statistical test, the active learning process terminates. Otherwise, a set of counterexample C is returned and used to expand the training set or guide the next learning iteration. Since monitors in each iteration are learned from a training set that includes (or is guided by) counterexamples collected in previous iterations, we lose the statistical independence and need to compensate for that by increasing the number of testing traces in each equivalence query round. A correction in the statistical parameters is therefore needed to compute a number of samples that retains statistical confidence over several learning iterations. An example of such a correction method is the Bonferroni method [8], which we will use in our experiments.

5 Preliminary Experimental Evaluation

We instantiate the active learning approach to one for learning monitors for uncertainty models given by Markov decision processes (MDPs) [11]. An MDP captures both the nondeterministic and stochastic behavior of system and environment. Particularly, the nondeterministic transitions in an MDP capture the part of the environment's behavior which is unknown (either due to the lack of data, or as a consequence of the discretization of the continuous state space). The stochastic transitions of an MDP capture any known probabilistic behavior of the environment, but foremost the stochastic error introduced by the imperfect sensors. Further, a risk function is given that associates each state in the MDP with a risk. For example, this could be the probability that our ego car in the motivating example will crash with another vehicle within a given number of steps.

In terms of our problem statement, if the nondeterminism were fixed by some policy, the MDP would become a labeled Markov chain (basically a hidden Markov model). This Markov chain can be thought of as a description of a distribution over paths $d \in \mathit{Distr}(\Sigma^*_{\mathsf{sys}})$. In the labeled model, every state maps to a distribution over observables, which can be lifted to a mapping $\mu\colon \Sigma^*_{\mathsf{sys}} \to \mathit{Distr}(\Sigma^*_{\mathsf{obs}})$. Finally, the risk measure $r\colon \Sigma^*_{\mathsf{sys}} \to \mathbb{R}$ is obtained by considering the last state of the sequence (the current state) and how likely it is to reach an error state from this state, i.e., by a reachability probability that can be computed for the Markov chain. The MDP with nondeterminism intuitively encodes that none of the functions are certain, but that they could be generated by any Markov chain that is induced by a policy of the MDP. The construction in [11] ensures that we overapproximate the left-hand side of Eq. (1), thereby creating a monitor that is overly cautious.

Below we explain our experimental setup, providing details on the used benchmark and the engines used to implement membership and equivalence queries.

5.1 Setup

In our experiments, we choose the class of decision trees to define the space of monitors. We use the scikit library[1] to learn decision trees with no bound on their depth. The inputs to the monitor were observation traces of a fixed length obtained from the benchmark MDPs. These ranged from traces of length 5 to 60. Our prototype was implemented in Python. All experiments where conducted on a machine with an M2 chip and with 16 GB of RAM.

Benchmarks. We applied our active learning framework on three benchmarks [11], defining MDP-based uncertainty models as described above:

– Airport: This benchmark defined a landing scenario with uncertainty about the observations of vehicles moving around the runway. The risk is defined

[1] https://scikit-learn.org.

by the probability of colliding with one of the on-ground vehicles. The MDP for this benchmark has 20910 states and a transition relation of size 114143.
- Evade: This benchmark defines a navigation task in a multi-agent setting in a grid. The robot moves randomly, and the risk is defined as the probability of crashing with another robot. The other robot has an internal incentive in the form of a cardinal direction, and nondeterministically decides to move or to uniformly randomly change its incentive. The MDP for this benchmark has 1001 states and a transition relation of size 5318.
- Refuel: This benchmark models a robot with a depleting battery moving on a grid with recharging stations. The world model consists of a robot moving randomly around in the grid with some dedicated charging cells The risk is to deplete the battery within a fixed horizon. The MDP for this benchmark has 45073 states and a transition relation of size 2431691.

Membership Queries. Membership queries are performed using PREMISE, a prototype tool for monitoring under uncertainty defined by Markov decision processes [11]. PREMISE computes the worst case risk over all possible policies of an MDP. It thus provides an overapproximates of the target risk in Eq. (1), as we described earlier. PREMISE comes with two algorithms for computing the risk. (1) A filtering approaches that instead of tracking the entire set of beliefs only tracks the vertices of a convex hull. (2) Another based on unrolling the MDP for the length of the observation trace and determining the risk by computing the maximal conditional reachability probability over the unrolling. In our experiments, we use the latter. Membership queries were (arbitrarily) bound to threshold equal to 0.5.

Conformance Testing. We used an estimation method based on the application of Chernoff bounds [1] and the Bonferroni correction methods. For a confidence value δ and an error margin ϵ, we can use a Chernoff bound to determine the number of samples n to obtain a monitor such that with confidence δ, the computed accuracy, $p' = \sum_{i=1}^{n} x_i/n$, where x_i represent the results of n different membership queries, does not deviate by more than ϵ from the true accuracy of the monitor. In all our experiments, we choose $\delta = 0.9$ and $\epsilon = 0.01$. The number of samples is determined by $n = \lceil ln(2\delta)/2\epsilon^2 \rceil$. We apply the Bonferonni correction by enlarging the size of the sample set in each round to $\delta' = \delta + (1 - \delta)/2$, similar to [2]. We ran our experiments for at most 10 iterations. The process terminates early when a monitor with accuracy >98% is found.

5.2 Results

Initial results are presented in Fig. 3. For all three benchmarks and trace length, we were able to obtain highly accurate monitors. Accuracy was measured in terms the number of samples, chosen during conformance testing, on which both the monitor and the oracle produced matching verdicts. We also report on the rates of false positives and false negatives. The rate of false positives was computed as the ratio of 1-misclassfication, i.e., where the monitor's verdict was

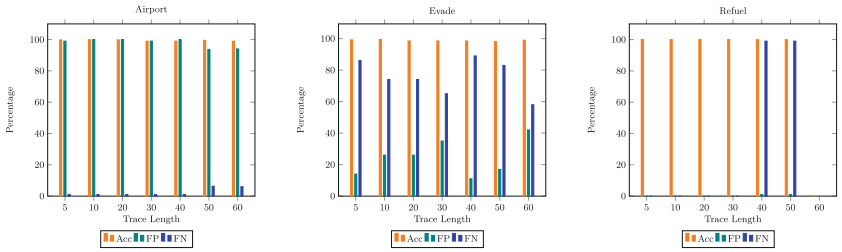


Fig. 3. Accuracy, false positives rates, and false negative rates (bars in the same order) of learned monitors for the Airport, Evade, and Refuel benchmarks. The results are presented for growing input trace lengths ranging from length 5 to 60. Statistical verification was performed using an estimation based on Chernoff bounds for values $\delta = 0.9$ and $\epsilon = 0.01$, and utilizing a Bonferroni correction. Note that the rates of false positives and negatives represent to ration with respect to the total number of misclassfications.

true and that of the oracle was false, to the total number of misclassifications. The rate of false negatives computes the complementary rate, that of the ratio of 0-misclassifications to the total number of misclassifications. Depending on the benchmark, we had variations in the false negatives and false positive rates. Note that in our current setting we are not biasing the learning towards any types of counterexamples, and thus the rates shown in our results may change if we choose to bias the training data refinement to one of the two. The number of iterations needed to learn the monitors depended on the uncertainty model. For the Evade benchmark we needed an average of 5 rounds. The average time for performing equivalence queries was about one minute testing more than 3000 samples. For larger benchmarks such as the Airport benchmark, and for traces of length 60, we required an average of 17 min to complete an equivalence query. For the Refuel benchmark, equivalence queries required less than 3min.

6 Conclusion

We investigated the problem of actively learning monitors under uncertainty. We provided a formalization for the problem and a framework for the active learning of monitors, describing challenges in implementing oracles for answering membership and equivalence queries. We further showed how the latter can be implemented for settings where uncertainty is defined by Markov decision processes, and demonstrated the potential of our methods using a prototypical implementation. With our paper, we intended to provide a foundation for defining monitor learning problems and a general framework for developing active learning approaches for different uncertainty models. In future work, we plan to investigate different instantiations of the framework and compare different statistical approaches to implementing membership and equivalence queries.

Acknowledgments. This work was partly supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation, by DARPA contracts FA8750-18-C-0101 (AA) and FA8750-23-C-0080 (TIAMAT), and by C3DTI.

References

1. Agha, G., Palmskog, K.: A survey of statistical model checking. ACM Trans. Model. Comput. Simul. **28**(1), 1–39 (2018). https://doi.org/10.1145/3158668
2. Ashok, P., Křetínský, J., Weininger, M.: PAC statistical model checking for Markov decision processes and stochastic games. In: Dillig, I., Tasiran, S. (eds.) CAV 2019. LNCS, vol. 11561, pp. 497–519. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-25540-4_29
3. Bartocci, E., et al.: Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications. In: Lectures on Runtime Verification (2018). https://api.semanticscholar.org/CorpusID:4539733
4. Bartocci, E., Falcone, Y., Francalanza, A., Reger, G.: Introduction to runtime verification. In: Bartocci, E., Falcone, Y. (eds.) Lectures on Runtime Verification - Introductory and Advanced Topics. LNCS, vol. 10457, pp. 1–33. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75632-5_1
5. Baumeister, J., Finkbeiner, B., Schirmer, S., Schwenger, M., Torens, C.: RTLola cleared for take-off: monitoring autonomous aircraft. In: Lahiri, S.K., Wang, C. (eds.) CAV 2020. LNCS, vol. 12225, pp. 28–39. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53291-8_3
6. Baumeister, J., Finkbeiner, B., Schwenger, M., Torfah, H.: FPGA stream-monitoring of real-time properties. ACM Trans. Embed. Comput. Syst. **18**(5s), 88:1–88:24 (2019).https://doi.org/10.1145/3358220
7. Cairoli, F., Bortolussi, L., Paoletti, N.: Neural predictive monitoring under partial observability. In: Feng, L., Fisman, D. (eds.) RV 2021. LNCS, vol. 12974, pp. 121–141. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88494-9_7
8. Dunn, O.J.: Confidence intervals for the means of dependent, normally distributed variables. J. Am. Statist. Assoc. **54**(287), 613–621 (1959). https://doi.org/10.1080/01621459.1959.10501524
9. Fremont, D.J., Chiu, J., Margineantu, D.D., Osipychev, D., Seshia, S.A.: Formal analysis and redesign of a neural network-based aircraft taxiing system with VerifAI. In: Lahiri, S.K., Wang, C. (eds.) CAV 2020. LNCS, vol. 12224, pp. 122–134. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53288-8_6
10. Gyori, B.M., Liu, B., Paul, S., Ramanathan, R., Thiagarajan, P.S.: Approximate probabilistic verification of hybrid systems. arXiv preprint arXiv:1412.6953 (2014)
11. Junges, S., Torfah, H., Seshia, S.A.: Runtime monitors for Markov decision processes. In: Silva, A., Leino, K.R.M. (eds.) CAV 2021. LNCS, vol. 12760, pp. 553–576. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-81688-9_26
12. Pellizzoni, R., Meredith, P., Caccamo, M., Rosu, G.: Hardware runtime monitoring for dependable cots-based real-time embedded systems. In: 2008 Real-Time Systems Symposium, pp. 481–491 (2008).https://doi.org/10.1109/RTSS.2008.43
13. Sen, K., Viswanathan, M., Agha, G.: Statistical model checking of black-box probabilistic systems. In: Alur, R., Peled, D.A. (eds.) CAV 2004. LNCS, vol. 3114, pp. 202–215. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27813-9_16

14. Seshia, S.A., Sadigh, D., Sastry, S.S.: Toward verified artificial intelligence. Commun. ACM **65**(7), 46–55 (2022)
15. Sistla, A.P., Srinivas, A.R.: Monitoring temporal properties of stochastic systems. In: Logozzo, F., Peled, D.A., Zuck, L.D. (eds.) VMCAI 2008. LNCS, vol. 4905, pp. 294–308. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78163-9_25
16. Stoller, S.D., et al.: Runtime verification with state estimation. In: Khurshid, S., Sen, K. (eds.) RV 2011. LNCS, vol. 7186, pp. 193–207. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-29860-8_15
17. Torfah, H., Xie, C., Junges, S., Vazquez-Chanlatte, M., Seshia, S.A.: Learning monitorable operational design domains for assured autonomy. In: Bouajjani, A., Holík, L., Wu, Z. (eds.) ATVA 2022. LNCS, vol. 13505, pp. 3–22. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19992-9_1
18. Wilcox, C.M., Williams, B.C.: Runtime verification of stochastic, faulty systems. In: Barringer, H., et al. (eds.) RV 2010. LNCS, vol. 6418, pp. 452–459. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16612-9_34
19. Yoon, H., Sankaranarayanan, S.: Predictive runtime monitoring for mobile robots using logic-based Bayesian intent inference. In: ICRA, pp. 8565–8571. IEEE (2021)
20. Zhang, P., Aurandt, A., Dureja, R., Jones, P.H., Rozier, K.Y.: Model predictive runtime verification for cyber-physical systems with real-time deadlines. In: Petrucci, L., Sproston, J. (eds.) FORMATS 2023, pp. 158–180. Springer, Cham (2023). https://doi.org/10.1007/978-3-031-42626-1_10
21. Zhao, Y., Hoxha, B., Fainekos, G., Deshmukh, J.V., Lindemann, L.: Robust conformal prediction for STL runtime verification under distribution shift. In: ICCPS 2024, pp. 169–179. IEEE (2024). https://doi.org/10.1109/ICCPS61052.2024.00022