Conformal Quantitative Predictive Monitoring and Conditional Validity

Francesca Cairoli[1], Tom Kuipers[2], Luca Bortolussi[1], and Nicola Paoletti[2]

[1] University of Trieste
[2] King's College London

Abstract. Predictive monitoring (PM) aims at predicting at runtime the satisfaction of a desired property from the current state of the system under analysis. PM methods need to be efficient, to enable timely interventions against predicted violations, and reliable, given the safety-criticality of the problem. *Quantitative predictive monitoring (QPM)* focuses on stochastic processes, supports rich specifications expressed in Signal Temporal Logic (STL) and provides a quantitative measure of satisfaction by predicting the quantitative (a.k.a. robust) STL semantics, either spatial or temporal. Our solution to QPM integrates machine learning and conformal inference to derive prediction intervals that are highly efficient to compute and with probabilistic guarantees of covering the STL robustness values relative to the stochastic evolution of the system. Conformal guarantees are in general only marginal. However, conditional guarantees can significantly enhance the consistency and reliability of the resulting monitor. To this end, we equip QPM with conformal techniques to ensure conditional validity of the prediction intervals, i.e., such that the probabilistic guarantees hold given some conditions of the system, e.g. the current state or the dynamical mode of the system.

Keywords: Predictive Monitoring; Stochastic Process; Conformal Inference; Conditional Validity.

1 Introduction

Consider a system operating in a potentially dangerous environment and assume there is a formal way to define what poses a danger to the system. The overarching aim of predictive monitoring (PM) [5,6,7] is to predict at runtime, meaning as the system evolves, if a safety violation will take place in the nearby future. Unlike traditional monitoring [3], PM has the potential to detect failures before they occur, thereby enabling preemptive countermeasures, such as switching to a fail-safe mode [22]. Moreover, if the system follows stochastic dynamics, we need to retrieve a measure of risk rather than a yes/no answer. To enable effective deployment at runtime, PM methods need to be efficient and respond quickly so that any system failure can be prevented in time.

Given a model of the system, performing model-checking at run-time would provide a precise solution to the PM problem (precise up to the accuracy of the system's model), but such a solution is computationally expensive in general unless the model is trivial or fully deterministic. In particular, when the system is stochastic, a statistical model checking solution [30] would require simulating at runtime a typically large number of Monte-Carlo trajectories to achieve desired error levels. For this reason, a number of approximate PM techniques based on machine learning have been recently proposed (see e.g. [23,29,11,18,4]), including the so-called Neural Predictive Monitoring (NPM) method [5,6,7,8] and the Quantitative Predictive Monitoring [9,16]. In NPM in particular, the predictive monitor is a neural network classifier trained using data generated through a model checker to predict for any system state whether or not the state satisfies some reachability property. To improve the reliability of the reachability predictions, NPM relies on conformal prediction [2,28] to produce prediction regions with guaranteed coverage and derive uncertainty measures used to infer whether a particular prediction can be trusted. On a similar note, QPM focuses on the predictive monitoring problem for stochastic processes and Signal Temporal Logic (STL) [12] specifications. By predicting STL robustness values (both in its spatial and temporal formulations), QPM provides key quantitative information on the degree of property satisfaction which can be meaningfully used to determine the extent of any corrective actions and enable efficient online model predictive control for STL [24]. For instance, depending on the specification ϕ, if QPM predicts a high value of R_ϕ, then little or no intervention might be needed, while a low R_ϕ value might require a more substantial or quicker intervention to steer the system back to safety. QPM is inspired by NPM but it addresses two significant limitations of the latter, which support only Boolean reachability specifications (as opposed to the full spectrum of STL properties and their quantitative interpretation) and cannot adequately deal with stochastic dynamics. Moreover, QPM monitors both the STL space robustness and the STL time robustness [12,25], which quantifies how much a signal can be perturbed in time (as opposed to space) before affecting its Boolean satisfaction value. This allows us to monitor the imminence of failures, thus providing crucial insights into the available time for an intervention.

A major challenge when dealing with stochastic processes is that every state induces a distribution of robustness values, relative to the future stochastic evolution of the system from the current state of the system. In general, such distribution is analytically intractable and an accurate empirical estimate of such conditional distribution can be very expensive to obtain, potentially requiring a high number of Monte-Carlo simulations. Our approach overcomes this computational bottleneck by deriving monitors able to directly predict some relevant quantiles of the conditional STL robustness distribution. Such quantiles have a two-fold purpose. First, they provide a *measure of risk* [19,17]: for instance, if the 10%-quantile of R_ϕ is zero, then "only" 10% of the system's future trajectories will violate ϕ (i.e., lead to a negative R_ϕ), which, depending on the application, can be interpreted as a low-risk scenario. Second, and most crucially, it allows us to derive *prediction intervals* that cover a certain mass of probability. For small α, if the $(\alpha/2, 1-\alpha/2)$-quantile interval for R_ϕ is entirely above (below) zero, then we can be fairly confident that the system will evolve into satisfying (violating) ϕ with high probability. On the other hand, intervals straddling the zero denote states whose future trajectories may or may not violate the property, which can be thus regarded as uncertain. To predict the quantiles reliably, our approach builds on *Conformalized Quantile Regression (CQR)* [26], a recent conformal inference technique that produces statistically valid prediction intervals on top of quantile regression models. Crucially, this method provides us with marginal probabilistic guarantees, in that, for an arbitrary significance level α, the resulting interval for R_ϕ is guaranteed to cover the true STL robustness value with probability at least $1 - \alpha$. With marginal guarantees, we mean that the coverage probability is satisfied on average w.r.t. the state distribution of the stochastic system. In this paper, we go beyond the marginal case and study techniques to obtain prediction intervals with guaranteed conditional coverage [13,14], meaning that guarantees are met by conditional distributions and not only by the marginal one. In predictive monitoring applications, the importance of conditional guarantees cannot be overstated, as they significantly improve the monitor's accuracy, reliability, and consistency. These stricter guarantees are particularly important in contexts where predictions drive critical decision-making, risk management, and process optimization. Relying solely on marginal guarantees could lead to catastrophic errors for a small proportion of system states – errors that instead a conditionally valid monitor can prevent. Furthermore, it is common for real-world datasets to display imbalances, where marginal guarantees often fail to adequately cover less prevalent groups. In particular, we introduce *input-conditional guarantees* using a conformalized localization technique [13,14] that reweights calibration samples based on a measure of proximity with the considered test input, so that every test state is associated with a prediction interval with guaranteed quasi-input-conditional coverage. We also extend the marginal guarantees of [9] to better address scenarios with unbalanced datasets, where, e.g., trajectories satisfying the specification (a.k.a. positive trajectories) outnumber those violating it (negative trajectories). In particular, we provide conformalized prediction intervals with *sign-conditional coverage guarantees*, meaning that there is an equal probability of positive samples being covered as there is for negative ones. Since failures are rare to observe, this extension enhances the reliability of our QPM in detecting safety hazards. This issue is particularly pronounced in predictive monitoring, where failures may be rare. Implementing class-balanced guarantees effectively addresses this challenge, ensuring the same coverage levels regardless of the satisfaction value. We can also combine the two methods above to obtain prediction intervals that offer both sign- and input-conditional guarantees.

2 Problem Statement

We illustrate the predictive monitoring problem we target with QPM, after introducing background on stochastic processes and Signal Temporal Logic.

Running example. Consider a point moving at a constant velocity on a two-dimensional plane (see Fig. 1). Given the system's current state s, a controller regulates the yawn angle to avoid obstacles (D_1 and D_2 in our example). The avoid property can be easily expressed as an STL formula: $\phi := G\,(\ (d(s, o_1) > r_1) \wedge (d(s, o_2) > r_2)\ )$, where o_i, r_i denote respectively the centre and the radius of obstacle $i \in \{1, 2\}$. Fig. 1 (top) shows the deterministic evolution for three randomly chosen initial states and an intuition of the concept of spatial STL robustness, i.e. how much we can perturb a trajectory with additive noise before changing its truth value. Fig. 1 (bottom) shows the evolution of stochastic dynamics for three randomly initial states. The dashed lines denote the upper and lower quantiles of the distribution over the trajectory space.

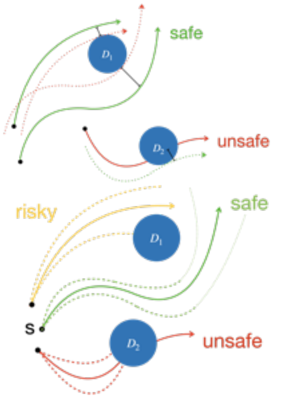

Fig. 1: Example of predictive monitoring of a safety property for deterministic (top) and stochastic dynamics (bottom).

2.1 Stochastic Processes

The systems we consider can be modeled as stochastic processes. A stochastic process is defined as a collection of random variables indexed by some index set T. These random variables are defined on a common probability space $(\Omega, \mathcal{F}, \mathbb{P})$, where Ω is the sample space, $\mathcal{F}$ is the σ-algebra and $\mathbb{P}$ is the probability measure. We can denote the stochastic process as $\{\mathbf{S}(t, \omega), t \in T\}$. A random variable $\mathbf{S}(t, \omega)$ in the collection is thus a function of two variables $t \in T$ and $\omega \in \Omega$. In our application, the index set is countable and represents *discrete time* $T = \{0, 1, \ldots\}$. Each random variable in the collection takes values in a space $S \subseteq \mathbb{R}^n$, the *state space* of dimension n, that should be measurable. A discrete-time step makes the stochastic process move from index i to index $i + 1$. Given a stochastic process $\{\mathbf{S}(t, \omega) : t \in T\}$, then for any point $\omega \in \Omega$, the mapping $\mathbf{S}(\cdot, \omega) : T \to S$, is called a *realization*, or a *sample trajectory* of the stochastic process $\{\mathbf{S}(t, \omega) : t \in T\}$. We assume that the dynamics of the system are Markovian. This assumption is not strict as most systems of interest — Markov chains, stochastic hybrid systems (without non-determinism), and stochastic difference equations — are Markovian or can be made so by augmenting the state space.

2.2 Signal Temporal Logic

System requirements can be expressed via Signal Temporal Logic (STL) [20,12], which enables the specification of properties of dense-time, real-valued signals, and the automatic generation of monitors for testing properties on individual trajectories. The rationale of STL is to transform real-valued signals into Boolean ones, using formulae built on the following *STL syntax*:

$$\phi := true \mid \mu \mid \neg\phi \mid \phi \wedge \phi \mid \phi\ U_I\ \phi,$$

where $I \subseteq \mathbb{T}$ is a temporal interval, either bounded, $I = [a, b]$, or unbounded, $I = [a, +\infty)$, for any $0 \leq a < b$. Atomic propositions μ are (non-linear) inequalities on the states of a signal $\boldsymbol{s}$ at a time t, $\mu = (g(\boldsymbol{s}(t)) > 0)$, where $g : S \to \mathbb{R}$ and $\boldsymbol{s}(t)$ is a state in S. From this essential syntax, it is easy to define other operators, used to abbreviate the syntax in an STL formula: *false* $:= \neg$*true*, $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$, $F_I :=$ *true* $U_I \phi$ and $G_I := \neg F_I \neg\phi$. Monitoring the satisfaction of a formula is done recursively by leveraging the tree structure of the STL formula. The satisfaction relation is defined as follows.

$$\begin{aligned}
(\boldsymbol{s}, t) \models \mu \quad &\Leftrightarrow \quad g(\boldsymbol{s}(t)) > 0 \\
(\boldsymbol{s}, t) \models \neg\phi \quad &\Leftrightarrow \quad \neg((\boldsymbol{s}, t) \models \phi) \\
(\boldsymbol{s}, t) \models \phi_1 \wedge \phi_2 \quad &\Leftrightarrow \quad (\boldsymbol{s}, t) \models \phi_1 \wedge (\boldsymbol{s}, t) \models \phi_2 \\
(\boldsymbol{s}, t) \models \phi_1 U_{a,b} \phi_2 \quad &\Leftrightarrow \quad \exists\, t' \in [t + a, t + b] \text{ s.t. } (\boldsymbol{s}, t') \models \phi_2 \wedge \\
&\qquad\qquad \forall\, t'' \in [t, t'],\ (\boldsymbol{s}, t'') \models \phi_1
\end{aligned}$$

Given a formula ϕ and a signal $\boldsymbol{s}$ over a bounded time interval, we can define the Boolean satisfaction signal as $\chi^\phi(\boldsymbol{s}, t) = 1$ if $(\boldsymbol{s}, t) \models \phi$ and $\chi^\phi(\boldsymbol{s}, t) = 0$ otherwise.

Quantitative semantics. The robustness of a trajectory quantifies the level of satisfaction w.r.t. ϕ. Positive robustness means that the property is satisfied, whereas negative robustness means that the property is violated. Robustness is denoted as a function $R_\phi : S^H \times T \to \mathbb{R}$ that maps a given signal $\boldsymbol{s}$ of length H, a formula ϕ and a time t to some real value, $R_\phi(\boldsymbol{s}, t) \in \mathbb{R}$. It measures the maximum perturbation that can be applied to the signal without changing its truth value w.r.t. ϕ. In particular, we distinguish between *space robustness*, which deals with perturbations in the space dimension, and *time robustness*, which deals with perturbations in time.

Similarly to the Boolean semantics, the quantitative semantics of a formula ϕ over a signal $\boldsymbol{s}$ is defined recursively over the tree structure of the STL formula.

Space robustness is defined as a function $C_\phi : S^H \times T \to \mathbb{R}$ such that:

$$\begin{aligned}
C_\mu(\boldsymbol{s}, t) \quad &= g(\boldsymbol{s}(t)) \\
C_{\neg\phi}(\boldsymbol{s}, t) \quad &= -\, C_\phi(\boldsymbol{s}, t) \\
C_{\phi_1 \wedge \phi_2}(\boldsymbol{s}, t) \quad &= \min(C_{\phi_1}(\boldsymbol{s}, t), C_{\phi_2}(\boldsymbol{s}, t)) \\
C_{\phi_1 U_{[a,b]} \phi_2}(\boldsymbol{s}, t) \quad &= \sup_{t' \in [t+a, t+b]} \Big(\min \big(C_{\phi_2}(\boldsymbol{s}, t'), \inf_{t'' \in [t, t']} C_{\phi_1}(\boldsymbol{s}, t'') \big) \Big).
\end{aligned}$$

Similarly, *time robustness* captures the effect on the satisfaction of shifting the signal in time. The (right) time robustness of an STL formula ϕ with respect to a trace $\boldsymbol{s}$ at time t is defined by:

$$Q^+_\phi(\boldsymbol{s}, t) = \chi^\phi(\boldsymbol{s}, t) \cdot \max\{d \geq 0 \ s.t.\ \forall t' \in [t, t + d],\ \chi^\phi(\boldsymbol{s}, t') = \chi^\phi(\boldsymbol{s}, t)\}.$$

While space robustness is most common, our QPM approach can support any other kind of STL quantitative semantics, e.g., based on a combined space-time robustness [12] or resiliency [10]. Hereafter, we represent a generic STL monitor, encompassing either spatial or temporal quantitative satisfaction, as $R_\phi \in \{C_\phi, Q_\phi^+\}$[3]. The sign of R_ϕ is the Boolean satisfaction of signal $\boldsymbol{s}$ [12,25]: $R_\phi(\boldsymbol{s}, t) > 0 \Rightarrow (\boldsymbol{s}, t) \models \phi$ and $R_\phi(\boldsymbol{s}, t) < 0 \Rightarrow (\boldsymbol{s}, t) \not\models \phi$.

2.3 Quantitative Predictive Monitoring

Given a stochastic process $\mathbf{S} = \{\mathbf{S}(t, \omega), t \in T\}$, an STL requirement ϕ and a state $s_0 \in S$, the robustness over future evolutions of the system starting from s_0 is stochastically distributed according to the conditional distribution

$$\mathbb{P}\left(R_\phi(\boldsymbol{s}, 0) \mid \boldsymbol{s}(0) = s_0\right),$$

where $\boldsymbol{s}$ is a random signal given by the sequence of random variables $\mathbf{S} := (\mathbf{S}(0, \cdot), \mathbf{S}(1, \cdot), \ldots, \mathbf{S}(H-1, \cdot))$. For time k, $\boldsymbol{s}(k) = \mathbf{S}(k, \cdot)$ denotes the random variable corresponding to the state at time k in $\boldsymbol{s}$. This conditional distribution captures the distribution of the STL robustness values for trajectories of length H starting in s_0.

We now formulate the quantitative predictive monitoring problem: from any state s_* of the stochastic process, we aim to construct a prediction interval guaranteed to include, with arbitrary probability, the true STL robustness of any (unknown) stochastic trajectory starting at s. A formal statement of the problem is given below.

Problem 1 (Quantitative Predictive Monitoring). Given a discrete-time stochastic process $\mathbf{S} = \{\mathbf{S}(t, \omega), t \in T\}$ over a state space S, temporal horizon H, a significance level $\alpha \in [0, 1]$ and an STL formula ϕ, derive a monitoring function I producing regions for i.i.d. test trajectories $\boldsymbol{s}$ of the stochastic process $\mathbf{S}$ that satisfy

$$\mathbb{P}_{\boldsymbol{s} \sim \mathbf{S}}\Big(R_\phi(\boldsymbol{s}, 0) \in I\big(\boldsymbol{s}(0)\big)\Big) \geq 1 - \alpha. \tag{1}$$

We will solve Problem 1 as a conditional quantile regression problem. This boils down to learning, for a generic state $\boldsymbol{s}(0)$, an upper and a lower quantile of the random variable $R_\phi(\boldsymbol{s}, 0)$ induced by $\boldsymbol{s}$. We then use these two quantiles to build the output of function I in $\boldsymbol{s}(0)$. To ensure that such an interval is well-calibrated — meaning that the probabilistic guarantees are satisfied theoretically and empirically — we resort to the framework of conformal prediction. These machine-learning techniques are introduced in the next section.

Mode-conditional guarantees. The methods presented in [9] and [16] provide two learning-based approaches to the QPM problem. In particular, [9] provides a prediction interval with guaranteed coverage of the STL robustness values.

[3] Time-robustness assumes discrete values, here we define R_ϕ as a real-valued function considering that $\mathbb{Z} \subset \mathbb{R}$.

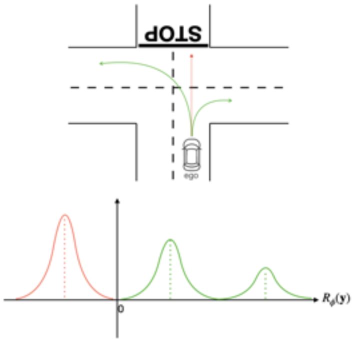

Fig. 2: Bi-modal scenario: dynamics (top), robustness (bottom).

The second returns an interval centered around the STL robustness of the expected dynamics in expectation. Both methods work well in general but they share the same limitation, i.e., they fail to provide useful results in multi-modal scenarios (as the ones shown in Fig.2). In a scenario like the one in Fig.2, the former would result in an extremely large interval covering all the robustness values. The latter would reconstruct the robustness of the averaged signal, whose values are much different compared to the one observed in practice. To this end, assume that robustness can be divided into C modes and that there is a function $\pi : \mathbb{R} \to \{1, \ldots, C\}$ mapping the robustness of every trajectory to its mode. We may be interested in obtaining coverage guarantees over every mode, i.e., we may require that the prediction intervals are $1-\alpha$ accurate both when observations are in each mode. For instance, the sign of robustness could identify two modes. Mathematically, mode-conditional guarantees are formulated as follows: for every mode $j \in \{1, \ldots, C\}$

$$\mathbb{P}_{\boldsymbol{s} \sim \mathbf{S}}\Big(R_\phi(\boldsymbol{s}, 0) \in I\big(\boldsymbol{s}(0)\big) \;\big|\; \pi\big(R_\phi(\boldsymbol{s})\big) = j)\Big) \geq 1 - \alpha. \tag{2}$$

Input-conditional guarantees. Both (1) and (2) provide a notion of marginal coverage, where coverage guarantees hold on average over the distribution of the stochastic process trajectories. Below we formulate input-conditional guarantees, which are stronger as they instead require the same level of coverage for every initial state $s_* \sim \mathbf{S}(0, \cdot)$:

$$\mathbb{P}_{\boldsymbol{s} \sim \mathbf{S}}\Big(R_\phi(\boldsymbol{s}, 0) \in I\big(\boldsymbol{s}(0)\big) \;\big|\; \boldsymbol{s}(0) = s_*\Big) \geq 1 - \alpha, \text{ for all } s_* \in S. \tag{3}$$

3 Background on Conformal Prediction

Consider a generic supervised learning setting where X denotes the input space, Y the target space, and $Z = X \times Y$. Let $\mathcal{Z}$ be the data-generating distribution, i.e., the distribution of the points $(x, y) \in Z$. We assume that the target y of a point $(x, y) \in Z$ is the result of the application of a function $f^* : X \to Y$, typically unknown or very expensive to evaluate. The goal of a supervised learning algorithm is to find a function $f : X \to Y$ that, from a finite set of observations, learns to behave as similarly as possible to f^* over the entire input space. For an input $x \in X$, we denote with y the true target value of x and with $\hat{y}$ the prediction by f, i.e. $\hat{y} = f(x)$. Test inputs, whose unknown true target values we aim to predict, are denoted by x_*. For the sake of clarity, we start by showing conformal prediction approaches for deterministic predictors and then move to present Conformalized Quantile Regression (CQR) [26], an approach to handle the stochastic case.

Conformal prediction (CP) associates measures of reliability with any traditional supervised learning problem, either regression or classification [2,28]. CP enriches point-wise predictions with *prediction regions with guaranteed marginal validity.*

Definition 1 (Prediction region). *For significance level $\alpha \in (0,1)$ and test input x_*, the α-prediction region for x_*, $\Gamma^\alpha(x_*) \subseteq Y$, is a set of target values s.t.*

$$\mathbb{P}_{(x_*,y_*)\sim\mathcal{Z}}(y_* \in \Gamma^\alpha(x_*)) \geq 1-\alpha. \tag{4}$$

The idea of CP is to construct the prediction region by "inverting" a suitable hypothesis test: given a test point x_* and a tentative target value y', we *exclude* y' from the prediction region only if it is unlikely that y' is the true value for x_*. The test statistic is given by a so-called *nonconformity function (NCF)* $\Delta : Z \to \mathbb{R}$, which, given a predictor f and a point $z = (x, y)$, measures the deviation between the true value y and the corresponding prediction $f(x)$. In this sense, Δ can be viewed as a generalized residual function. In other words, CP builds the prediction region $\Gamma^\alpha(x_*)$ for a test point x_* by excluding all targets y' whose NCF values are unlikely to follow the NCF distribution of the true targets:

$$\Gamma^\alpha(x_*) = \{y \in Y : \Delta(x_*, y) \leq Q(1-\alpha; \mathcal{F})\}\,, \tag{5}$$

where $\mathcal{F} := \frac{1}{|Z_c|+1}\big(\sum_{z_i\in Z_c}\delta_{\beta_i} + \delta_\infty\big)$ is the calibration distribution, with δ_{β_i} being the Dirac distribution centered at $\beta_i = \Delta(x_i, y_i)$ and Q is the quantile function. This prediction region is guaranteed to contain the true (unknown) value y_* with confidence $1-\alpha$. The probability term in Eq. (5) is often called the p-value. From a practical viewpoint, the NCF distribution $Pr_{(x,y)\sim\mathcal{Z}}(\Delta(x, y))$ cannot be derived in an analytical form, and thus we use an empirical approximation derived using a sample Z_c of $\mathcal{Z}$. This approach is called *inductive CP* [21] and Z_c is referred to as *calibration set*. CP's theoretical guarantees hold under the *exchangeability* assumption (a "relaxed" version of i.i.d.) by which the joint probability of calibration and test points is invariant to permutations.

Validity and Efficiency. CP performance is measured via two quantities: 1) *validity* (or *coverage*), i.e. the empirical error rate observed on a test sample, which should be as close as possible to the significance level α, and 2) *efficiency*, i.e. the size of the prediction regions, which should be small in order to have informative (i.e., non-trivial) regions. CP-based prediction regions are automatically valid, whereas the efficiency depends on the chosen nonconformity function and on the accuracy of the underlying model.

3.1 Conformal Prediction for Regression

In regression problems, we have a continuous target space $Y \subseteq \mathbb{R}^n$. The inductive CP algorithm is divided into an offline phase, executed only once, and an online phase, executed for every test point x_*. In the offline phase (steps 1–3 below), we train the classifier f and construct the calibration distribution, i.e., the empirical approximation of the NCF distribution. In the online phase (steps 4–5), we derive the prediction region for x_* using the computed regressor and distribution.

1. Draw sample Z' of $\mathcal{Z}$. Split Z' into training Z_t and calibration set Z_c.
2. Train regressor f using Z_t. Use f to define an NCF Δ.
3. Construct the empirical distribution $\mathcal{F} = \frac{1}{|Z_c|+1}\big(\sum_{z_i \in Z_c} \delta_{\beta_i} + \delta_\infty\big)$ of calibration scores by computing, for each $z_i \in Z_c$, the NCF score $\beta_i = \Delta(z_i)$.
4. Identify the critical value $\beta_{(\alpha)} = Q(1-\alpha; \mathcal{F})$ of the calibration distribution, i.e. its empirical $(1-\alpha)$-quantile, or the $\lfloor \alpha \cdot (|Z_c|+1) \rfloor$-th largest calibration score.
5. Return the prediction region

$$\Gamma^{\alpha}(x_*) = f(x_*) \pm \beta_{(\alpha)}. \tag{6}$$

Notice that such prediction intervals have the same width ($\beta_{(\alpha)}$) for all inputs. A natural NCF in regression is the norm of the difference between the real and the predicted target value, i.e., $\Delta(x) = \|y - f(x)\|$.

Predictive uncertainty. A CP-based prediction region provides a set of plausible predictions with statistical guarantees, and as such, also captures the uncertainty about the prediction. The size of the prediction region is determined by the chosen significance level α. Specifically, from Eq. (6) we can see that, for levels $\alpha_1 \geq \alpha_2$, the corresponding prediction regions are such that $\Gamma^{\alpha_1} \subseteq \Gamma^{\alpha_2}$, as a smaller α yields a larger critical value $\beta_{(\alpha)}$.

3.2 Conformalized Quantile Regression

Let us now consider the stochastic setting with a probabilistic function mapping an input $x \in X$ into a distribution over the target space Y.

Quantile Regression. The aim of conditional Quantile Regression (QR) is to estimate a given quantile of such a distribution over Y conditional on an input $x \in X$. Let $F(y'|x = x') := \mathbb{P}_{(x,y)\sim\mathcal{Z}}(y \leq y'|x = x')$ be the conditional distribution function of y given x. Then, the α-th conditional quantile function is defined as

$$q_\alpha := \inf\{y' \in \mathbb{R} \mid F(y'|x = x') \geq \alpha\}. \tag{7}$$

Given a significance level α, we consider lower and upper quantiles w.r.t. $\alpha_{lo} = \alpha/2$ and $\alpha_{hi} = 1 - \alpha/2$, respectively. We define the desired prediction interval as $PI(x) := [q_{\alpha_{lo}}(x), q_{\alpha_{hi}}(x)]$. By construction, this interval satisfies

$$\mathbb{P}_{(x,y)\sim\mathcal{Z}}\big(y \in PI(x)\big) \geq 1 - \alpha. \tag{8}$$

Since the prediction interval is conditional on the input, the length of the interval is not fixed in general and changes at different values of x. QR infers such prediction interval from the data. In particular, estimating the quantiles can be expressed as approximating the quantile function and thus it can be framed as an optimization problem. In a nutshell, the idea is to propose a parametric function $f(x;\theta)$ as a candidate approximator for $q_\alpha(x)$ and then optimize over its parameters θ so that it closely resembles the quantile function. In this work, we choose $f(\cdot,\theta)$ to be a deep neural network optimized w.r.t. the pinball loss function $\mathcal{L}_\alpha(y, f(x,\theta)) = \alpha \max(y - f(x,\theta), 0) + (1-\alpha)\max(f(x,\theta) - y, 0)$. In general, each quantile requires the training of a different neural network. However, one could also train a single multi-output NN that learns to approximate multiple quantile functions at the same time. The multi-output objective function is obtained by averaging over the respective losses.

Conformalized Quantile Regression. The goal of Conformalized Quantile Regression (CQR) is to adjust the QR prediction interval so that it is guaranteed to contain the $(1-\alpha)$ mass of probability, i.e. to satisfy (8). As for CP, we divide the dataset Z' in a training set Z_t and a calibration set Z_c. We train the QR $f(\cdot;\theta)$ over Z_t and on Z_c we compute the nonconformity scores as

$$E_i := \max\{\hat{q}_{\alpha_{lo}}(x_i) - y_i, y_i - \hat{q}_{\alpha_{hi}}(x_i) \mid (x_i, y_i) \in Z_c\}. \tag{9}$$

In our notation, $\hat{q}_{\alpha_{lo}}(x)$ and $\hat{q}_{\alpha_{hi}}(x)$ denote the two outputs of $f(x;\hat{\theta})$. The conformalized prediction interval is thus defined as

$$CPI(x_*) = [\hat{q}_{\alpha_{lo}}(x_*) - \tau, \hat{q}_{\alpha_{hi}}(x_*) + \tau],$$

where τ is the $(1-\alpha)$-quantile of the distribution $\mathcal{F} := \frac{1}{|Z_c|+1}\left(\sum_{z_i \in Z_c} \delta_{E_i} + \delta_\infty\right)$ of $\{E_i : z_i \in Z_c\} \cup \{\infty\}$, i.e., $\tau = Q(1-\alpha; \mathcal{F})$.

In the following, we will abbreviate with PI a (non-calibrated) QR prediction interval and with CPI a (calibrated) conformalized prediction interval.

Remark 1. This nonconformity function, and thus τ, can be negative and thus the conformalized prediction interval can be tighter than the original prediction interval. This means that the CPI can be more efficient than the PI, where the efficiency is the average width of the prediction intervals over a test set. The CPI has guaranteed coverage (the PI does not), meaning $\mathbb{P}_{(x,y)\sim\mathcal{Z}}(y \in CPI(x)) \geq 1 - \alpha$.

Remark 2. In principle, one could use traditional CP for regression (see Section 3.1) to obtain valid prediction intervals for the stochastic case. The main advantage of CQR is that it produces CPIs that are adaptive to heteroscedasticity, i.e., they account for the fact that the variability in the output may be affected by the value of the input. On the contrary, intervals produced by CP for regression have fixed sizes and hence, do not account for heteroscedasticity. Moreover, the PI of CP for regression would be relative to the conditional mean of the STL robustness, instead of the conditional quantile range that we are interested in.

Class-conditional guarantees. One might require prediction intervals that maintain similar error rates across various subsets of the data [1]. Suppose for instance that observations (x, y) can be categorized in a discrete and finite set of classes $\{1, \ldots, G\}$. We could ask for *class-balanced* guarantees of coverage:

$$\mathbb{P}_{(x,y)\sim\mathcal{Z}}(y \in CPI(x) \mid (x, y) \in g) \geq 1 - \alpha \tag{10}$$

for every class $g \in \{1, \ldots, G\}$. This solution typically relies on splitting the calibration set Z_c^ϕ in G sub-calibration set and using only the group-specific calibration scores to recalibrate the prediction interval. Within the QPM framework, classes are determined by the Boolean satisfaction value, namely, the sign of R_ϕ. As a result, we require guarantees that are conditional on the sign. However, one could propose a different partition of the robustness space and provide guarantees w.r.t. such partition.

Input-conditional coverage. We may require prediction intervals that capture the uncertainty of a specific input point. However, the guarantees obtained so far are averaged over the data distribution, and the critical value τ used to "conformalize" the predictions is constant and does not adapt to the predictive uncertainty of a single data point. This kind of adaptivity is typically formalized by the following conditional coverage [27] requirement:

$$\mathbb{P}_{(x,y)\sim\mathcal{Z}}(y \in CPI(x) \mid x = x_*) \geq 1 - \alpha, \text{ for all } x_* \in X. \tag{11}$$

That is, for every value of the input x, we seek to return a prediction interval with $1-\alpha$ coverage over the outputs of x. However, distribution-free conditional coverage is impossible to achieve with a finite sample [27,15], and hence, we adopt a localization-based strategy to achieve approximate conditional validity, as described below.

Quasi-input-conditional coverage. In conformal inference, each calibration point contributes equally to the critical value τ, without considering its proximity to the test input x_*. To achieve approximate conditional coverage, however, it is essential to give more importance to points that are close to x_* and less importance to those further away. This intuition is at the core of the so-called *localized CP* approach [13,14].

A *localizer* is a function $L : X \times X \to [0, 1]$ such that, for any pair of inputs $x_i, x_j \in X$, $L(x_i, x_j)$ grows as the distance between x_i and x_j shrinks. For example, one could define $L(x_i, x_j) = \exp(-\|x_i - x_j\|)$. In general, L must be defined so that, for every $x_i \in X$, $L(x_i, x_i) = 1$. We denote with $L_* = L(x_*, \cdot) : X \to [0, 1]$ the localizer relative to test point x_*. The localizer is used to reweight the calibration distribution so that calibration points close to x_* have a higher probability, resulting in the distribution $\mathcal{F}_* := \frac{1}{|Z_c|+1}\big(\sum_{z_i \in Z_c} w_*^i \cdot \delta_{E_i} + w_* \cdot \delta_\infty\big)$, where $w_*^i := \frac{L_*(x_i)}{L_*(x_*)+\sum_{z_i \in Z_c} L(x_i)}$ and $w_* := \frac{L_*(x_*)}{L_*(x_*)+\sum_{z_i \in Z_c} L(x_i)}$.

However, this reweighting breaks the exchangeability assumption, as permutations of calibration and test points no longer hold equal probability, meaning that it may compromise marginal validity. To address this problem and uphold finite sample coverage guarantees, localized CP provides a criterion to select the error level $\tilde{\alpha}$ used to derive the critical value $\tau^* = Q(1-\tilde{\alpha};\mathcal{F}_*)$ and construct the quasi-conditional prediction region at x_*.

For a calibration point $z_j = (x_j, y_j) \in Z_c$, let $\mathcal{F}_j^* := \frac{1}{|Z_c|+1}\big(\sum_{z_i \in Z_c} w_j^i \cdot \delta_{E_i} + w_j^* \cdot \delta_{\tau^*}\big)$ be distribution of $\{E_i : z_i \in Z_c\} \cup \{\tau^*\}$ reweighted around x_j, and let $\mathcal{F}_j^0 := \frac{1}{|Z_c|+1}\big(\sum_{z_i \in Z_c} w_j^i \cdot \delta_{E_i} + w_j^* \cdot \delta_0\big)$ be the reweighted distribution of $\{E_i : z_i \in Z_c\} \cup \{0\}$. Let $\tau_{j1}^* := Q\big(1-\tilde{\alpha};\mathcal{F}_j^*\big)$, and $\tau_{j2}^* := Q\big(1-\tilde{\alpha};\mathcal{F}_j^0\big)$. The criterion consists of selecting $\tilde{\alpha}$ as the highest value in the range $[0,\alpha]$ such that the following two conditions are met

$$\frac{1}{|Z_c|+1}\left(\sum_{z_i \in Z_c} \mathbb{1}_{\{E_i \leq \tau_{i1}^*\}}\right) \geq 1-\alpha \text{ and } \frac{1}{|Z_c|+1}\left(\sum_{z_i \in Z_c} \mathbb{1}_{\{E_i \leq \tau_{i2}^*\}}\right) \geq 1-\alpha, \tag{12}$$

or $\tau^* = \infty$. The resulting prediction interval

$$CPI^L(x_*) = [\hat{q}_{\alpha_{lo}}(x_*) - \tau^*, \hat{q}_{\alpha_{hi}}(x_*) + \tau^*], \tag{13}$$

can be proven to satisfy marginal validity [13] and provides the following quasi-conditional guarantees

$$\mathbb{P}_{z \sim \mathcal{Z}^{L_*}}(y \in CPI^L(x) \mid x = x_*) \geq 1-\alpha, \text{ for all } x_*,$$

where $\mathcal{Z}^{L_*}$ represents the distribution $\mathcal{Z}$ localized around x_*.

The search for $\tilde{\alpha}$ has to be carried out for each test input, potentially impacting computational efficiency due to its dependence on the calibration set size. Specifically, the matrix of weights w_j^i needs to be computed once for every test point, leading to a computational complexity that scales quadratically with the number of calibration points, $n := |Z_c|$. Additionally, verifying the condition requires, for every candidate $\tilde{\alpha}$, the computation of n quantiles from sorted arrays, which is linear in n. Therefore, the overall complexity for each test point is $O(n^2 + n^2 \cdot k)$ where k is the number of candidate values for $\tilde{\alpha}$. An efficient implementation (see [14]) may reduce the cost to $O(n \cdot \log n)$. However, this process typically does not present a significant computational challenge for moderately sized calibration sets, given that the condition is often satisfied long before all k candidate values of $\tilde{\alpha}$ are explored.

4 Conformal Quantitative Predictive Monitoring

We present a method to solve Problem 1. For a given discrete-time stochastic process $\mathbf{S} = \{\mathbf{S}(t,\omega), t \in T\}$ over state space S and a state $s \sim \mathbf{S}(k,\cdot)$ at time $k \in T$, the stochastic evolution (bounded by horizon H) of the system starting at s can be described by the conditional distribution $\mathbb{P}(\boldsymbol{s} \mid \boldsymbol{s}(k) = s)$, where $\boldsymbol{s} = (\boldsymbol{s}(k), \dots, \boldsymbol{s}(k+H)) \in S^H$ is the random trajectory of length H starting at time t_k, $\boldsymbol{s}(i) = \mathbf{S}(i,\cdot)$ for any $i \in T$.

The quantitative STL semantics, either spatial or temporal, inherits the stochasticity from the dynamics of the system. For an STL property ϕ, we denote

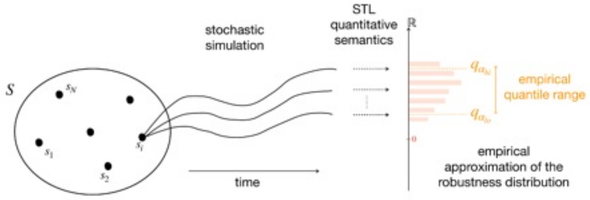

Fig. 3: Diagram illustrating the generation of the dataset.

with $\mathcal{R}_\phi(s,k)$ the random variable denoting the STL robustness value relative to ϕ of trajectories starting from $s \in S$ at time k, i.e.,

$$\mathcal{R}_\phi(s,k) \sim \mathbb{P}(R_\phi(\boldsymbol{s},k) \mid \boldsymbol{s}(k) = s).$$

Our conformal solution to the QPM aims at finding, for each state $s \in S$ of the system, a prediction region that covers a certain probability mass of the STL robustness distribution $\mathcal{R}_\phi(s,k)$, see Problem 1. In simpler terms, we aim at monitoring how safe the system is relative to the unknown and stochastic future evolution from its current state s. In this way, one can intervene preemptively on the system in order to prevent any failures. However, the distribution of $\mathcal{R}_\phi(s,k)$ is impossible to compute exactly and in an efficient manner. Thus, we resort to Conformalized Quantile Regression (CQR), introduced in Section 3.2, to compute a prediction interval that, for each state s, is guaranteed to cover a desired level $(1-\alpha)$ of the probability mass for the distribution of robustness values $\mathcal{R}_\phi(s,k)$. In short, our solution consists of four steps, detailed below: *dataset generation*, *QR training*, *NCF scores computation* and *inference*. Note that only the last step, which is by far the quickest, is performed online, the others are performed offline and hence, do not affect runtime performance.

Dataset generation. In this step, we collect data for training the QR function and constructing the calibration set. To do so, we perform Monte-Carlo simulations of the process in order to obtain an empirical approximation of $\mathcal{R}_\phi(s,0)$. In particular, we randomly sample N states $s_1,\dots,s_N \sim \mathbf{S}(\cdot,\cdot)$. Then, for each state s_i, we simulate M trajectories of length H, $\boldsymbol{s}_i^1,\dots,\boldsymbol{s}_i^M$ where $\boldsymbol{s}_i^j$ is a realization of $\mathbb{P}(\boldsymbol{s} \mid \boldsymbol{s}(k) = s_i)$, and compute the robustness value $R_\phi(\boldsymbol{s}_i^j,0)$ of each of these trajectories. We note that $\{R_\phi(\boldsymbol{s}_i^j,0)\}_{j=1}^M$ is an empirical approximation of $\mathcal{R}_\phi(s_i,0)$. Fig. 3 shows an overview of the steps needed to generate the dataset. The dataset is thus defined as $Z^\phi = \big\{\big(s_i, R_\phi(\boldsymbol{s}_i^j,0)\big); i = 1,\dots,N;\ j = 1,\dots,M\big\}$.

The generation of the test set Z^ϕ_{test} is very similar to that of Z^ϕ. The main difference is in that the number of trajectories that we simulate from each state s is much larger than M, i.e., $M_{test} \gg M$. This allows us to obtain a highly accurate empirical approximation of the distribution of $\mathcal{R}_\phi(s)$, which we use as the ground-truth baseline in our experimental evaluation.

QR training and residuals computation. We divide the dataset Z^ϕ into a training set Z^ϕ_t and a calibration set Z^ϕ_c. We then use Z^ϕ_t to train a QR that learns how to map states s into three quantiles $f(s;\hat{\theta}) = \{\hat{q}_{\alpha_{lo}}(s), \hat{q}_{0.5}(s), \hat{q}_{\alpha_{hi}}(s)\}$, where

$\alpha_{lo} = \alpha/2$ and $\alpha_{hi} = 1 - \alpha/2$. In order to better reconstruct the shape of the target distribution, we also predict the median quantile, $q_{0.5}(s)$. We then apply CQR, i.e. we compute the residuals of the QR over Z_c^ϕ, as in (9), and find the critical residual value τ.

Inference. For a test state s_*, this step involves predicting the relevant quantiles, $\hat{q}_{\alpha_{lo}}(s_*)$ and $\hat{q}_{\alpha_{hi}}(s_*)$, using the QR predictor and correct the resulting interval using the critical residual value τ. The calibrated interval returned at inference time becomes

$$CPI(s_*) := [\hat{q}_{\alpha_{lo}}(s_*) - \tau, \hat{q}_{\alpha_{hi}}(s_*) + \tau] \subseteq \mathbb{R} \tag{14}$$

and provides the marginal coverage guarantees stated in Eq. 1.

4.1 Conditional validity

System failures are rare to observe, and so, the datasets used to train our predictive monitors are often unbalanced in that they contain more examples of satisfying signals (positive samples) than violating signals (negative samples). In this setting, marginal coverage does not guarantee adequate coverage of both classes of signals. Suppose, for instance, that positive samples have frequency $1 - \alpha$, whereas negative samples have frequency α, and prediction intervals always cover positive samples but never cover negative ones. Then these prediction intervals meet the $1 - \alpha$ nominal coverage in a marginal sense but fail to cover negative instances. Sign-conditional coverage would imply that the prediction intervals cover the output at least $1 - \alpha$ of the time in both classes. Similarly, input-conditional coverage is a very strong property that states the probability of the prediction interval needs to be greater than $1 - \alpha$ for each input, i.e., for any subset of the data [1]. Localized conformal predictions offer local coverage guarantees.

1. Sign-conditional coverage. The goal here is to provide calibrated prediction intervals where both positive and negative samples are ensured equal coverage. To achieve this, the calibration set Z_c^ϕ is partitioned into its positive $Z_c^{\phi^{(+)}}$ and negative $Z_c^{\phi^{(-)}}$ parts, i.e., in the subsets containing only calibration points with positive and negative outputs, respectively. The prediction interval $PI(s)$ induces two separate calibration distributions of nonconformity scores from which we extract the two $(1-\alpha)$-quantiles, τ^+ and τ^- respectively. The prediction interval $PI(s)$ undergoes a sign-specific calibration obtaining $\overline{CPI}^+(s) = CPI^+(s) \cap \mathbb{R}^+$,

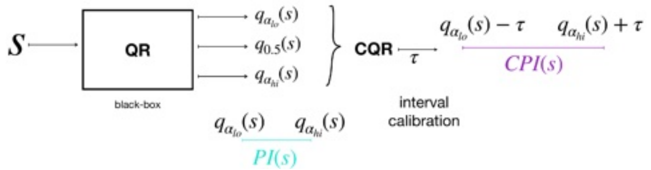

Fig. 4: Overview of the conformal quantitative predictive monitoring technique.

where $CPI^{+}(s)$ is defined as in (14), guaranteed to cover $1-\alpha$ of the positive values of $R_\phi(\boldsymbol{s})$:

$$\mathbb{P}\Big(R_\phi(\boldsymbol{s}) \in \overline{CPI}^{+}\big(\boldsymbol{s}(0)\big)\Big|\ R_\phi(\boldsymbol{s}) \geq 0\Big) \geq 1-\alpha \tag{15}$$

and analogously $\overline{CPI}^{-}(x) = CPI^{-}(x) \cap \mathbb{R}^{-}$ is guaranteed to cover $1-\alpha$ of the negative values of $R_\phi(\boldsymbol{s})$. The union $CPI(s) := \overline{CPI}^{-}(s) \cup \overline{CPI}^{+}(s)$ of the two recalibrated intervals is guaranteed to cover $1-\alpha$ of all the values of $R_\phi(\boldsymbol{s})$ for $\boldsymbol{s}(0) = s$:

$$\mathbb{P}\Big(R_\phi(\boldsymbol{s}) \in CPI\big(\boldsymbol{s}(0)\big)\Big) \geq 1-\alpha. \tag{16}$$

2. Localized Conformal Prediction. We build the localizer function $L_* = L(s_*)$ (introduced in Sect. 3.2, *Quasi-input-conditional coverage*) as a k-nearest-neighbors function, where $L_*(s) = 1$ if s is among the k nearest neighbors of s_* and 0 otherwise. Each sample in the calibration set is weighted according to L_*, resulting in a state-specific calibration of the prediction interval $CPI^{L}(s_*)$ as in (13) resulting in quasi-conditional guarantees

$$\mathbb{P}_{\boldsymbol{s}\sim \boldsymbol{S}^{L_*}}\Big(R_\phi(\boldsymbol{s}) \in CPI^{L}(s_*)\ \Big|\ \boldsymbol{s}(0) = s_*\Big) \geq 1-\alpha, \tag{17}$$

where $\boldsymbol{S}^{L_*}$ denotes the distribution $\boldsymbol{S}$ localized around s_*.

Remark 3. It is straightforward to combine the two approaches and obtain prediction intervals that offer both sign and input conditional guarantees.

Experimental evaluation. Due to space constraints we keep this paper methodological and refer the interested reader to [9] for all the details about the experimental evaluation. Our implementation is available at `https://github.com/francescacairoli/Conformal_QPM.git`.

5 Conclusions

We presented quantitative predictive monitoring (QPM), an efficient and reliable technique to monitor the evolution of a stochastic system at runtime. In particular, given a requirement expressed as an STL formula, QPM quantifies how robustly this requirement is satisfied both in space and in time by means of a range of STL robustness values. This interval undergoes a principled recalibration that guarantees a desired level of coverage, i.e. the interval covers the exact STL robustness values with a given confidence. By carefully designing the recalibration strategy we can achieve either marginal or conditional guarantees of coverage. The proposed technique avoids expensive Monte-Carlo simulations at runtime by leveraging conformalized quantile regression. The resulting method has very little overhead during runtime execution. The conformal approach reaches higher accuracy and it also adjusts the width of the interval w.r.t. desired confidence level, reducing the number of over-conservative predictions.

In future work, we will investigate a possible dynamics-aware approach to inference and how to exploit it in highly multi-modal scenarios. The latter should aim at limiting the inference only to an estimate of the system manifold, i.e. the region of the state space that is likely to be visited by the evolving stochastic process.

References

1. Angelopoulos, A.N., Bates, S.: A gentle introduction to conformal prediction and distribution-free uncertainty quantification. arXiv preprint arXiv:2107.07511 (2021)
2. Balasubramanian, V., Ho, S.S., Vovk, V.: Conformal prediction for reliable machine learning: theory, adaptations and applications. Newnes (2014)
3. Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G., Maler, O., Ničković, D., Sankaranarayanan, S.: Specification-based monitoring of cyber-physical systems: a survey on theory, tools and applications. In: Lectures on Runtime Verification, pp. 135–175. Springer (2018)
4. Bortolussi, L., Cairoli, F., Carbone, G., Pulcini, P.: Scalable stochastic parametric verification with stochastic variational smoothed model checking. In: International Conference on Runtime Verification. pp. 45–65. Springer (2023)
5. Bortolussi, L., Cairoli, F., Paoletti, N., Smolka, S.A., Stoller, S.D.: Neural predictive monitoring. In: International Conference on Runtime Verification. pp. 129–147. Springer (2019)
6. Bortolussi, L., Cairoli, F., Paoletti, N., Smolka, S.A., Stoller, S.D.: Neural predictive monitoring and a comparison of frequentist and bayesian approaches. International Journal on Software Tools for Technology Transfer **23**(4), 615–640 (2021)
7. Cairoli, F., Bortolussi, L., Paoletti, N.: Neural predictive monitoring under partial observability. In: Runtime Verification: 21st International Conference, RV 2021, Virtual Event, October 11–14, 2021, Proceedings 21. pp. 121–141. Springer (2021)
8. Cairoli, F., Bortolussi, L., Paoletti, N.: Learning-based approaches to predictive monitoring with conformal statistical guarantees. In: International Conference on Runtime Verification. pp. 461–487. Springer (2023)
9. Cairoli, F., Paoletti, N., Bortolussi, L.: Conformal quantitative predictive monitoring of stl requirements for stochastic processes. In: Proceedings of the 26th ACM International Conference on Hybrid Systems: Computation and Control. pp. 1–11 (2023)
10. Chen, H., Lin, S., Smolka, S.A., Paoletti, N.: An STL-based formulation of resilience in cyber-physical systems. In: International Conference on Formal Modeling and Analysis of Timed Systems. pp. 117–135. Springer (2022)
11. Chou, Y., Yoon, H., Sankaranarayanan, S.: Predictive runtime monitoring of vehicle models using bayesian estimation and reachability analysis. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 2111–2118 (2020)
12. Donzé, A., Maler, O.: Robust satisfaction of temporal logic over real-valued signals. In: Proceedings of International Conference on Formal Modeling and Analysis of Timed Systems. pp. 92–106. Springer, Klosterneuburg, Austria (Sep 2010). https://doi.org/10.1007/978-3-642-15297-9_9
13. Guan, L.: Conformal prediction with localization. arXiv preprint arXiv:1908.08558 (2019)

14. Guan, L.: Localized conformal prediction: A generalized inference framework for conformal prediction. Biometrika **110**(1), 33–50 (2023)
15. Lei, J., Wasserman, L.: Distribution-free prediction bands for non-parametric regression. Journal of the Royal Statistical Society Series B: Statistical Methodology **76**(1), 71–96 (2014)
16. Lindemann, L., Qin, X., Deshmukh, J.V., Pappas, G.J.: Conformal prediction for stl runtime verification. In: Proceedings of the ACM/IEEE 14th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2023). pp. 142–153 (2023)
17. Lindemann, L., Rodionova, A., Pappas, G.: Temporal robustness of stochastic signals. In: 25th ACM International Conference on Hybrid Systems: Computation and Control. pp. 1–11 (2022)
18. Ma, M., Stankovic, J., Bartocci, E., Feng, L.: Predictive monitoring with logic-calibrated uncertainty for cyber-physical systems. ACM Transactions on Embedded Computing Systems (TECS) **20**(5s), 1–25 (2021)
19. Majumdar, A., Pavone, M.: How should a robot assess risk? towards an axiomatic theory of risk in robotics. In: Robotics Research, pp. 75–84. Springer (2020)
20. Maler, O., Nickovic, D.: Monitoring temporal properties of continuous signals. In: Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems, pp. 152–166. Springer (2004)
21. Papadopoulos, H.: Inductive conformal prediction: Theory and application to neural networks. In: Tools in artificial intelligence, chap. 18, pp. 315–330. InTech (2008)
22. Phan, D.T., Grosu, R., Jansen, N., Paoletti, N., Smolka, S.A., Stoller, S.D.: Neural simplex architecture. In: NASA Formal Methods Symposium. pp. 97–114. Springer (2020)
23. Qin, X., Deshmukh, J.V.: Predictive monitoring for signal temporal logic with probabilistic guarantees. In: Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control. pp. 266–267. ACM (2019)
24. Raman, V., Donzé, A., Maasoumy, M., Murray, R.M., Sangiovanni-Vincentelli, A., Seshia, S.A.: Model predictive control with signal temporal logic specifications. In: Proceedings of the IEEE Conference on Decision and Control. pp. 81–87. IEEE, Los Angeles, USA (Dec 2014). https://doi.org/10.1109/CDC.2014.7039363
25. Rodionova, A., Lindemann, L., Morari, M., Pappas, G.J.: Time-robust control for stl specifications. In: 2021 60th IEEE Conference on Decision and Control (CDC). pp. 572–579. IEEE (2021)
26. Romano, Y., Patterson, E., Candès, E.J.: Conformalized quantile regression. arXiv preprint arXiv:1905.03222 (2019)
27. Vovk, V.: Conditional validity of inductive conformal predictors. In: Asian conference on machine learning. pp. 475–490. PMLR (2012)
28. Vovk, V., Gammerman, A., Shafer, G.: Algorithmic learning in a random world. Springer Science & Business Media (2005)
29. Yel, E., Carpenter, T.J., Di Franco, C., Ivanov, R., Kantaros, Y., Lee, I., Weimer, J., Bezzo, N.: Assured runtime monitoring and planning: Toward verification of neural networks for safe autonomous operations. IEEE Robotics & Automation Magazine **27**(2), 102–116 (2020)
30. Younes, H.L., Kwiatkowska, M., Norman, G., Parker, D.: Numerical vs. statistical probabilistic model checking. International Journal on Software Tools for Technology Transfer **8**(3), 216–228 (2006)