Neural Predictive Monitoring

Nicola Paoletti
Royal Holloway, University of London, UK

JWW: L Bortolussi, F Cairoli (Università di Trieste), SA Smolka, SD Stoller (Stony Brook University)

RV 2019 – Porto, 11 October 2019

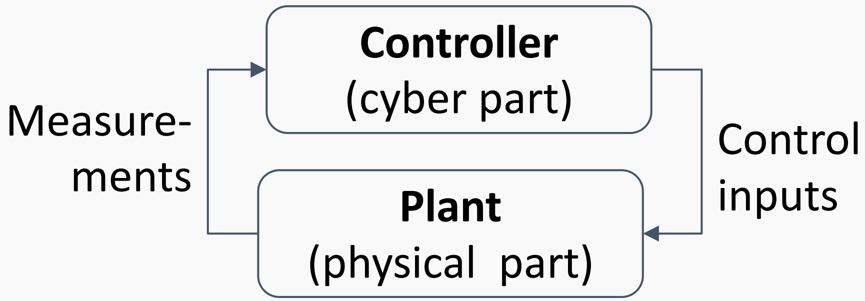
Outline

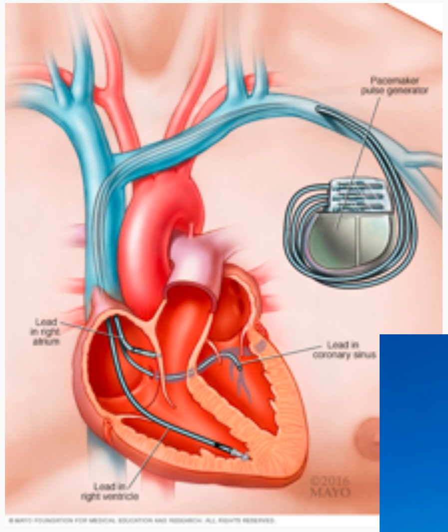
- Background
  - Reachability checking vs predictive monitoring for hybrid systems
- Neural Predictive Monitoring
  - Predictive monitoring with neural networks
  - Reject uncertain predictions with statistical guarantees
  - Active learning to improve uncertain predictions
- Experimental results

Hybrid system verification

Hybrid and cyber-physical systems are ubiquitous and found in many safety-critical applications

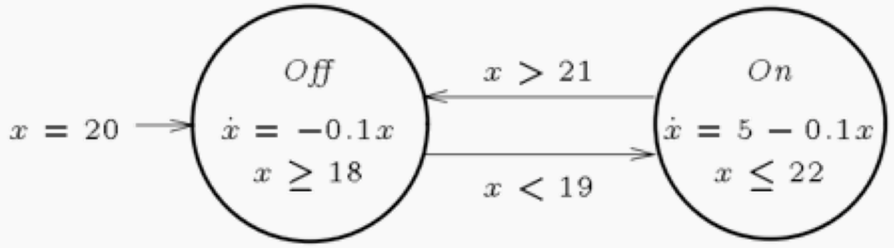
Cyber-physical system





Hybrid system verification

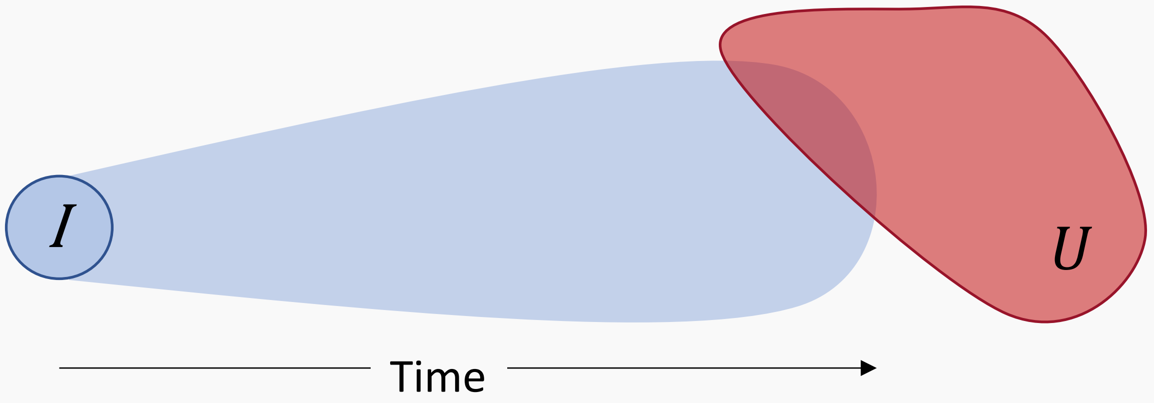
- Hybrid automata (HA) are a common formal model for hybrid and cyber-physical systems



Thermostat from *Henzinger, The Theory of Hybrid Automata*

- HA verification problem usually formulated as reachability

(Time-bounded) reachability:
can an HA $\mathcal{M}$, starting in an initial region I, reach a state $u \in U$ (within time T)?

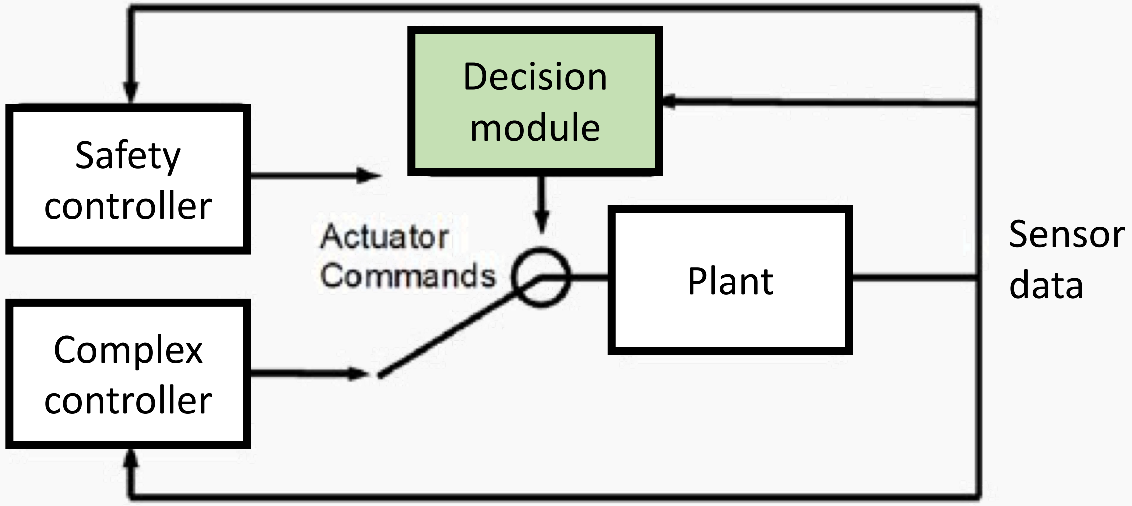


Both bounded and unbounded versions are undecidable

[Henzinger et al, *JCSS 57 1* (1998); Brihaye et al, *ICALP* (2011)]

Motivation – Predictive Monitoring (PM)

- PM: *predicting at runtime future violations from current state*
- PM is important for runtime safety assurance of HSs and CPSs
- For example, in the Simplex Architecture [Sha, *IEEE Software* (2001)], *decision module* gives control to *safety controller* if a potential safety violation is imminent.



Motivation - Predictive Monitoring (PM)

(Offline) Reachability checking

- Reachability from a (large) region
- One-off analysis, potentially long time horizons
- No hard time constraints

(Online) Predictive Monitoring

- Reachability from a single state
- Analysis is periodic $\Rightarrow$ short time horizons
- Strict time constraints

- Fully-fledged reachability checking is too expensive for online analysis
- At runtime, real system can deviate from offline model $\Rightarrow$ strong guarantees of reachability checking no longer valid
- **For PM, we need accurate and fast methods**

PM problem

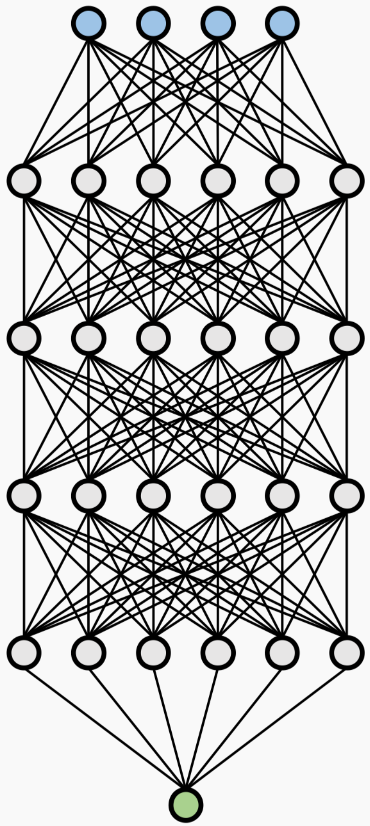
Problem 1 (Predictive monitoring for HA reachability). Given an HA $\mathcal{M}$ with state space X, time bound T, and set of unsafe states $U \subset X$, find a *predictor* h^*, i.e., a function $h^* : X \rightarrow \{0,1\}$ such that for all $x \in X$, $h^*(x) = 1$ if $\mathcal{M} \models \mathsf{Reach}(U,x,T)$, i.e., if it is possible for $\mathcal{M}$, starting in x, to reach a state in U within time T; $h^*(s) = 0$ otherwise.

A state $x \in X$ is called *positive* if $\mathcal{M} \models \mathsf{Reach}(U,x,T)$. Otherwise, x is *negative*.

THIS IS A BINARY CLASSIFICATION PROBLEM!

Neural networks (NNs) as state classifiers

- *Can we train an NN as a state classifier?*
- In principle, yes: NNs are universal approximators
  [Hornik et al, *Neural networks 2(5)* (1989)]
- Trained NN state classifier runs in **constant time** -> suitable for predictive monitoring
- Very good accuracy but prediction errors can't be entirely avoided



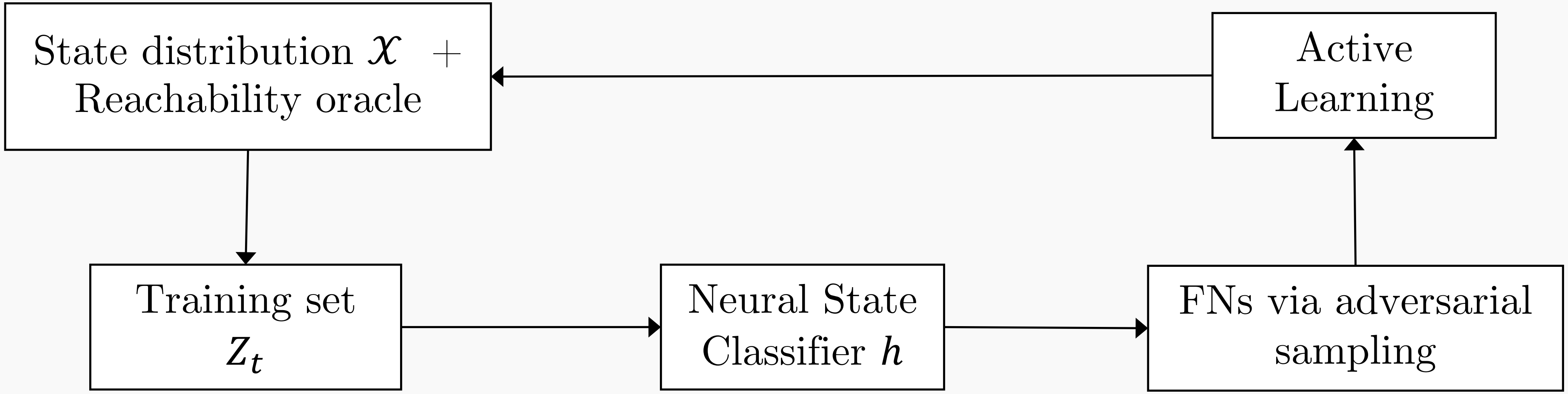
Neural networks (NNs) as state classifiers

Two kinds of prediction errors:

- **False positives (FPs):** a negative state is predicted to be positive
  - Conservative decision
- **False negatives (FNs):** a positive state is predicted to be negative
  - Can compromise system's safety!

https://xkcd.com/1838/

Neural State Classification [ATVA'18]

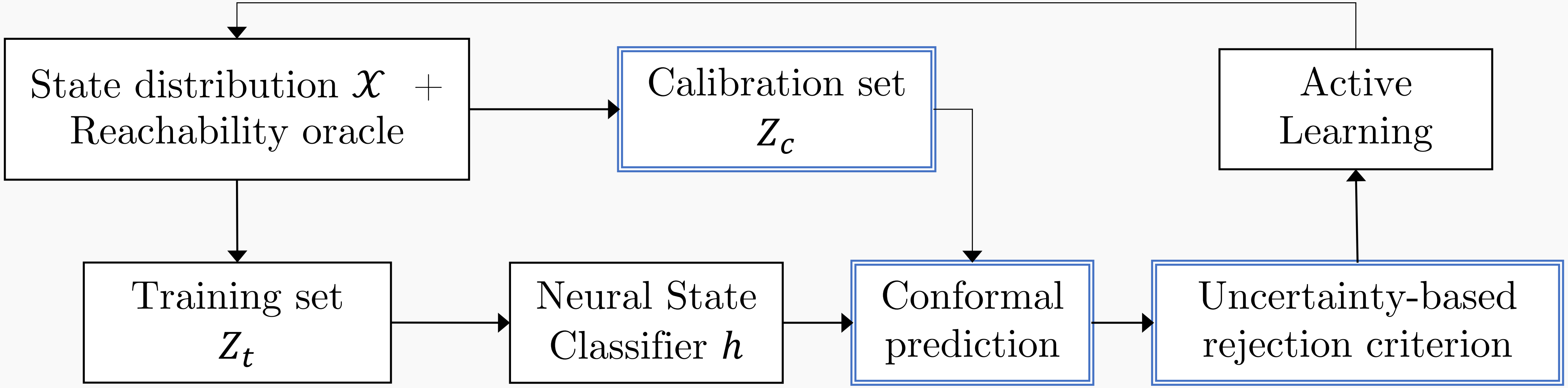


$h(x)$ = likelihood that state x is positive.

Limitation: it can't detect and prevent prediction errors at runtime

D. Phan et al., Neural state classification for hybrid systems. In *Proc. ATVA 2018*.

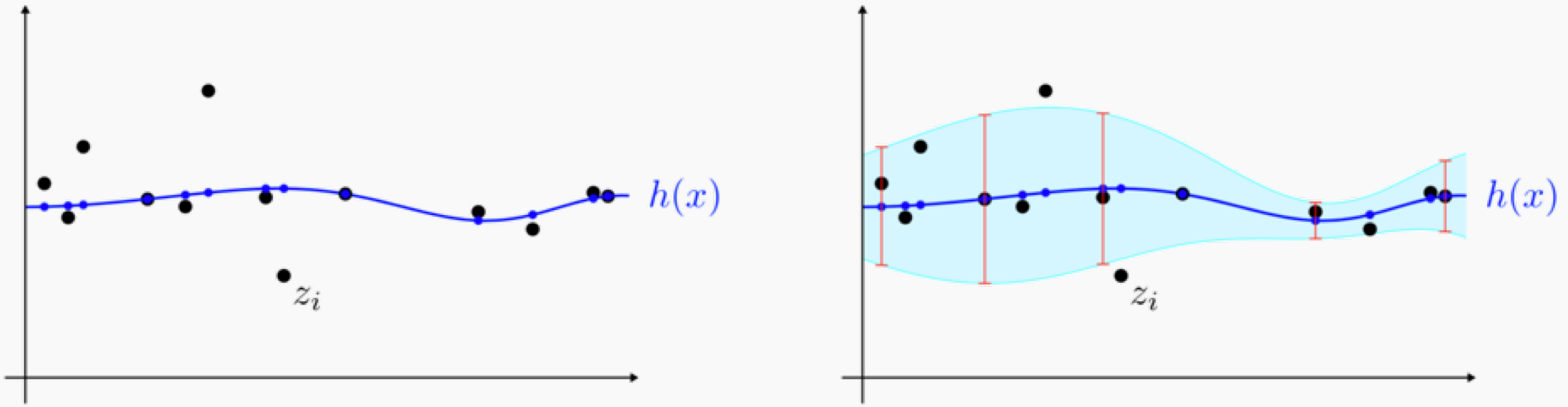
Neural Predictive Monitoring [this work]



- **Conformal Prediction** [Vovk et al] provides **statistical guarantees** on machine learning predictions
- Allows one to derive **sound measures of prediction uncertainty**, which we use to **reject unreliable predictions**, more likely to be wrong

Conformal prediction

- CP works on top of any supervised learning model
- CP complements single-point predictions with a **prediction region** and **uncertainty measures**
- Given significance $\epsilon \in (0,1)$ and a test point x^*, prediction region Γ_*^ϵ is guaranteed to contain the true class of x^* with probability $1 - \epsilon$
- CP is distribution-free (only assumption is exchangeability, a weaker version of iid)

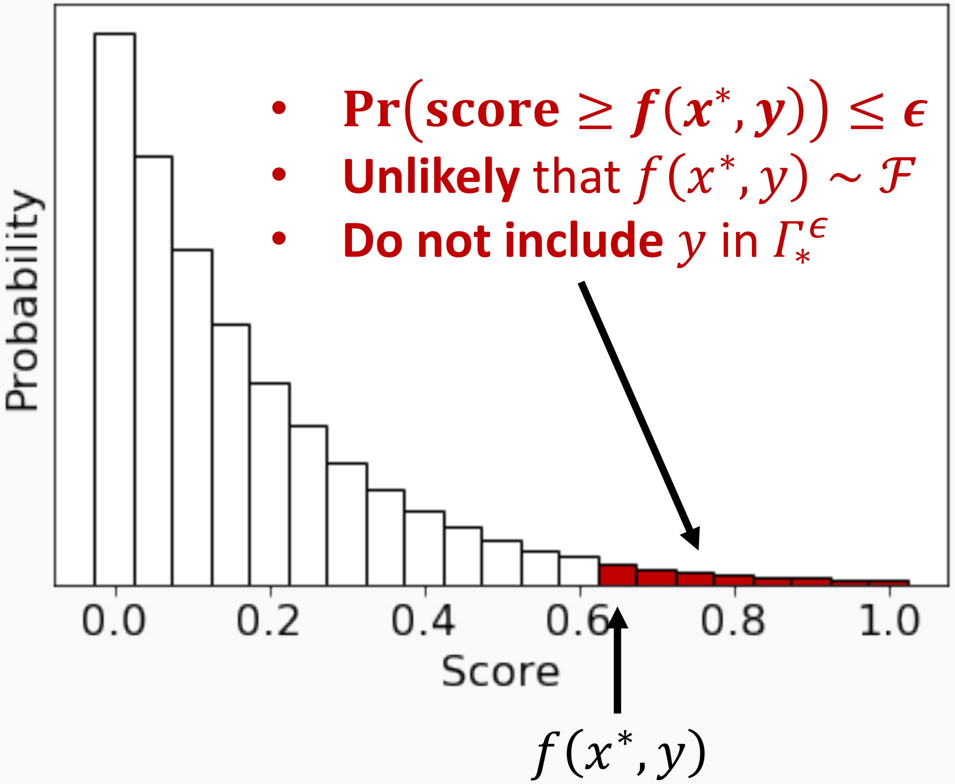


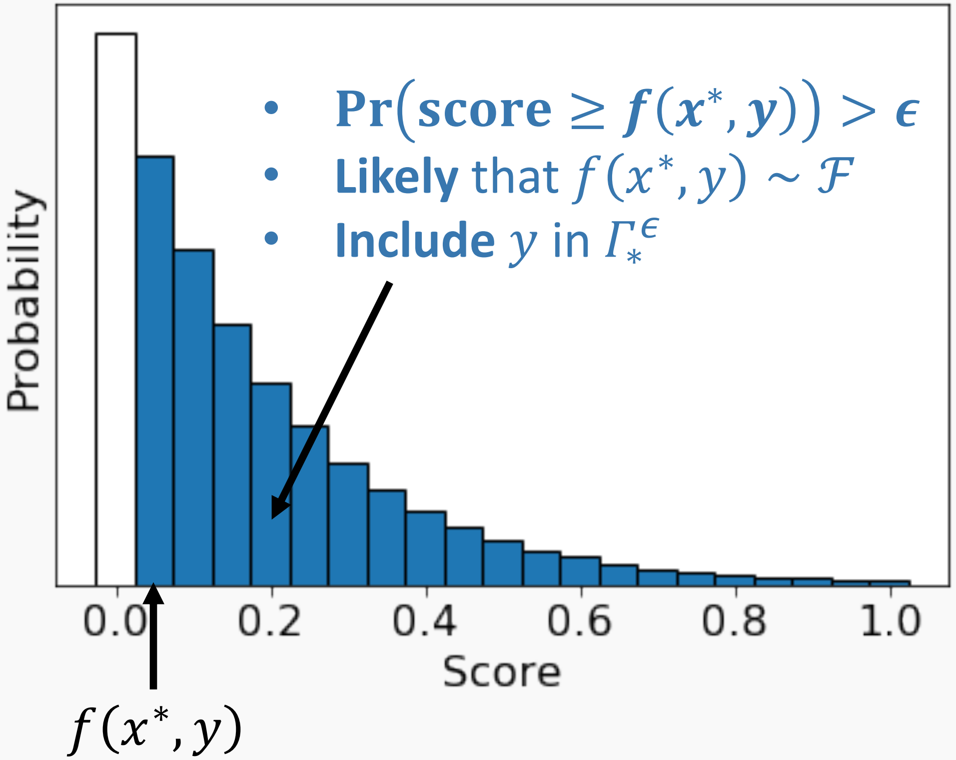
Conformal prediction – Idea (1/2)

- Prediction region Γ_*^ϵ contains the classes likely to be true
- Define non-conformity function (NCF) f that, for a point (x, y), measures the distance between y and the model prediction $h(x)$
  - In our case, $f(x, y) = |y - h(x)|$ $(h(x) \in [0,1], y \in \{0,1\})$
  - The distribution of scores $\boldsymbol{\mathcal{F}} = \mathbf{Pr}_{\boldsymbol{x} \sim \boldsymbol{\mathcal{X}}}(\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{h}^*(\boldsymbol{x}))$ fully characterizes distance between predictions and true classes
- True $\boldsymbol{\mathcal{F}}$ is unknown → estimate it using a set of calibration points Z_c sampled from $\boldsymbol{\mathcal{X}}$ and disjoint from training set
  - Resulting empirical distribution converges to true distribution for large samples

Conformal prediction – Idea (2/2)

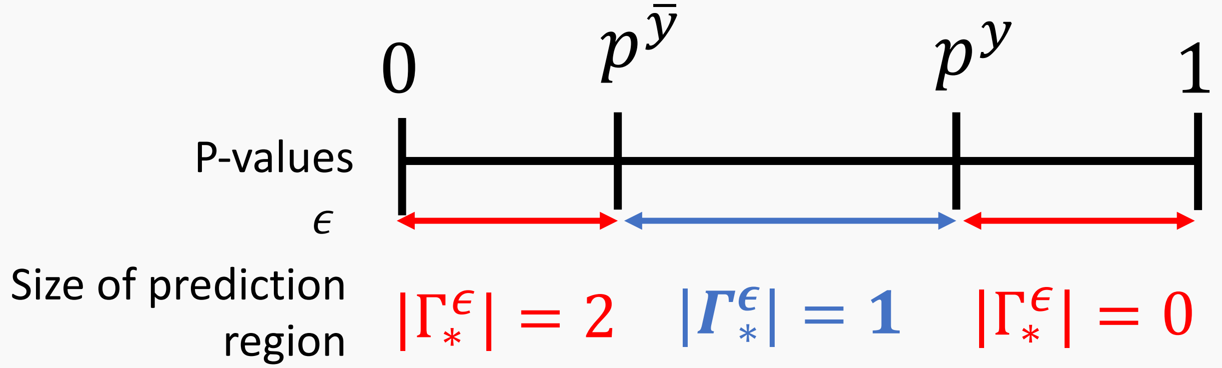
- Γ_*^ϵ for test point x^* contains all y s.t. it is likely that $f(x^*, y) \sim \mathcal{F}$
  - hypothesis testing at level ϵ of $H_0: f(x^*, y) \sim \mathcal{F}$ VS $H_a: f(x^*, y) \nsim \mathcal{F}$





Prediction reliability measures

- Let y be the class predicted by h. Call p^y the p-value $\Pr(score \geq f(x^*, y))$
- Easy to see that $p^y \geq p^{\bar{y}}$ ($\bar{y} = \{0,1\} \setminus \{y\}$)
  - Because $f(x^*, y) \leq f(x^*, \bar{y})$



- **Prediction is reliable when $|\Gamma_*^\epsilon| = 1$** (i.e., Γ_*^ϵ contains only one class, the true one with probability $1 - \epsilon$)
- high p^y and low $p^{\bar{y}}$ → large range of ϵ values for which $|\Gamma_*^\epsilon| = 1$

Uncertainty-based rejection criterion

- **Idea:** at runtime, reject all reachability predictions with low values of p^y (aka credibility, c) and $1 - p^{\bar{y}}$ (aka confidence, $1 - \gamma$)
- Very efficient criterion → it reduces to just computing two p-values
- Independent of the choice of ϵ
- *But how to select thresholds for $1 - \gamma$ and c?*
- Learn $1 - \gamma$ *and* c thresholds that optimally separate correct and wrong predictions

Learning optimal rejection thresholds

- Cross validation strategy using Z_c as validation set
  - compute $1-\gamma^i$ and c^i for each calibration point i (after removing i from Z_c)
- Train two support vector classifiers (SVCs) over $\{(1-\gamma^i, err^i)\}_i$ and $\{(c^i, err^i)\}_i$ (err^i true iff h correctly predicts point i)
- Results in thresholds $1-\gamma_\tau$ and c_τ below which prediction is rejected
  - Four thresholds if we distinguish between FN and FP errors
- **The rejection criterion is optimal**
  - SVCs maximize separation between classes.
  - 1-dimensional input, so linear SVCs suffice

Uncertainty-based active learning

- **Idea**: retrain after augmenting training and calibration sets with rejected sample, to improve prediction accuracy and rejection rate

Algorithm

1. Draw a random input sample. Keep only rejected (unreliable) points R.
2. Label R using reachability oracle. Redistribute samples into training and calibration sets.
3. Train a new predictor on augmented training set
4. Train new rejection thresholds on augmented calibration set
5. Repeat 1-4 as desired

Experimental evaluation

Initial training

Model	accuracy	fp	fn	rej. rate
Spiking Neuron (SN)	99.582%	24.4/24.6	17.2/17.2	5.68%
Artificial Pancreas (AP)	99.488%	30.4/30.6	20.6/20.6	6.23%
Helicopter (HE)	99.180%	47.4/48.8	33/33.2	9.88%
Water Tank (WT)	99.818%	8.6/8.6	9.6/9.6	5.97%
Cruise Controller (CC)	99.848%	8.2/8.2	7/7	3.46%

20K training set (70% training, 30% calibration). 100K for Helicopter. 10K test set.
Results averaged over 5 runs.

- **Rejection criterion identifies almost all FP and FN errors**
- Excessive rejection rate

Experimental evaluation

Passive re-training (random samples) vs Active Learning

		PASSIVE			ACTIVE			
Model	# samples	**fp**	**fn**	rej. rate	accuracy	**fp**	**fn**	rej. rate
SN	5748.2	18.2/18.2	10.6/10.8	3.91%	99.918%	2.8/2.8	5.4/5.4	1.16%
AP	6081.8	23/23.4	19.4/19.4	5.94%	99.892%	6.2/6.2	4.4/4.6	1.02%
HE	22014.6	31.4/31.6	26/26.6	7.21%	99.772%	11.2/11.2	10.4/11.6	2.74%
WT	4130.2	8.4/8.4	10.2/10.4	4.43%	99.962%	2.8/2.8	1/1	0.70%
CC	2280.6	6/6	6/6	5.15%	99.962%	2/2	1.8/1.8	0.51%

One re-training iteration. Re-training samples selected from batches of 200K (500K for helicopter)

- **Active learning greatly reduces prediction error and rejection rate**
- No significant improvement with passive approach

Related work on predictive monitoring

- Linear systems [Chen et al, *RTSS* (2017), Yoon et al, *RV* (2019)]
- Discrete-space Markov models
  [Babaee et al, *RV* (2018), *RV* (2019)]
- Prediction regions for STL over ARMA models [Quin et al, *HSCC* (2019)]
- Neural approximation of PDEs for HJ reachability [Djeridane et al, *CDC* (2006)]
  [Rubies-Royo et al, *arXiv:1803.03237* (2019)]
- Smoothed model checking: Gaussian processes to approximate the satisfaction function of continuous-time Markov chains [Bortolussi et al, *Information and Computation 247* (2016)]

Summary

- Method to derive predictive monitors for hybrid systems
- Based on neural networks → high prediction accuracy
- Optimal uncertainty-based rejection criteria with statistical guarantees based on conformal prediction
- Computationally efficient → suitable for runtime analysis
- Active learning to improve accuracy and reduce rejection rate