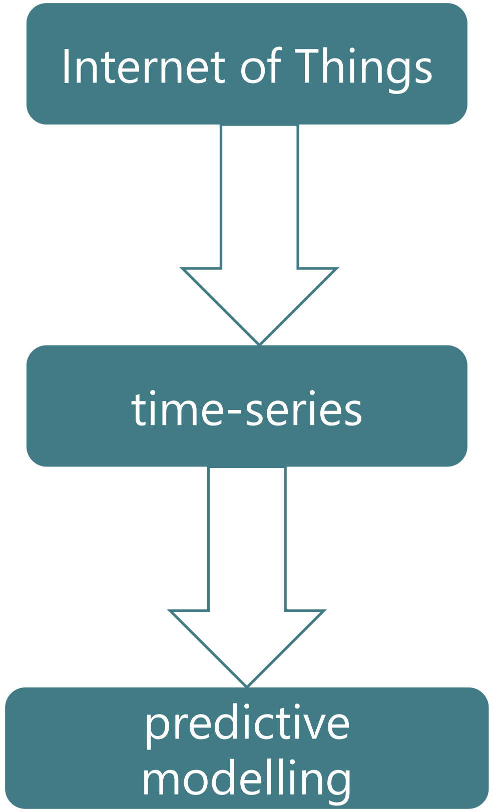
Learning Temporal Logic Formulas from Time-series Data

Laura Nenzi
University of Trieste, Italy

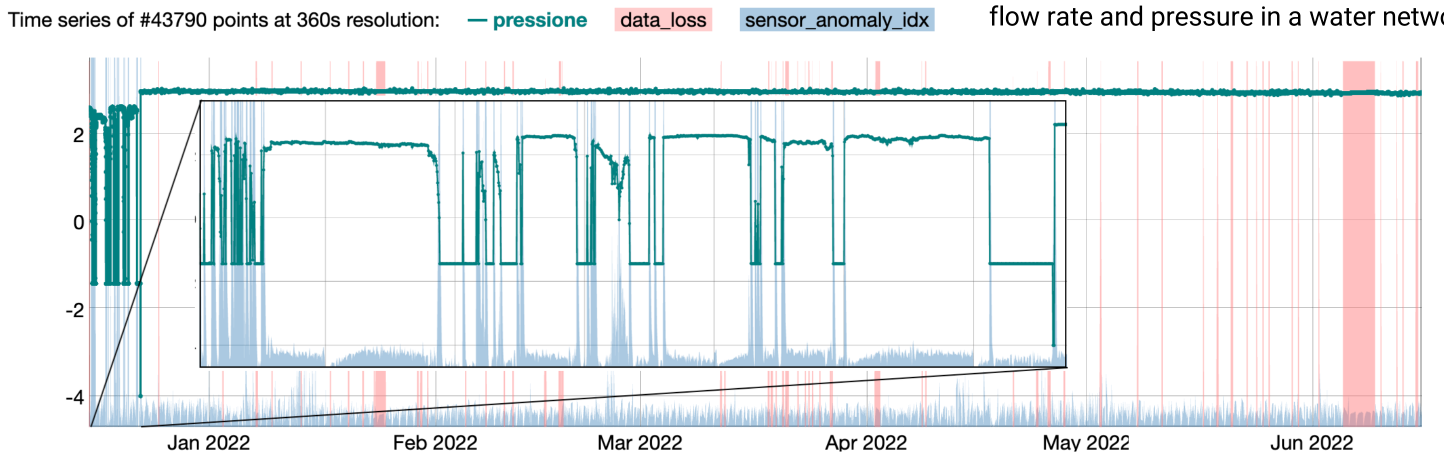
Joint work with L. Bortolussi, E. Medvet, J. V. Deshmukh, S. Silvetti, F. Pigozzi, P. Indri, S. Mohammadinejad, E. Bartocci, and A. Bartoli, Gaia Saveri, Jan Kretisky

NCSR, National Centre of Scientific Research "Demokritos"
Athens, September 26, 2023

Context and Problem

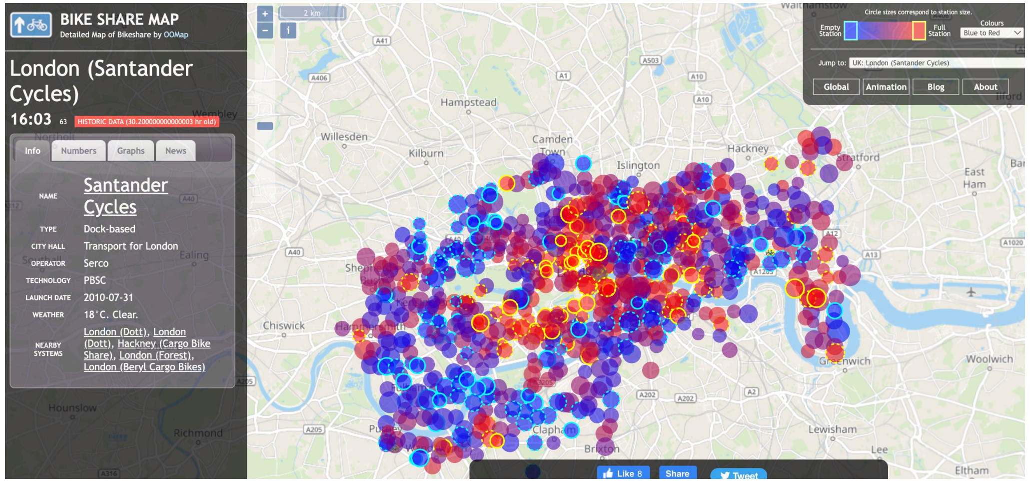


(e.g. anomaly detection)



flow rate and pressure in a water network

courtesy of idrostudi s.r.l.



Need of human-interpretable models

Menu of the day

- Starter: Formal specification
  - Signal Temporal Logic (STL)
  - Spatio-Temporal Reach and Escape Logic (STREL)
- Main: Temporal Logic requirement mining
  - STL classifier (supervised and semi-supervised learning)
  - STREL-based clustering (unsupervised learning)
- Dessert: related and ongoing work
  - Fruit Salad
  - Some heavy cake

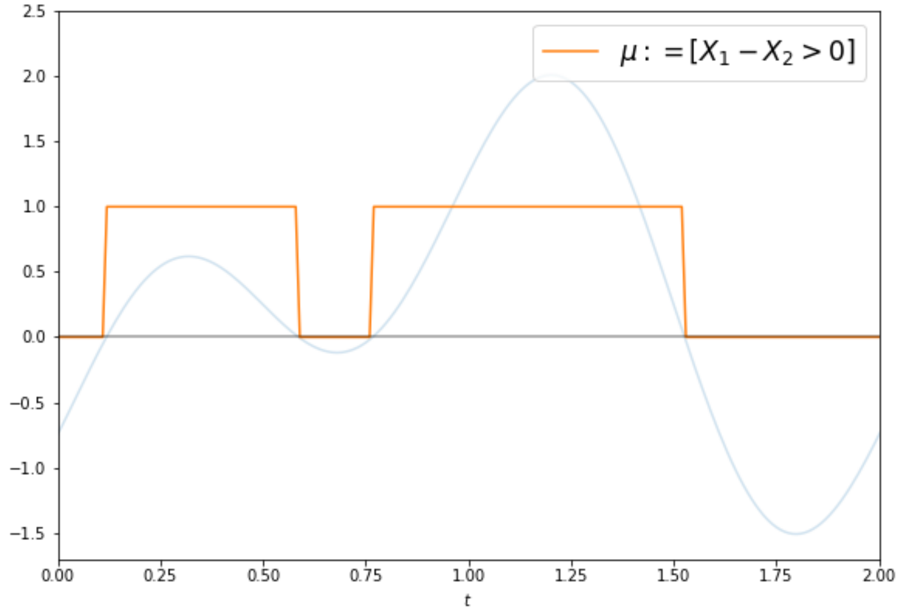
Formal Specification

Signal Temporal Logic (STL)

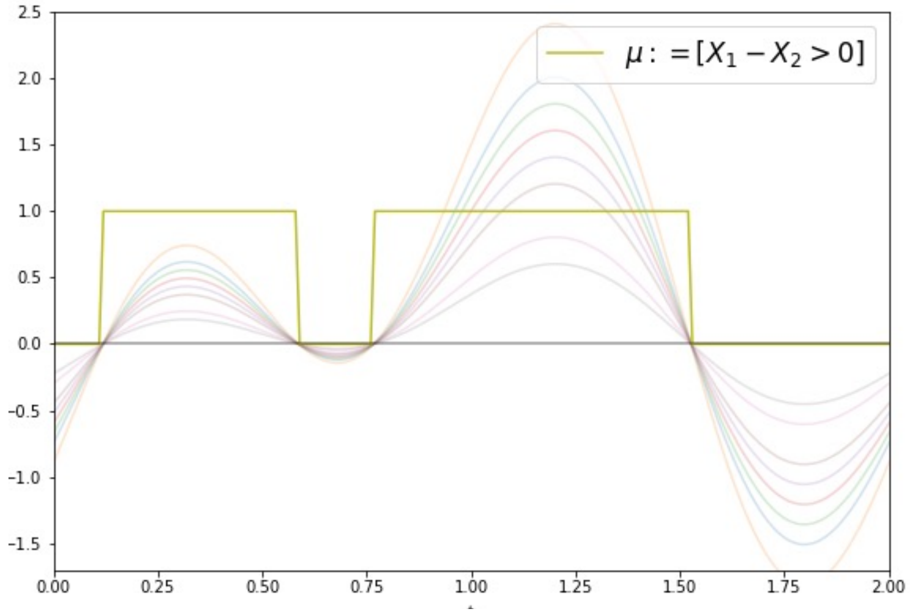
STL extends MITL by having signal predicates over real values as atomic formulas:

STL Syntax	$\varphi := true \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathrm{U}_I \, \varphi_2$

In addition $F_I\varphi := \top \, U_I\varphi$ $\quad G_I\varphi := \neg F_I \neg\varphi$



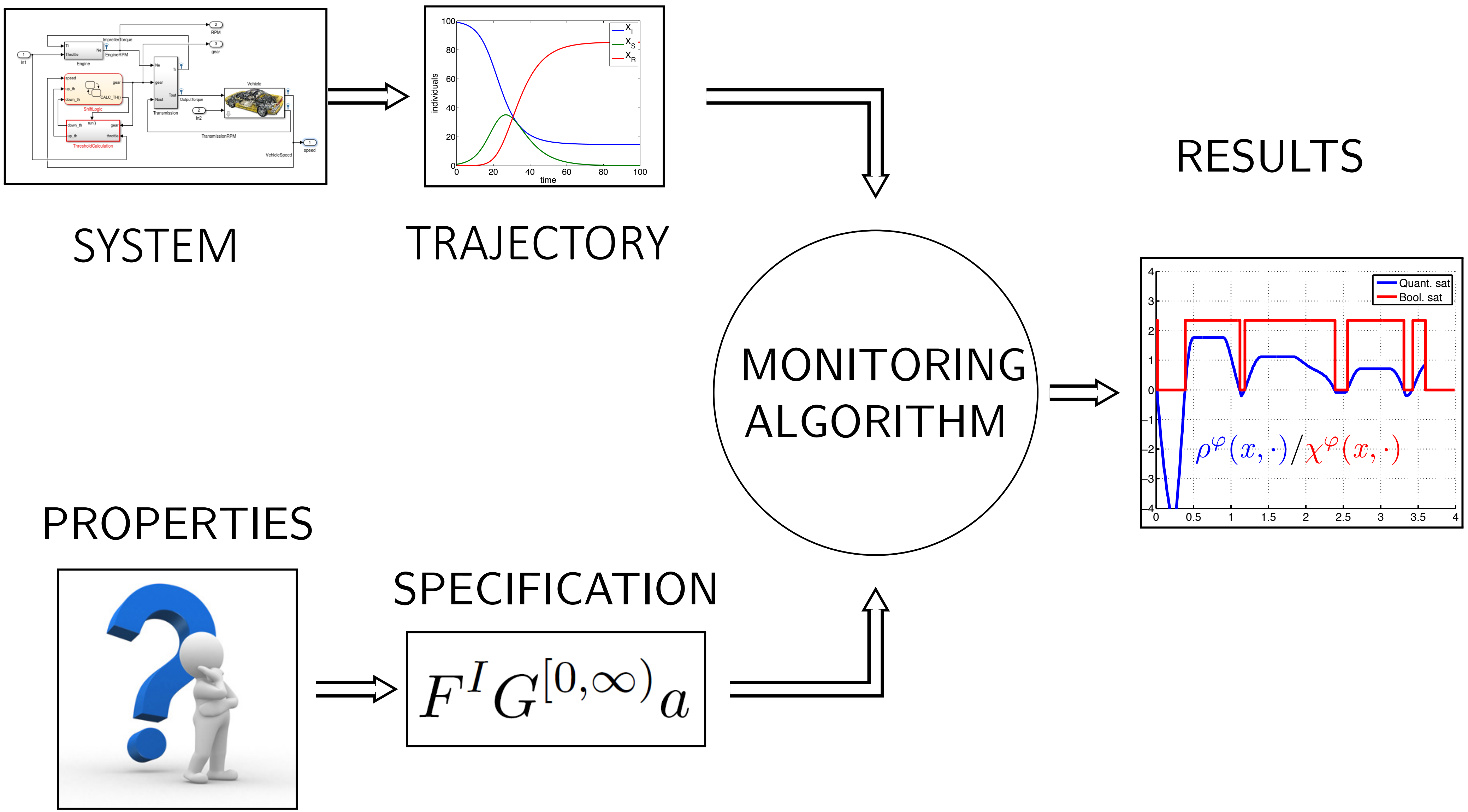
Boolean semantics: $\chi(\vec{x}, t, \varphi) \in \{0, 1\}$



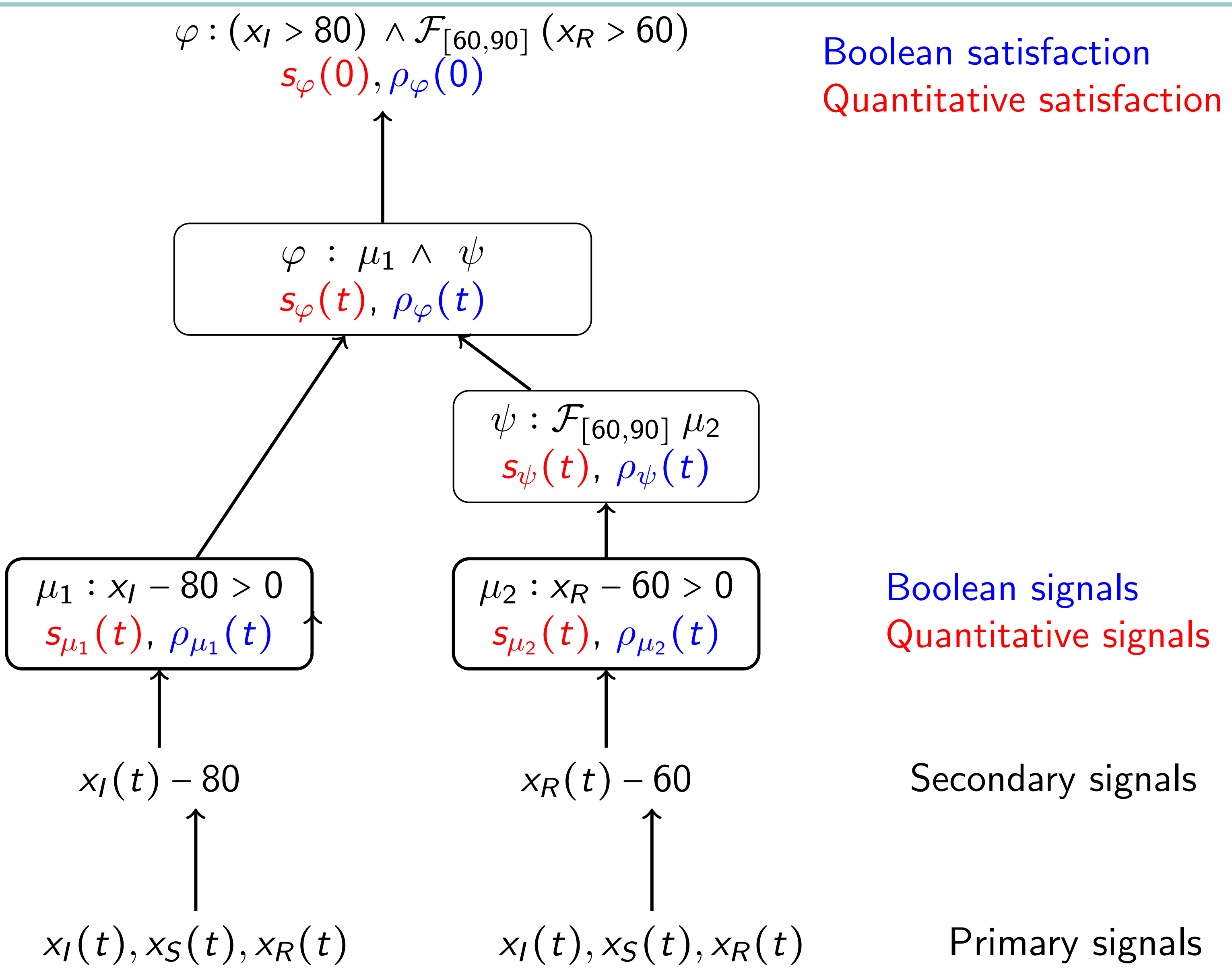
Quantitative semantics: $\rho(\vec{x}, t, \varphi) \in \mathbb{R} \cup \{+\infty, -\infty\}$

[O. Maler, D. Nickovic:, Monitoring Temporal Properties of Continuous Signals. FORMAT 2004]

Monitoring STL



Monitoring STL



Parametric Signal Temporal Logic (PSTL)

Definition (PSTL syntax)

$$\phi := (x_i \bowtie \pi) \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathcal{U}_{[\tau_1, \tau_2]} \, \varphi_2$$

with $\bowtie \in \{>, \leq\}$

- π is **threshold** parameter
- τ_1, τ_2 are **temporal** parameters

- $\mathbb{K} = (\mathcal{T} \times \mathcal{C})$ be the **parameter space**
- $\boldsymbol{\theta} \in \mathbb{K}$ is a **parameter configuration**

e.g., $\phi = \mathcal{F}_{[a,b]}(x_i > k), \boldsymbol{\theta} = (0, 2, 3.5)$ then $\phi_{\boldsymbol{\theta}} = \mathcal{F}_{[0,2]}(x_i > 3.5)$.

Spatio-Temporal Reach and Escape Logic (STREL):

STREL is extension of STL with two spatial operators: Reach and Escape

It considers a discrete space described as a weighted (direct) graph

STREL Syntax

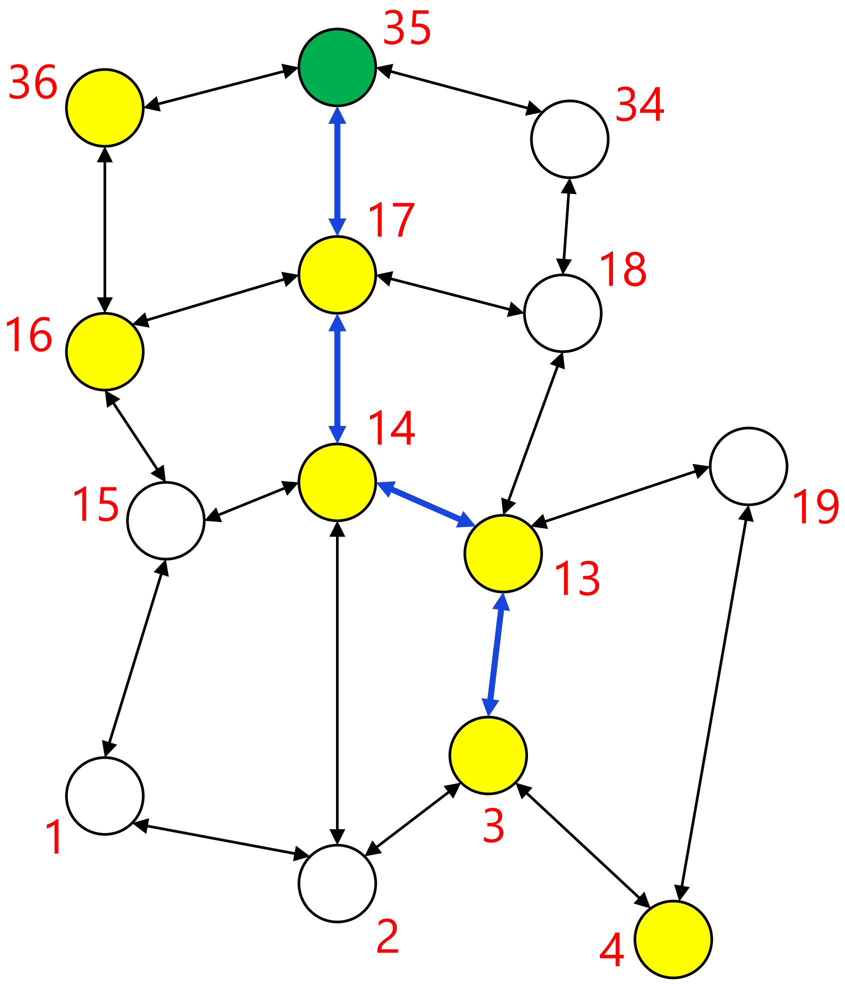
$$\varphi := true \mid \mu \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \, \mathrm{U}_I \, \varphi_2 \mid \varphi_1 \, \mathrm{S}_I \, \varphi_2 \mid \varphi_1 \, \mathcal{R}^f_d \, \varphi_2 \mid \mathcal{E}^f_d \, \varphi$$

Signal Temporal Logic — Reach — Escape

- Somewhere, Everywhere and Surround operators can be derived from Reach and Escape

[Nenzi et al , A Logic for Monitoring Dynamic Networks of Spatially-distributed Cyber-Physical Systems. LMCS, 2022]
[Nenzi et al, Monitoring spatio-temporal properties, invited tutorial., 2020]

Reach operator ($\mathcal{R}$)

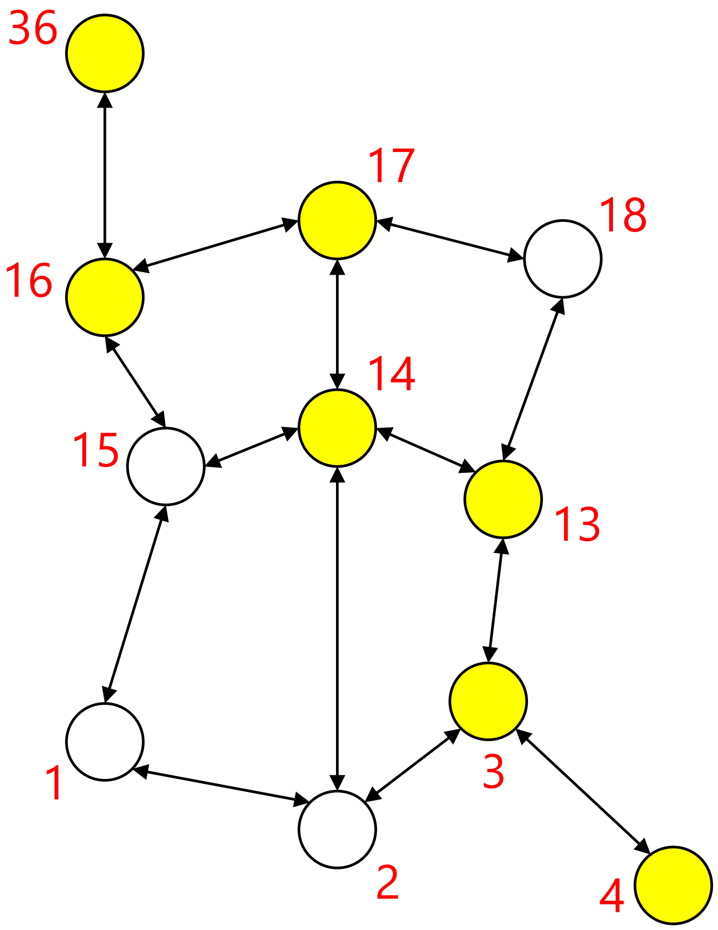


$\varphi = yellow\ R_{[1,4]}\ green$

l_3 *satisfies* φ
path $= l_3, l_{13} l_{14} l_{17} l_{35}$

l_4 *does not satisfy* φ

Everywhere operator (□)

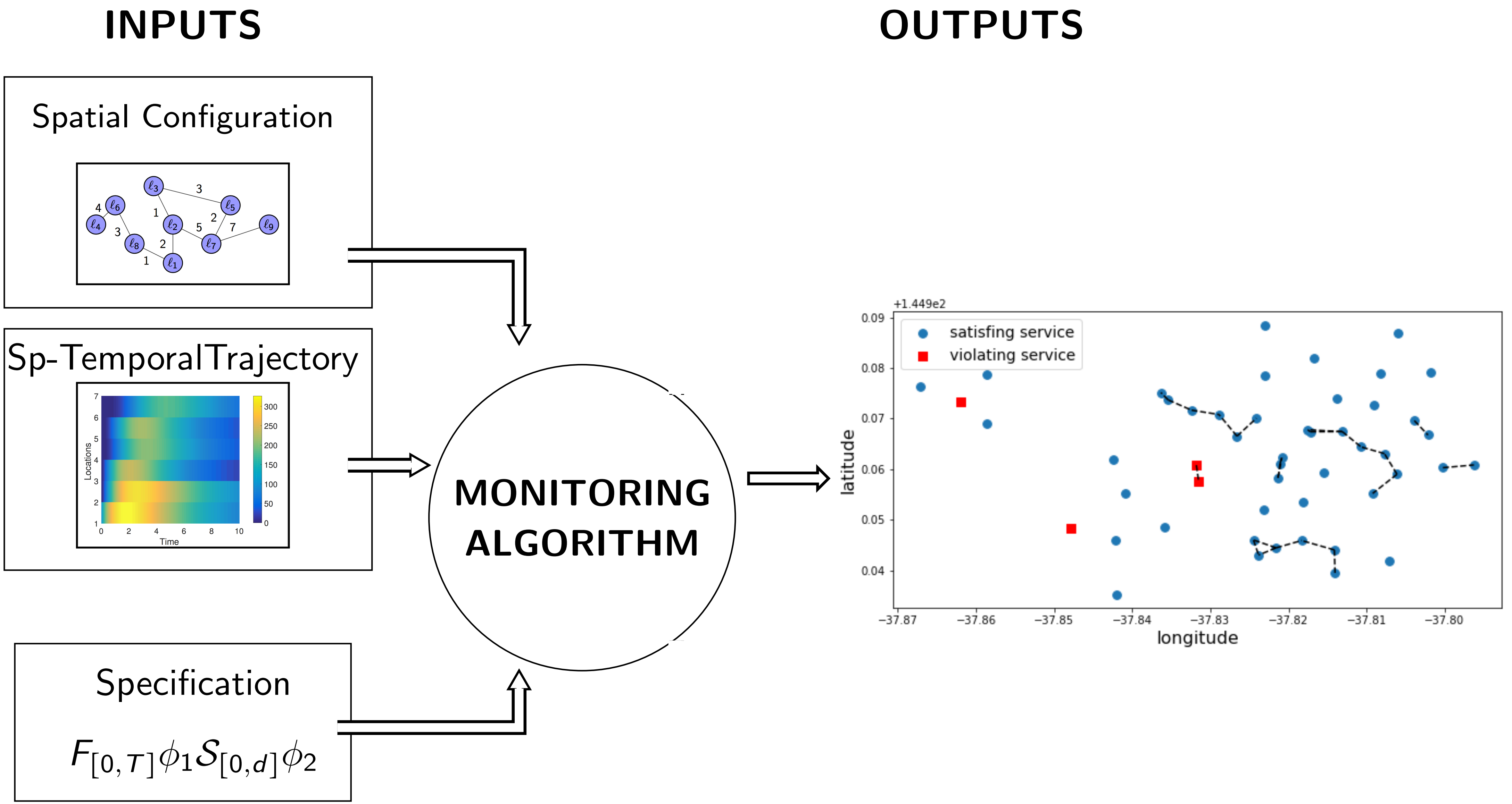


$\varphi = \square_{[2,3]} yellow$

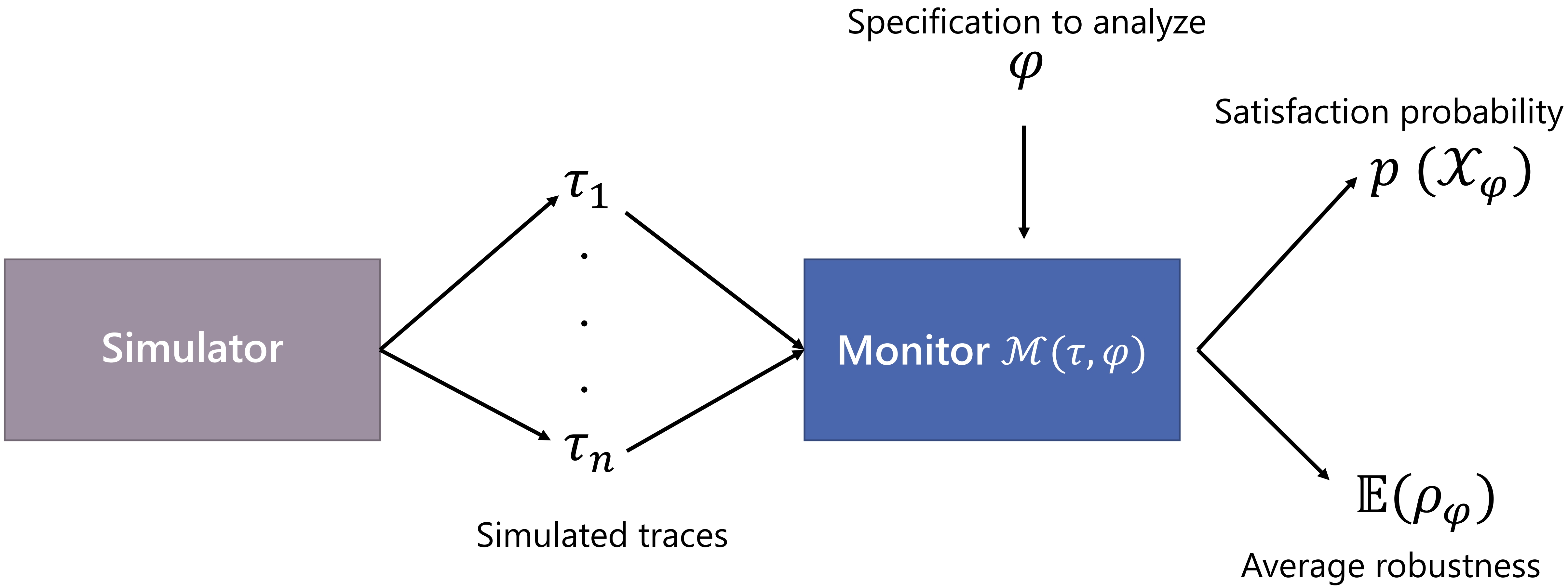
l_1 *satisfies* φ

l_2 *does not satisfy* φ

Monitoring STREL

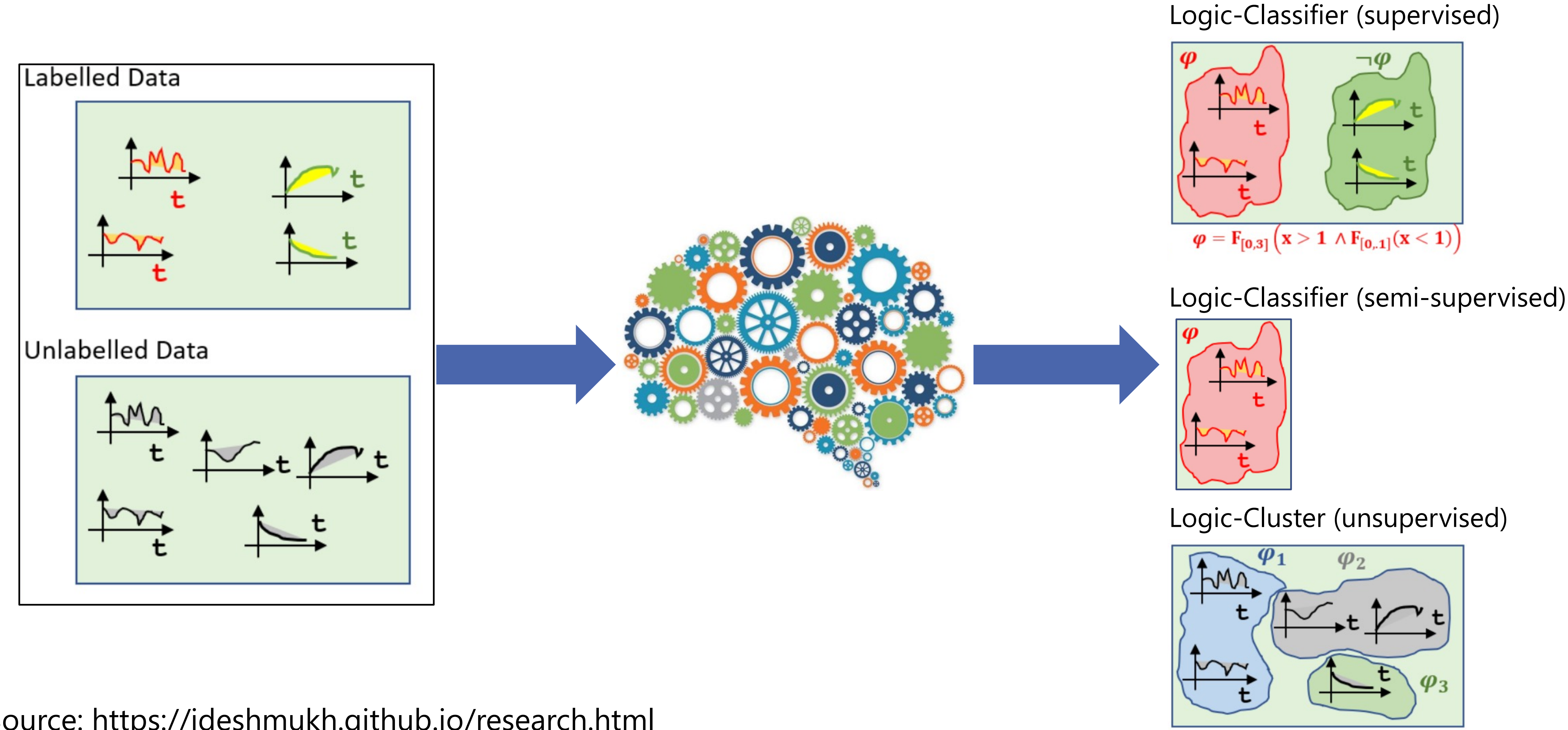


Statistical Model Checking



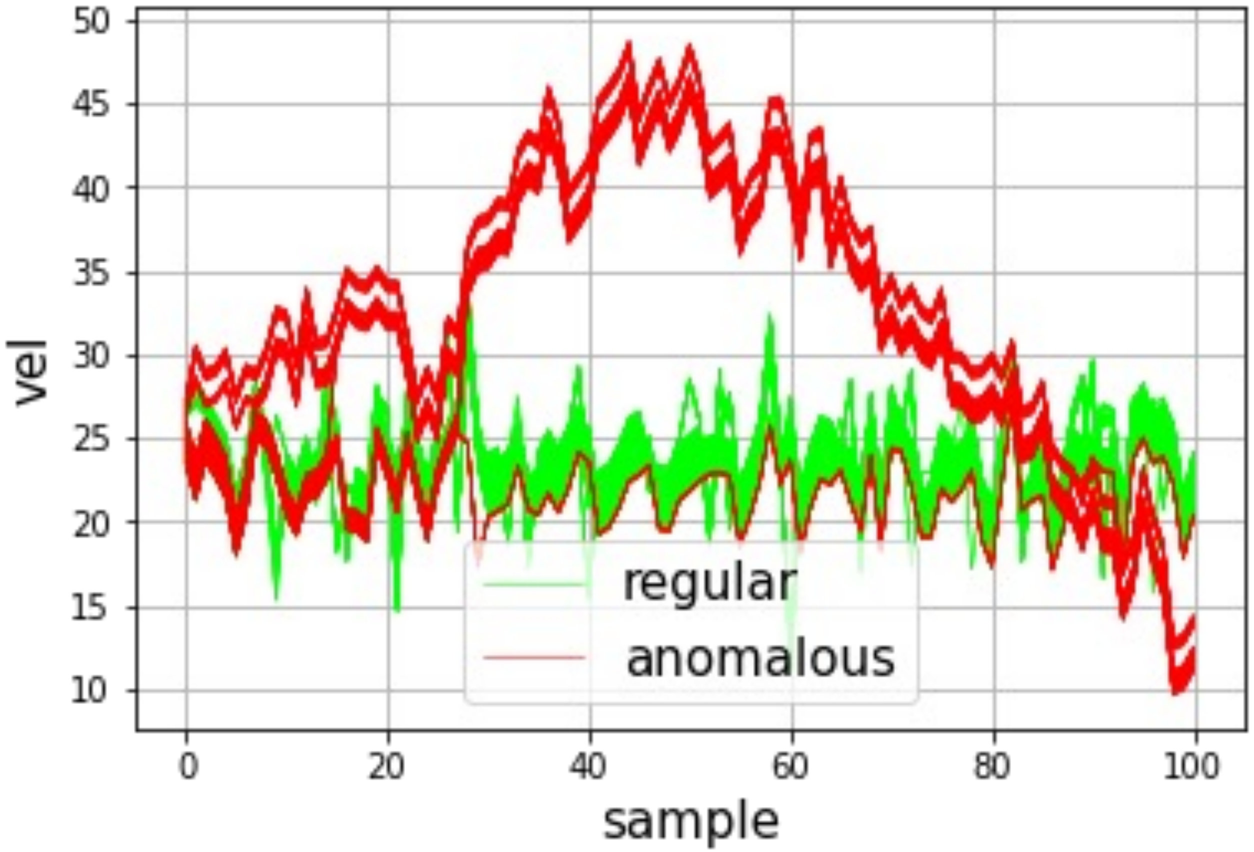
TL Requirement mining

Temporal Logic requirement mining



Source: https://jdeshmukh.github.io/research.html

STL Classifiers ((Semi-)Supervised Learning)



Goal: learning a specification/ classifier as a temporal logic formula to discriminate as much as possible between regular and anomalous behaviours.

We want to learn both the structure and the parameters of the formula

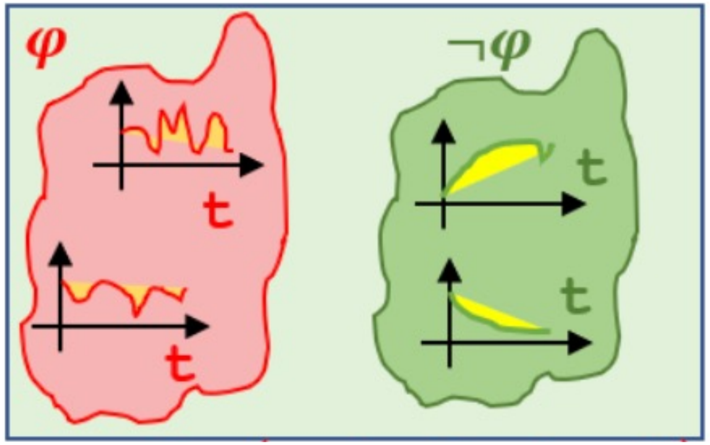
STL Classifier: Problem Statement

We want a way to search in the space of STL formulae considering training data X_{learn}

Supervised two-class classification problem

Training data set: two sets

- regular X^+_{learn}
- anomalous X^-_{learn}

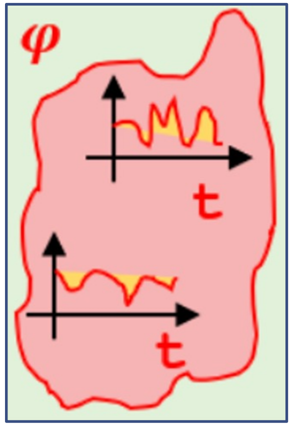


Find the best φ that better separates the two sets.

Semi-supervised one-class classification prob

Training data set: one set

- regular X^+_{learn}



Find the "tight" φ that is satisfied by the set

STL classifier (supervised): ROGE

- Bi-level algorithm:
  - learning formula structure via Genetic Programming (GP)
  - learn parameters of the formula using by Bayesian Optimisation

- A **fitness function** f measures the quality of candidate solutions and depends on the kind of problem at hand (two-classes, one-class)

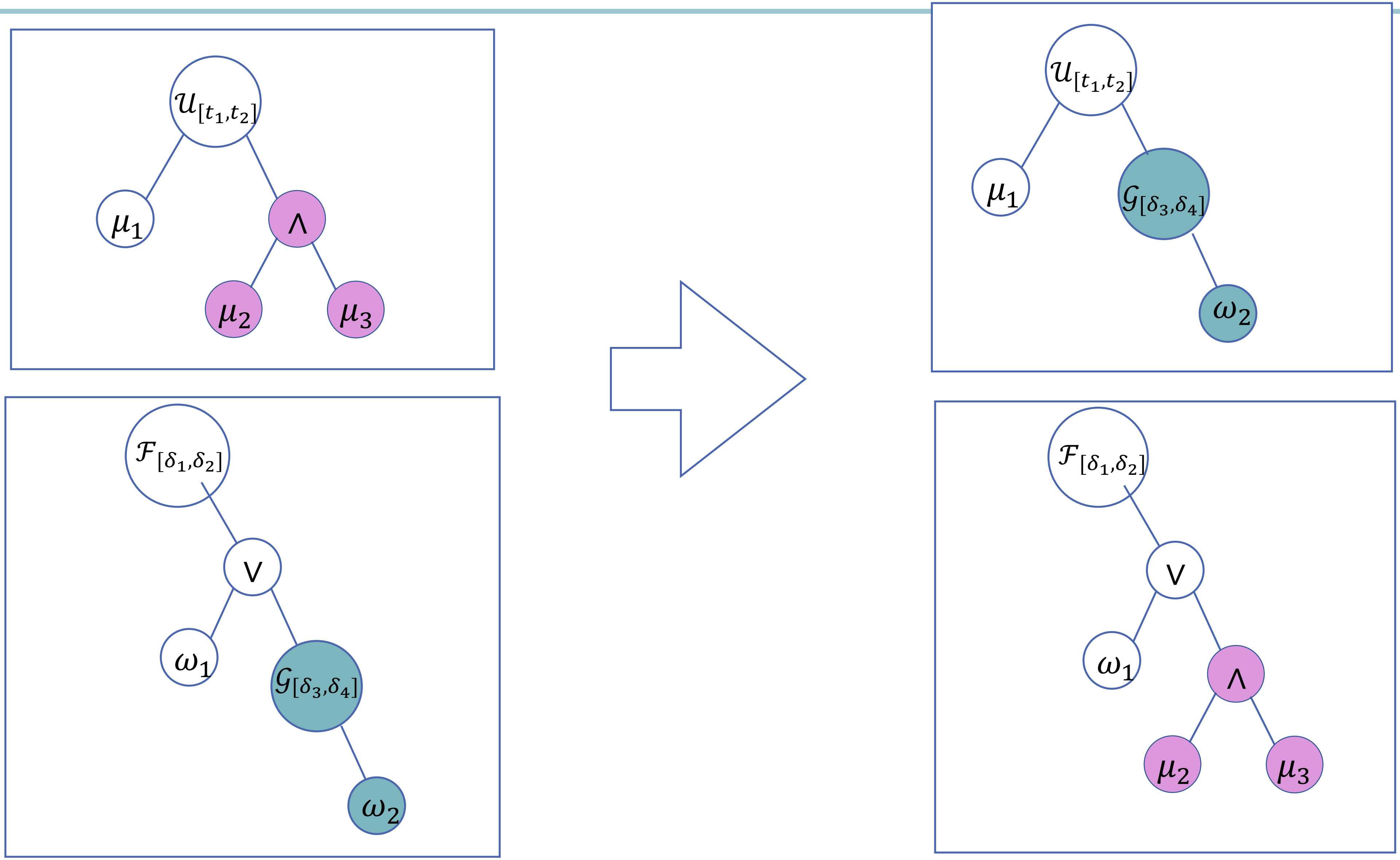
$$f(\varphi; X^+_{\text{learn}}, X^-_{\text{learn}}) = -\frac{\mathbb{E}_{X^+_{\text{learn}}}(\rho_\varphi) - \mathbb{E}_{X^-_{\text{learn}}}(\rho_\varphi)}{\sigma_{\varphi, X^+_{\text{learn}}} + \sigma_{\varphi, X^-_{\text{learn}}}}$$

Require: $\mathcal{D}_p, \mathcal{D}_n, \mathbb{K}, Ne, Ng, \alpha, s$

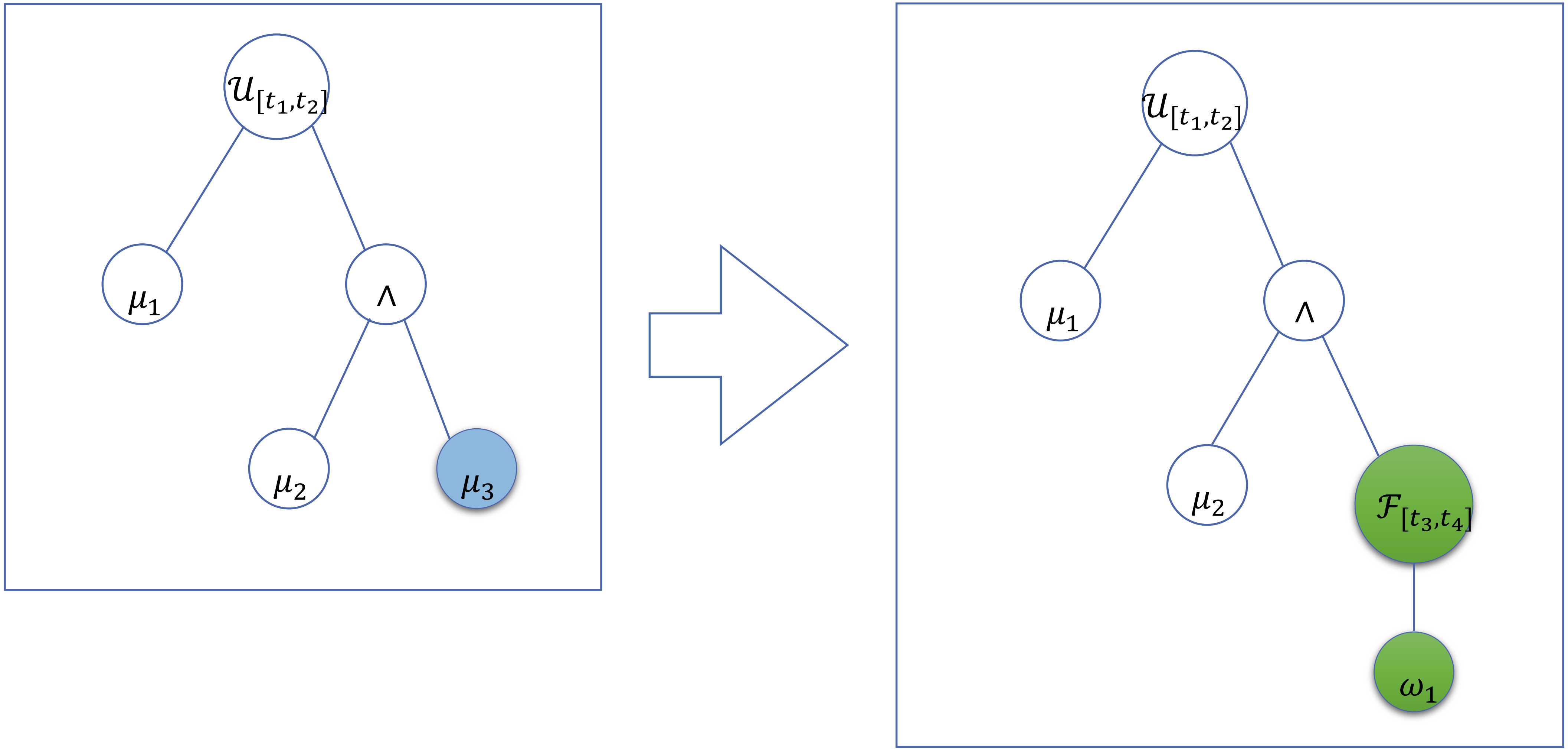
```
1: gen ← GENERATEINITIALFORMULAE(Ne, s)
2: gen_Θ ← LEARNINGPARAMETERS(gen, G, 𝕂)
3: for i = 1 ... Ng do
4:     subg_Θ ← SAMPLE(gen_Θ, F)
5:     newg ← EVOLVE(subg_Θ, α)
6:     newg_Θ ← LEARNINGPARAMETERS(newg, G, 𝕂)
7:     gen_Θ ← SAMPLE(newg_Θ ∪ gen_Θ, F)
8: end for
9: return gen_Θ
```

[L. Nenzi, S. Silvetti, E. Bartocci, L. Bortolussi: A Robust Genetic Algorithm for Learning Temporal Specifications from Data. QEST 2018]

Crossover Operator



Mutation Operator



STL classifier (supervised): ROGE

- Bi-level algorithm:
  - learning formula structure via Genetic Programming (GP)
  - learn parameters of the formula using by Bayesian Optimisation
- A **fitness function** f measures the quality of candidate solutions and depends on the kind of problem at hand (two-classes, one-class)

$$f(\varphi; X^+_{\text{learn}}, X^-_{\text{learn}}) = -\frac{\mu_{\varphi, X^+_{\text{learn}}} - \mu_{\varphi, X^-_{\text{learn}}}}{\sigma_{\varphi, X^+_{\text{learn}}} + \sigma_{\varphi, X^-_{\text{learn}}}}$$

Require: $\mathcal{D}_p, \mathcal{D}_n, \mathbb{K}, Ne, Ng, \alpha, s$

$gen \leftarrow$ GENERATEINITIALFORMULAE(Ne, s)

$gen_\Theta \leftarrow$ LEARNINGPARAMETERS$(gen, G, \mathbb{K})$

for $i = 1 \dots Ng$ **do**

  $subg_\Theta \leftarrow$ SAMPLE(gen_Θ, F)

  $newg \leftarrow$ EVOLVE$(subg_\Theta, \alpha)$

  $newg_\Theta \leftarrow$ LEARNINGPARAMETERS$(newg, G, \mathbb{K})$

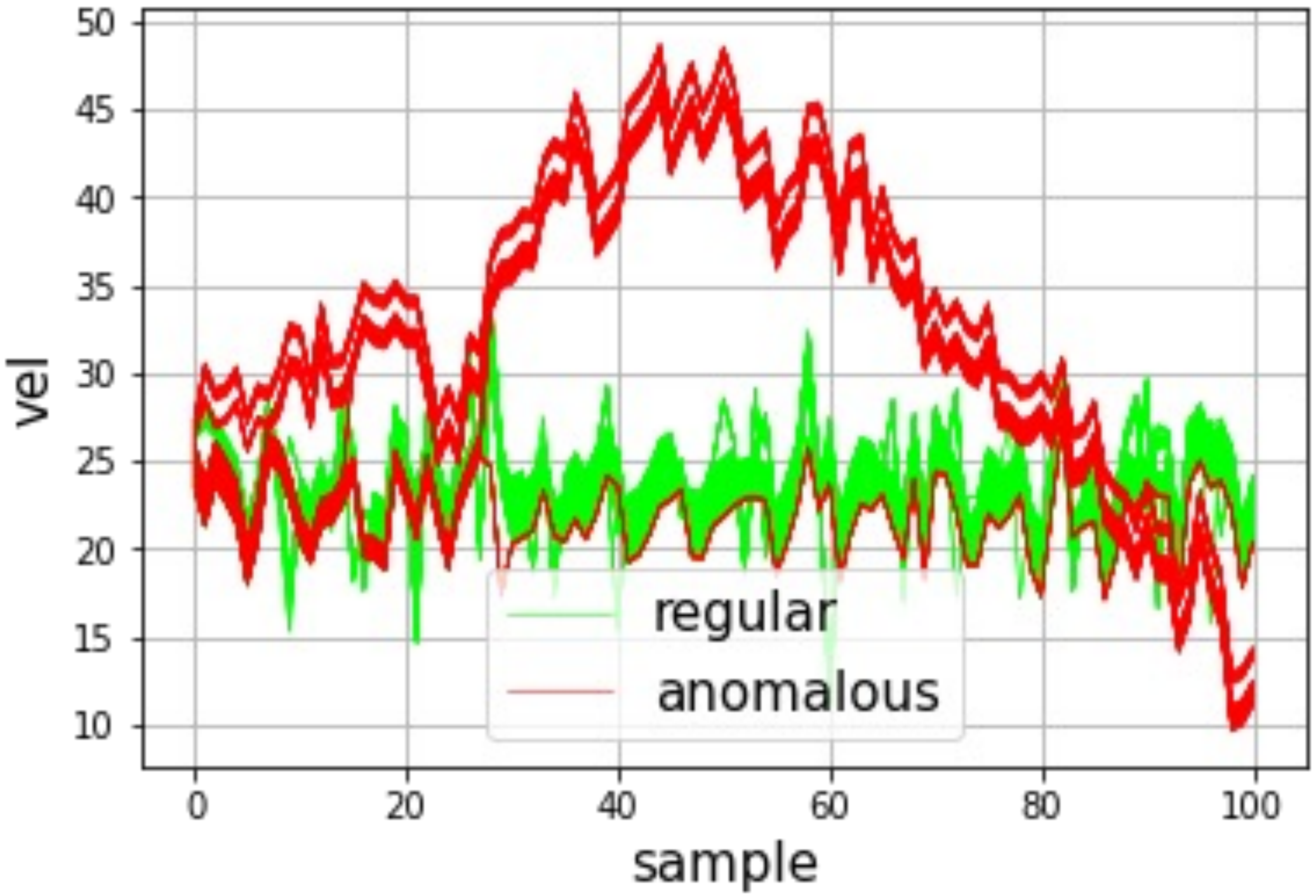
  $gen_\Theta \leftarrow$ SAMPLE$(newg_\Theta \cup gen_\Theta, F)$

end for

return gen_Θ

[L. Nenzi, S. Silvetti, E. Bartocci, L. Bortolussi: A Robust Genetic Algorithm for Learning Temporal Specifications from Data. QEST 2018]

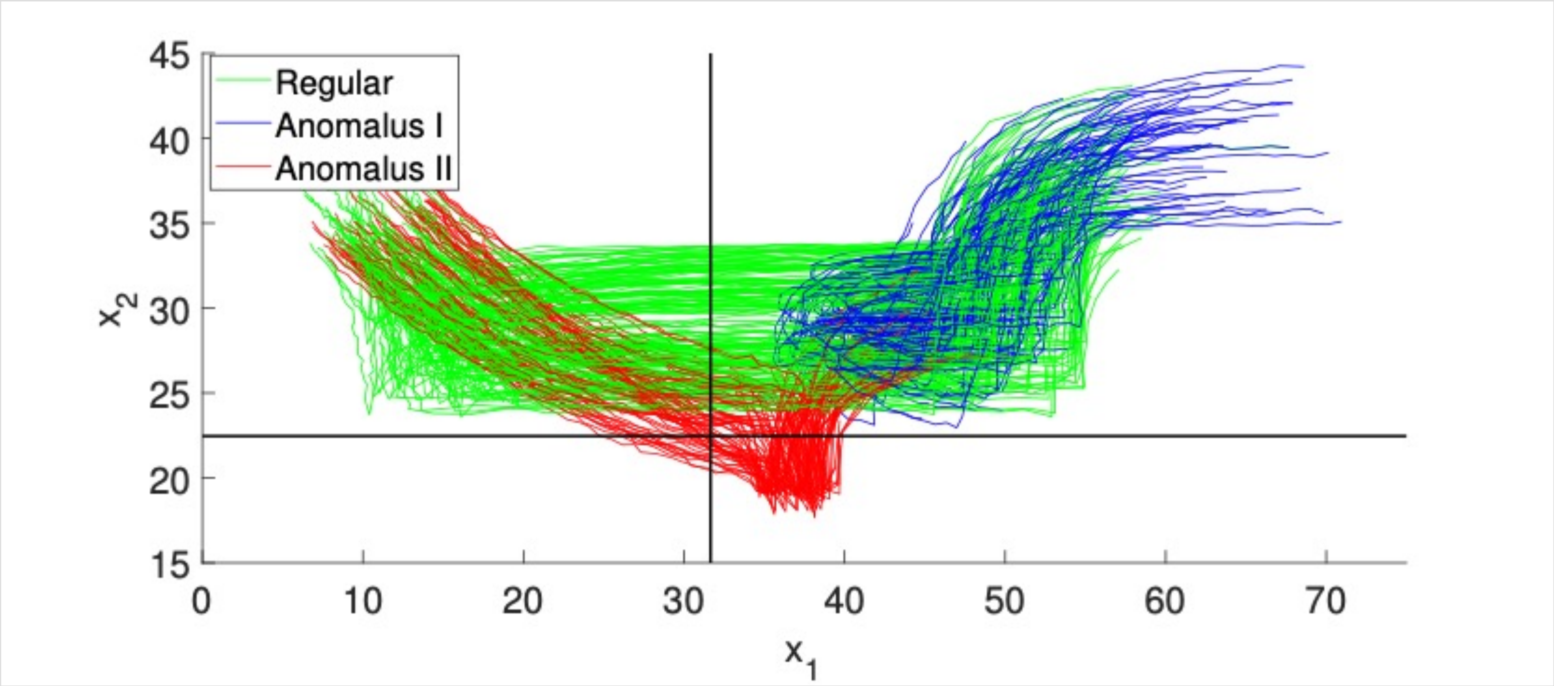
Results: Train Cruise



$$(\mathrm{F}_{[22,40]}(\mathtt{vel} > 24.48)) \wedge (\mathrm{F}_{[46,49]}(19.00 < \mathtt{vel} < 26.44))$$

Results: Maritime Surveillance

Synthetic dataset of naval surveillance of 2-dimensional coordinates traces of vessels behaviours.



$$((x_2 > 22.46)\,\mathcal{U}_{[49,287]}\,(x_1 \leq 31.65))$$

Limitation of ROGE

- Initial population designed ”by hand”
- The learning parameter algorithm can be slow (depending on the size parameter space)

STL Classifier: Context Free Grammar

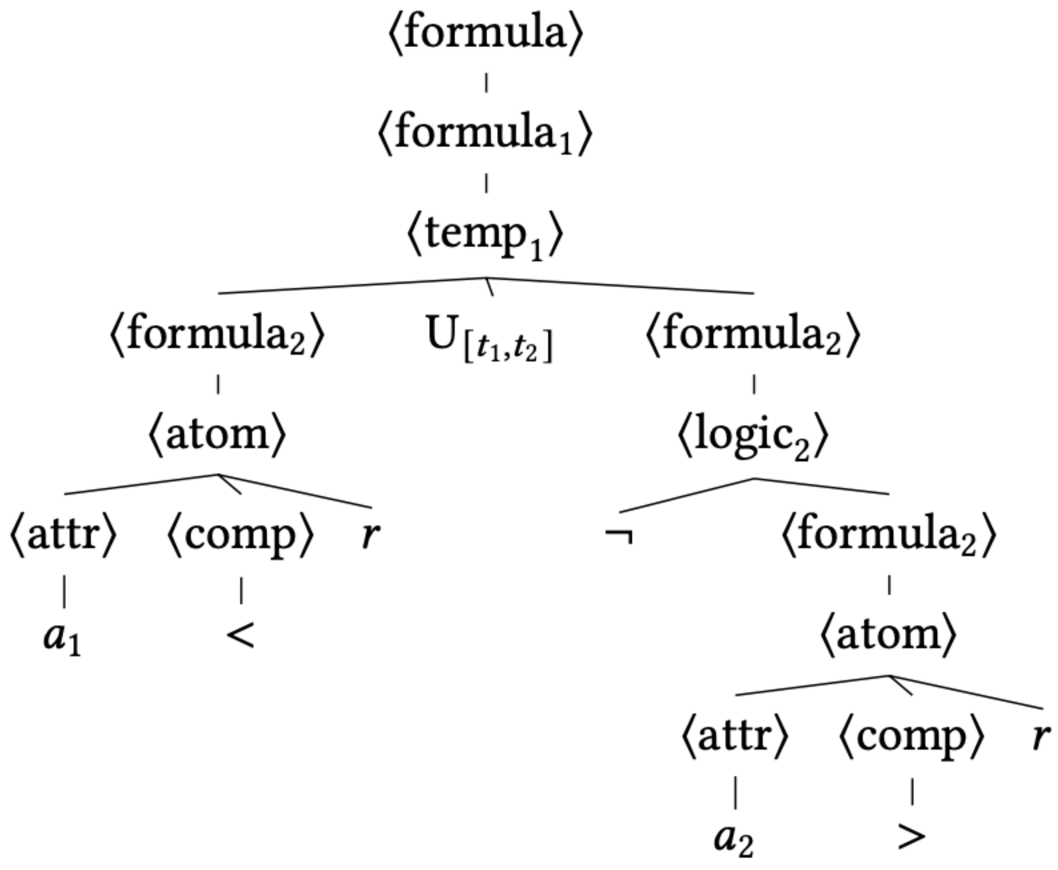
$$
\begin{aligned}
\langle \text{formula} \rangle &::= \langle \text{formula}_1 \rangle \\
\langle \text{formula}_i \rangle &::= \begin{cases} \langle \text{atom} \rangle \mid \langle \text{logic}_i \rangle \mid \langle \text{temp}_i \rangle & \text{if } i < i_{\text{max}} \\ \langle \text{atom} \rangle \mid \langle \text{logic}_i \rangle & \text{otherwise} \end{cases} \\
\langle \text{logic}_i \rangle &::= \neg \langle \text{formula}_i \rangle \mid \langle \text{formula}_i \rangle \wedge \langle \text{formula}_i \rangle \\
\langle \text{temp}_i \rangle &::= \langle \text{formula}_{i+1} \rangle \mathrm{U}_{\langle \text{interval} \rangle} \langle \text{formula}_{i+1} \rangle \mid \\
& \mathrm{G}_{\langle \text{interval} \rangle} \langle \text{formula}_{i+1} \rangle \mid \\
& \mathrm{F}_{\langle \text{interval} \rangle} \langle \text{formula}_{i+1} \rangle \\
\langle \text{interval} \rangle &::= [\langle \text{num} \rangle, \langle \text{num} \rangle] \\
\langle \text{atom} \rangle &::= \langle \text{attr} \rangle \langle \text{comp} \rangle 0.\langle \text{num} \rangle \\
\langle \text{attr} \rangle &::= a_1 \mid a_2 \mid \dots \mid a_{|A|} \\
\langle \text{comp} \rangle &::= \; < \; \mid \; > \\
\langle \text{num} \rangle &::= \langle \text{digit} \rangle \langle \text{digit} \rangle \\
\langle \text{digit} \rangle &::= 0 \mid 1 \mid 2 \mid 3 \mid 4 \mid 5 \mid 6 \mid 7 \mid 8 \mid 9
\end{aligned}
$$

[F. Pigozzi, E. Medvet, L. Nenzi. Mining Road Traffic Rules with Signal Temporal Logic and Grammar-Based Genetic Programming, Applied Sciences, 2022]

[F. Pigozzi, L. Nenzi., E. Medvet, BUSTLE: a Versatile Tool for the Evolutionary Learning of STL Specifications from Data (second revision on Evolutionary Computation]

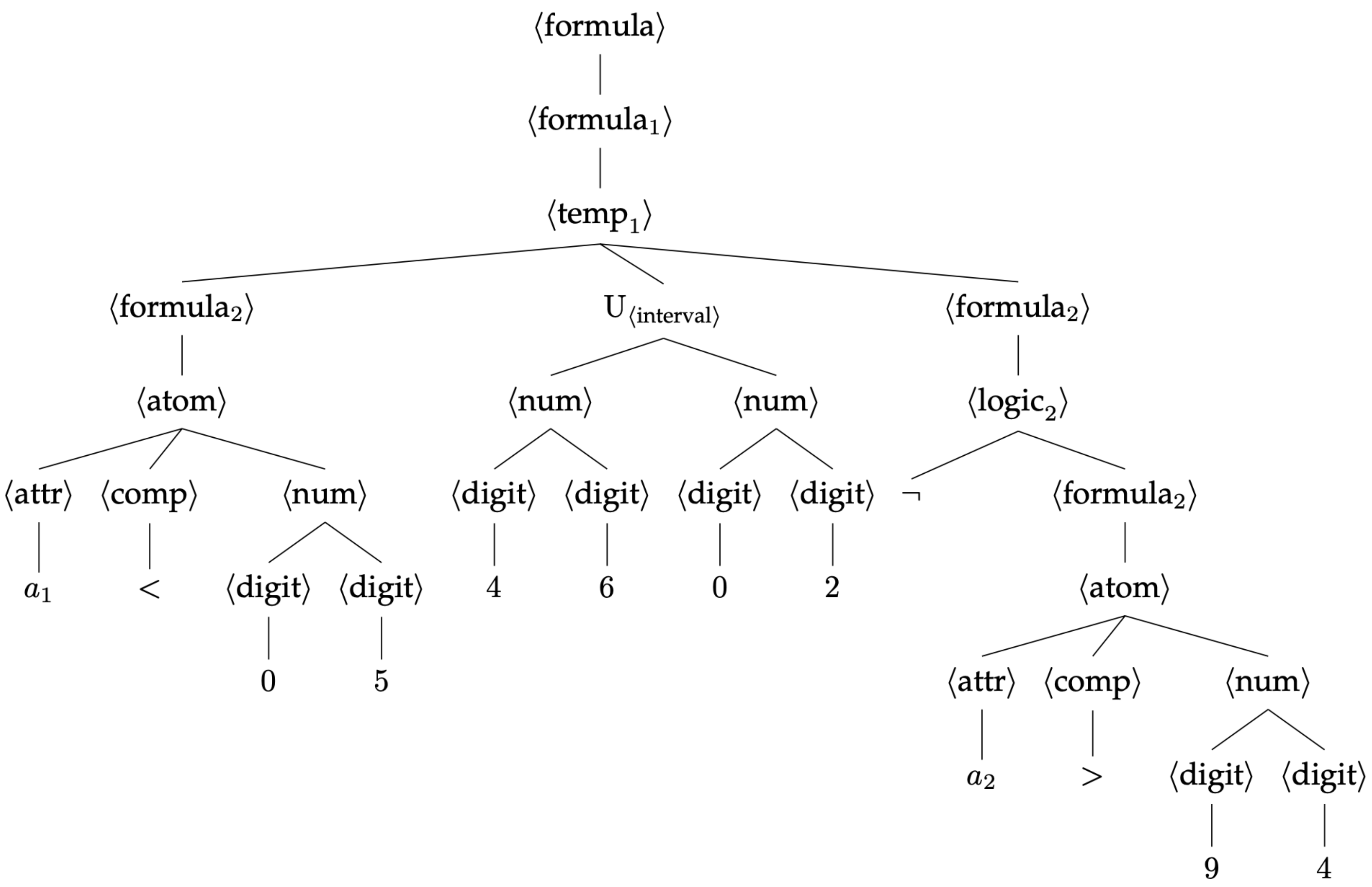
STL classifier: Building the population

- Candidate formulas are represented as derivation trees of a grammar

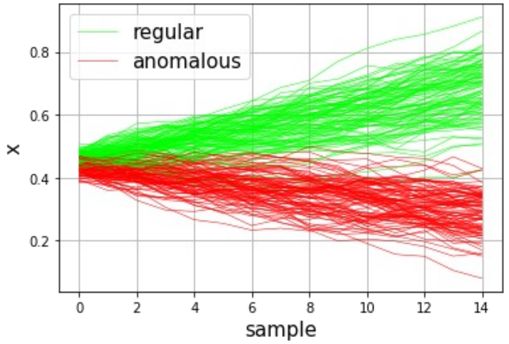


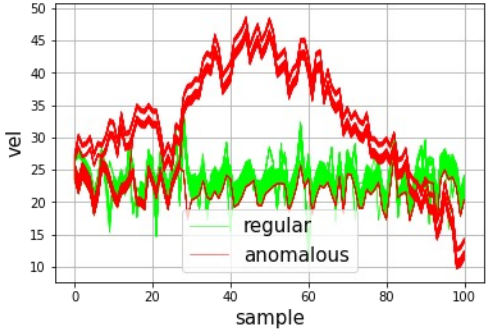
STL classifier: Building the population

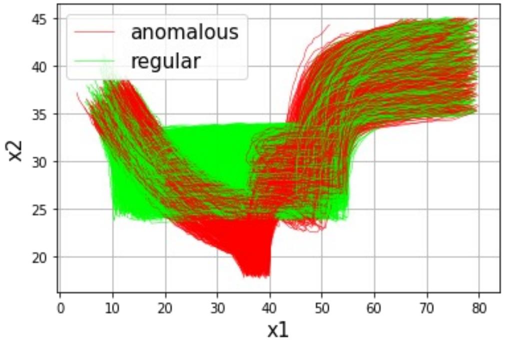
- Candidate formulas are represented as derivation trees of a grammar



Results

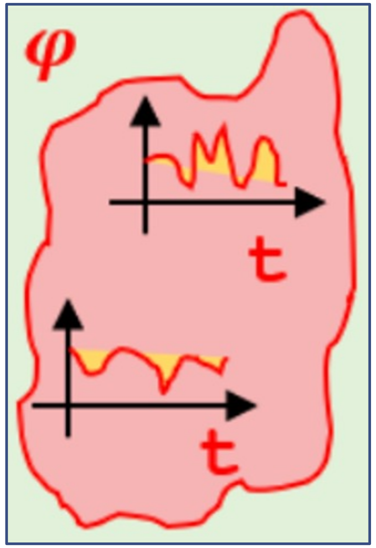






Dataset	Algorithm	FNR	FPR	Acc	Time
Linear	Random	0.20	0.20	0.80	11
	BUSTLE (single-level)	0.00	0.00	1.00	15
	BUSTLE (bi-level)	0.00	0.00	1.00	112
	Nenzi et al. (2018)	0.00	0.00	1.00	113
	Mohammadinejad et al. (2020b)	N/A	N/A	0.98	39
Train	Random	0.55	0.53	0.46	31
	BUSTLE (single-level)	0.03	0.05	0.96	26
	BUSTLE (bi-level)	0.00	0.03	0.98	523
	Nenzi et al. (2018)	0.10	0.00	0.95	576
	Mohammadinejad et al. (2020b)	N/A	N/A	0.98	32
Maritime	Random	0.52	0.50	0.49	84
	BUSTLE (single-level)	0.00	0.00	1.00	109
	BUSTLE (bi-level)	0.00	0.00	1.00	1477
	Nenzi et al. (2018)	0.00	0.00	1.00	1599
	Mohammadinejad et al. (2020b)	0.05	0.02	0.96	73
	Bombara and Belta (2021)	N/A	N/A	0.98	140

STL Classifier: Fitness Function for the one-class problem



Training data set: one set

- regular X^{+}_{learn}

Fitness, two high level requirements:

1. **Tight** formulas should be preferred
2. Formulas that lead to few false anomalies should be preferred

1

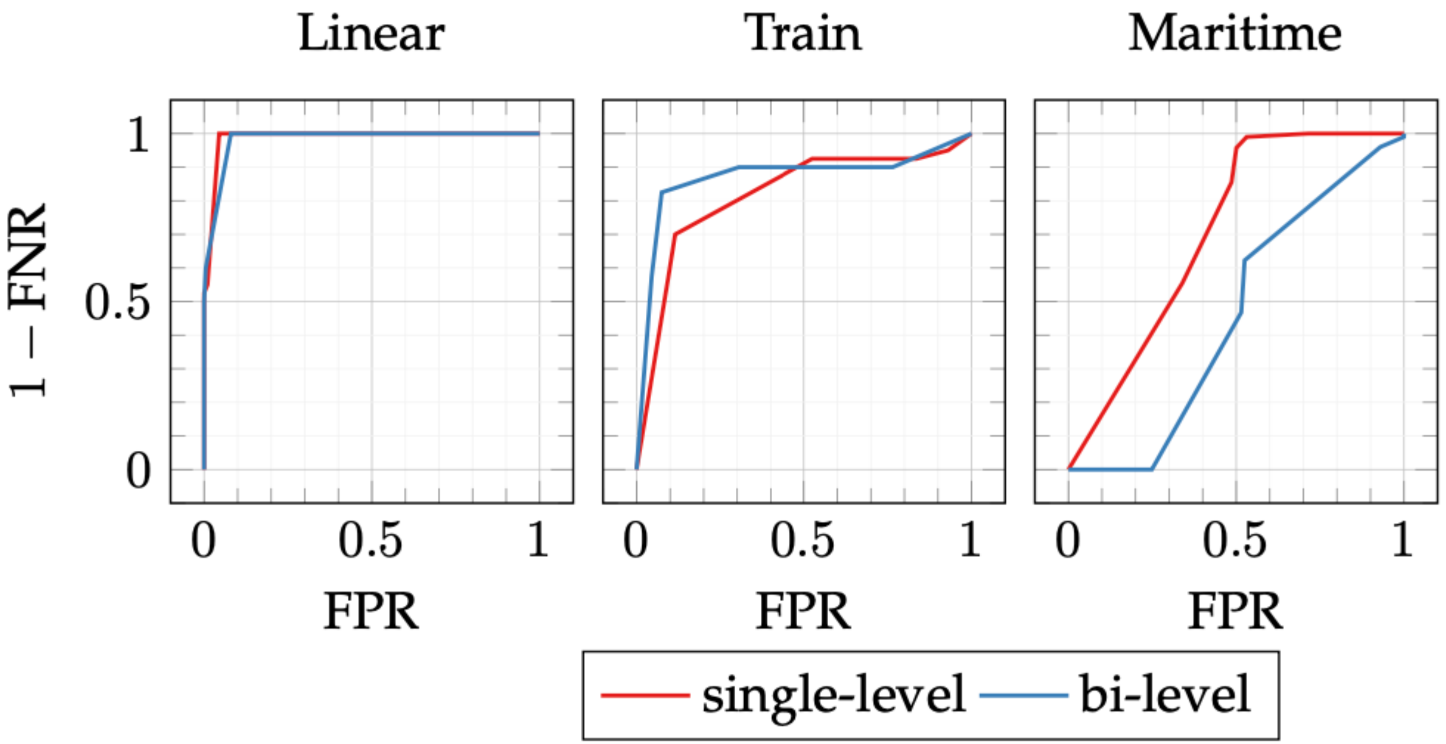
$$f(\varphi; X^{+}_{\text{learn}}) = \alpha \frac{1}{|X^{+}_{\text{learn}}|} \left|\{\boldsymbol{x} \in X^{+}_{\text{learn}} : \boldsymbol{x} \not\models \varphi\}\right| + \frac{1}{\sigma'_{\varphi, X^{+}_{\text{learn}}} |X^{+}_{\text{learn}}|} \sum_{\boldsymbol{x} \in X^{+}_{\text{learn}}} |\rho(\varphi, \boldsymbol{x})|$$

2

Results

	Variant	Two-classes					One-class				
		FNR	FPR	Acc	Time	c	FNR	FPR	Acc	Time	c
Lin.	Random	0.20	0.20	0.80	11	8.0	0.98	0.20	0.41	10	8.0
	BUSTLE (single-l.)	0.00	0.00	1.00	15	9.5	0.45	0.00	0.77	11	11.0
	BUSTLE (bi-l.)	0.00	0.00	1.00	112	12.5	0.40	0.00	0.80	145	11.0
Train	Random	0.55	0.53	0.46	31	8.0	0.81	0.15	0.52	18	8.0
	BUSTLE (single-l.)	0.03	0.05	0.96	26	12.0	0.30	0.12	0.79	25	11.0
	BUSTLE (bi-l.)	0.00	0.03	0.98	523	13.0	0.18	0.08	0.87	438	13.5
Marit.	Random	0.52	0.50	0.49	84	8.0	0.77	0.21	0.51	73	8.0
	BUSTLE (single-l.)	0.00	0.00	1.00	109	9.5	0.15	0.49	0.68	72	9.5
	BUSTLE (bi-l.)	0.00	0.00	1.00	1477	9.0	0.38	0.52	0.55	2008	12.0

Results



Limitations:

- There may be several good classifiers
- Finding the best classifier might be unfeasible
- There may not exist a single, good classifier

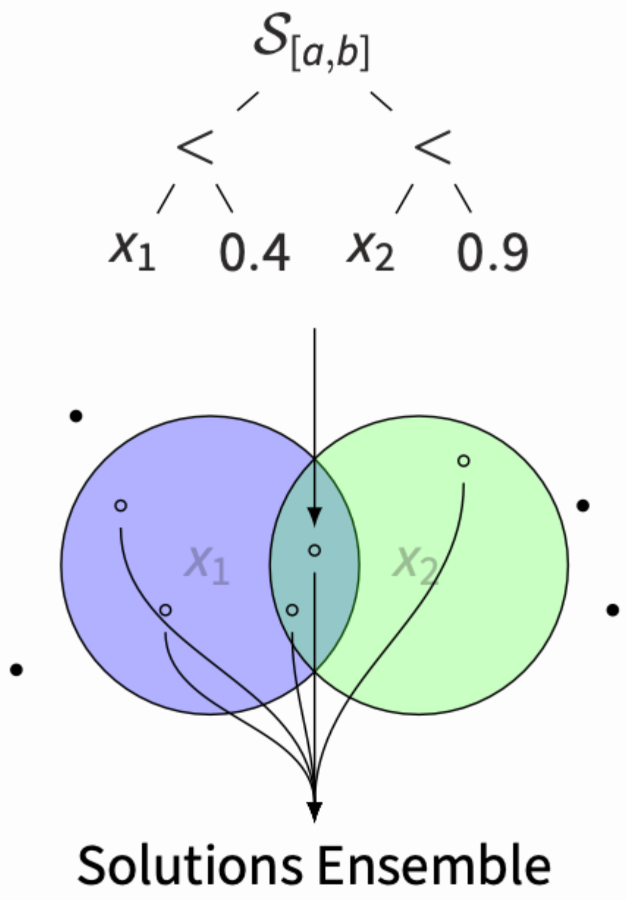
A one-shot algorithm

An evolutionary algorithm that learns an ensemble of solutions in a single run

- Population update:
  - Divide population in groups, one for each variable
  - The fittest formula of each group goes to next generation (elitism)
  - The remaining offspring is obtained reproducing the individuals

- Solutions update. If some individuals solve the problem (f $< \epsilon$), consider their groups:
  - Remove from the population the individuals in these groups (extinction)
  - Add them to the solutions ensemble
  - Refill the population with new individuals (random immigrants)

Stop once n_{target} variables have been solved



[Patrick Indri, Alberto Bartoli, Eric Medvet, Laura Nenzi: One-Shot Learning of Ensembles of Temporal Logic Formulas for Anomaly Detection in Cyber-Physical Systems. EuroGP 2022: 34-50]

An online application

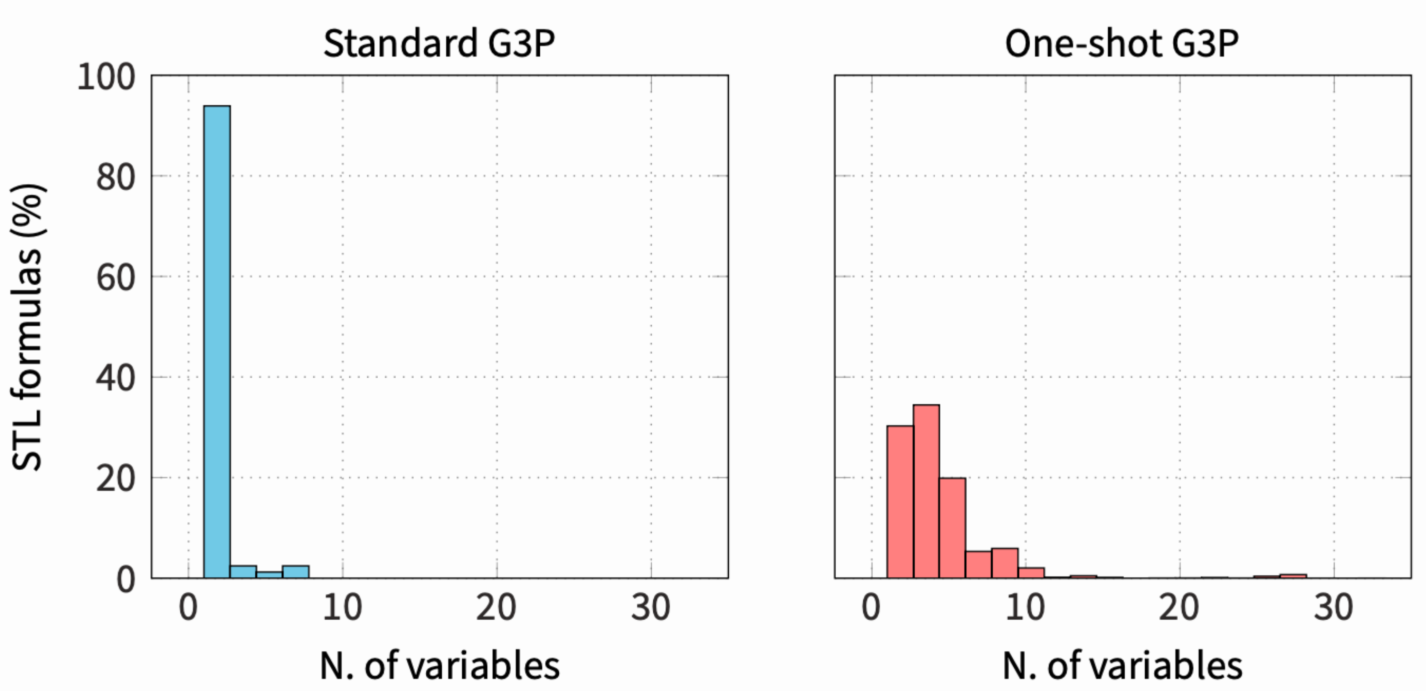
- For “online” anomaly detection
- using Past STL
- a single trajectory x, with several variables (> 50)
- x is divided as $x^{+}_{train}, x^{+}_{test}$, x^{-}_{test}
- Sensor readings are numerical variables, whilst actuator readings are ternary non-ordinal variables

Results

	Multi-run G3P (30 runs)				One-shot G3P ($n_{target} = 20$)			
Dataset	TPR	FPR	AUC	f_{evals}	TPR	FPR	AUC	f_{evals}
SWaT	0.6648	0.0005	0.8321	43 243	0.6571	0.0007	**0.8401**	**11 767**
N-BaIoT-1	0.9981	0.0000	**0.9990**	47 152	0.8952	0.0011	0.9475	**3297**
N-BaIoT-2	0.9996	0.0016	0.9989	355 696	1.0000	0.0422	**0.9998**	**5732**
N-BaIoT-3	0.9949	0.0000	**0.9974**	51 979	0.9596	0.0076	0.9739	**5965**
N-BaIoT-4	0.0000	0.0002	0.4998	298 158	0.9272	0.0025	**0.9632**	**35 811**
N-BaIoT-5	0.6152	0.0012	0.8073	156 033	0.7492	0.0010	**0.8742**	**7898**
N-BaIoT-6	0.7192	0.0011	**0.8594**	371 358	0.6807	0.0023	0.8387	**12 235**
N-BaIoT-7	0.7070	0.0000	0.8534	269 708	0.6896	0.0009	**0.9072**	**16 736**
N-BaIoT-8	0.0000	0.0000	0.5000	1 015 286	0.4166	0.0027	**0.7050**	**88 921**
N-BaIoT-9	0.7812	0.0005	**0.8905**	260 259	0.7440	0.0011	0.8702	**13 696**

Results

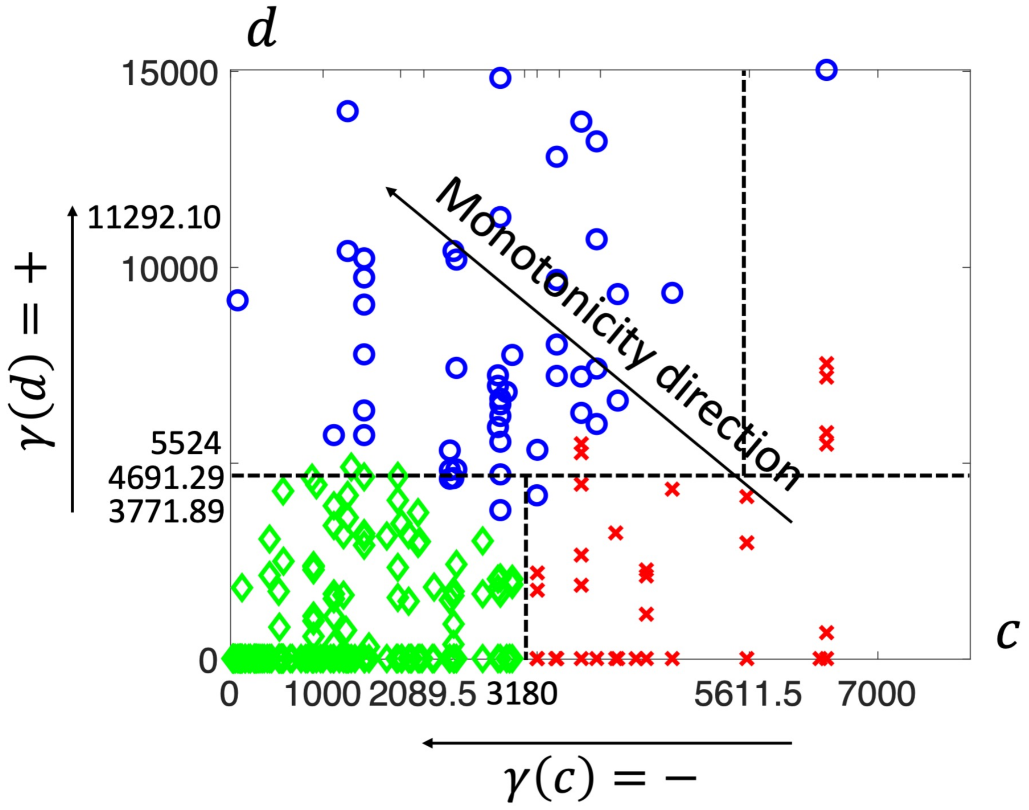
- Standard GP more than 60 % of the formulas containing a single variable.
- The one-shot algorithm produces a larger percentage of solutions with more variables, with some STL formulas containing more than 20 variables

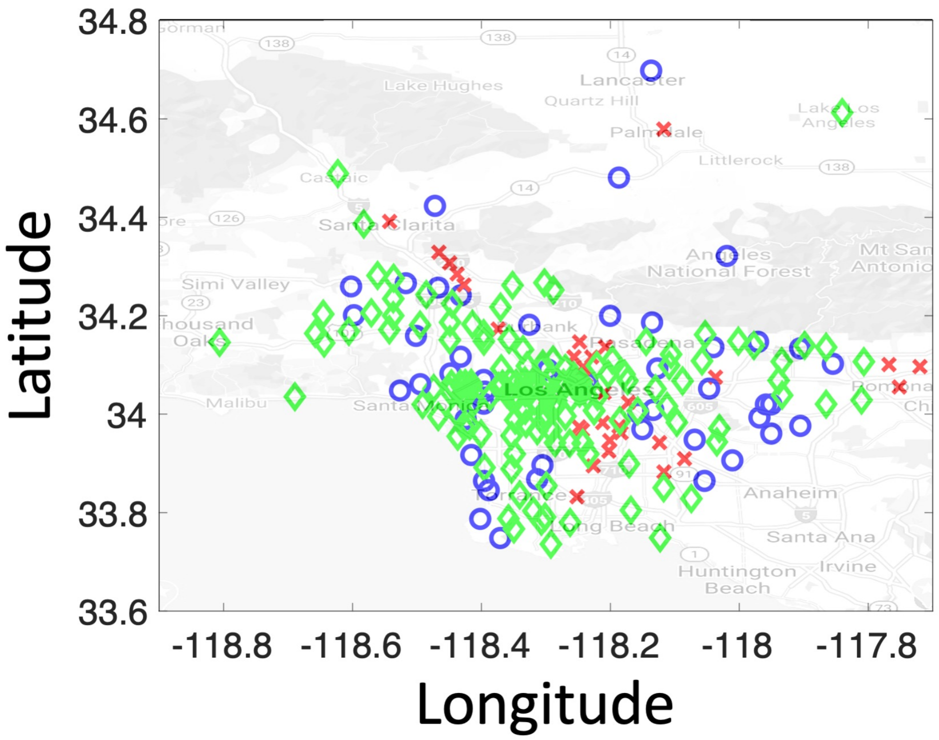


Comparison with classical ML: it is

- competitive on SWaT
- it compares unfavourably on N-BaIoT, where it reaches a perfect detection rate only on N-BaIoT-2. However on N-BaIoT at least one anomalous instant for each attack is correctly identified, and all attacks might thus be considered as identified.

Learning STL-based clustering (Unsupervised Learning)





Goal: clusterizing spatio-temporal data using formal logic

[Mohammadinejad et al, Mining Interpretable Spatio-temporal Logic Properties for Spatially Distributed Systems, ATVA, 2021]

Monotonic PSTREL $\varphi(p)$:

- The polarity of a parameter p is:
  - $+$ if it is easier to satisfy φ as we increase the value of p
  - $-$ if it is easier to satisfy φ as we decrease the value of p

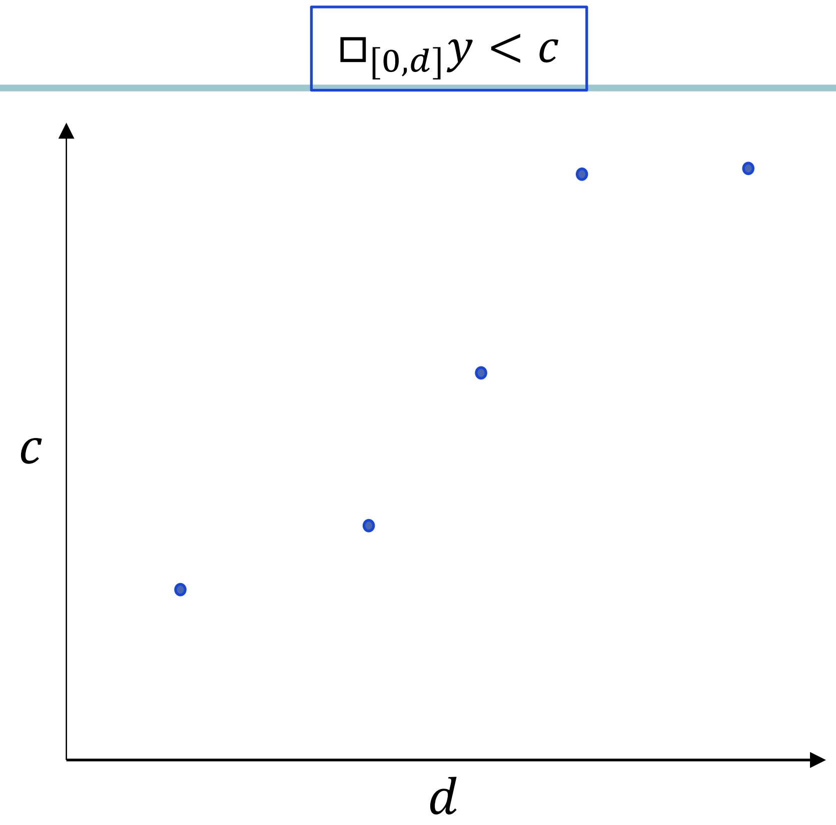
- Monotonic PSTREL:
  - All parameters have either $+$ or $-$ polarity

- Example: $\Box_{[0,d]}\varphi$
  - Polarity of d is $-$

Validity Domain of PSTREL $\varphi(p)$

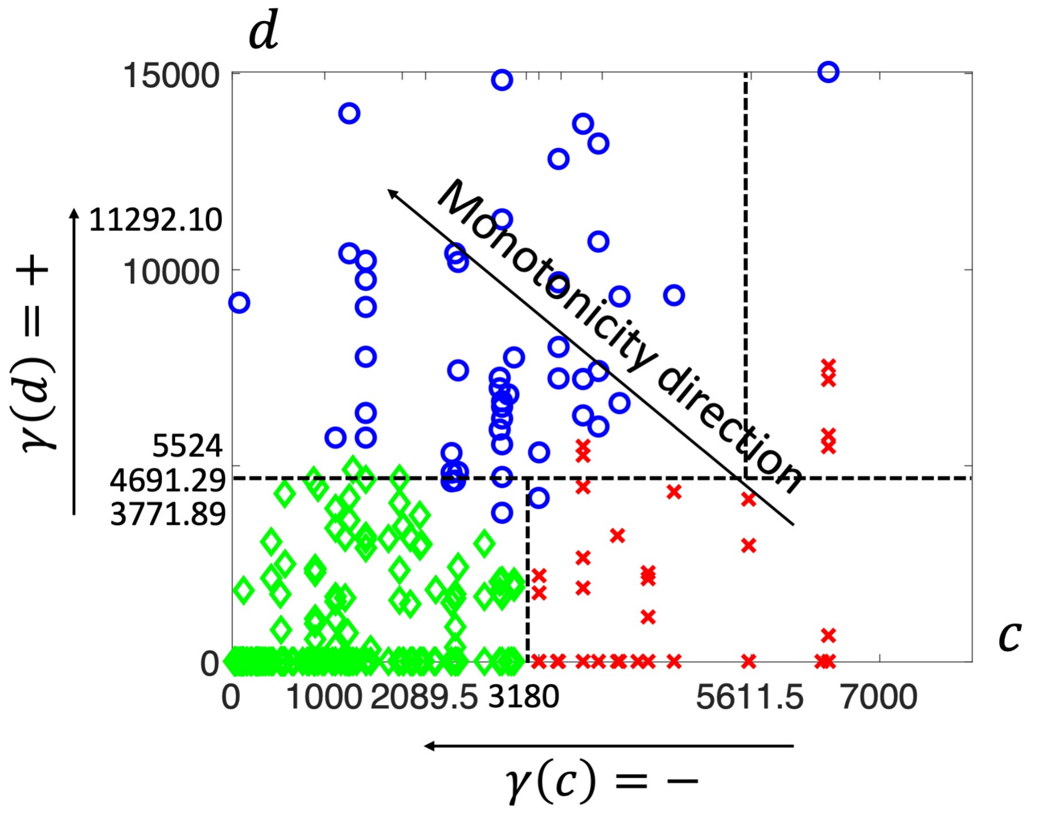
- Given a location l
- A set of spatio-temporal traces X associated with l
- The set of all valuations to p such that each trace in X satisfies the STREL formula
- Boundary of the validity domain:

  The robustness value with respect to at least one trace in X is ≈ 0



High-level steps

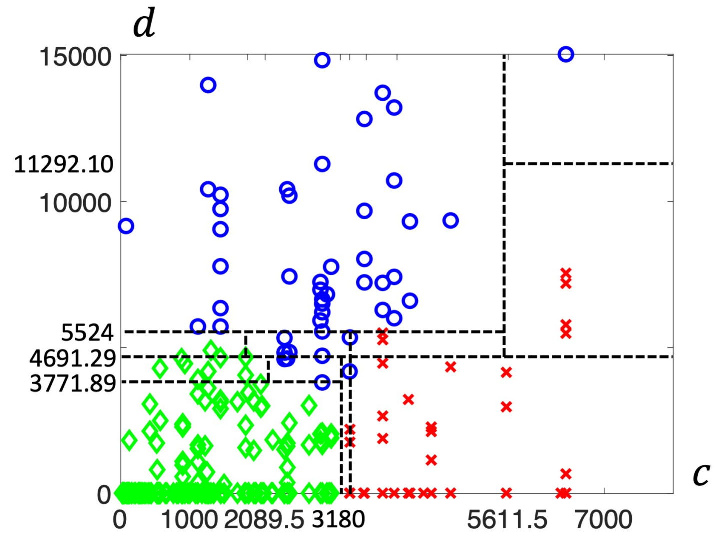
- Constructing the spatial model
- Projecting each spatio-temporal trace to a tight valuation in the parameter space of a given PSTREL formula
- Clustering the trace projections throught AHC
- Learning bounding boxes for each cluster using a Decision Tree based approach
- Learning a STREL formula for each cluster
- Improving the interpretability of the learned STREL formulas

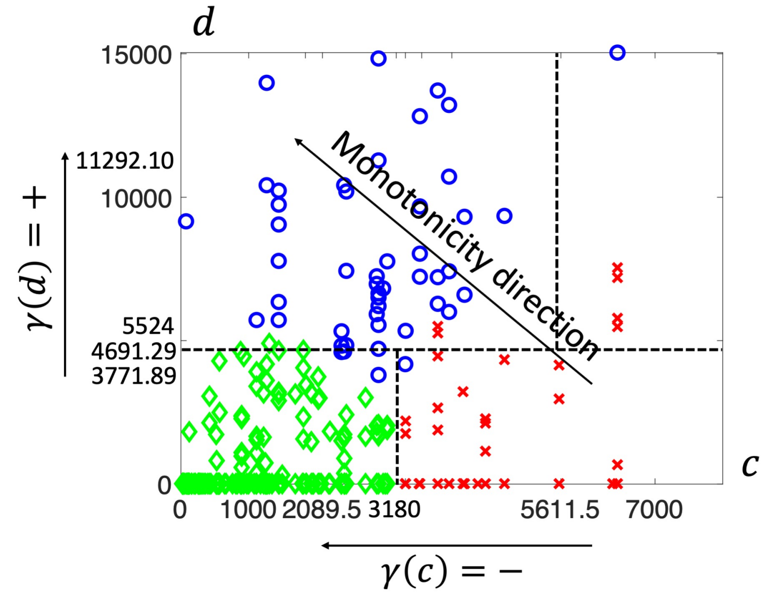


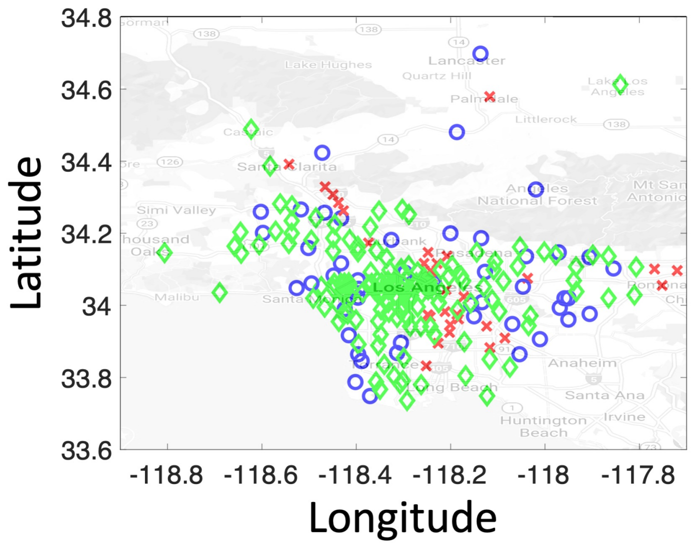
COVID-19 data from LA County

PSTREL formula: $\Diamond_{[0,d]}\{F_{[0,\tau]}(x > c)\}$

- We fix τ to 10 days
- Small d and large c are **hot spots**





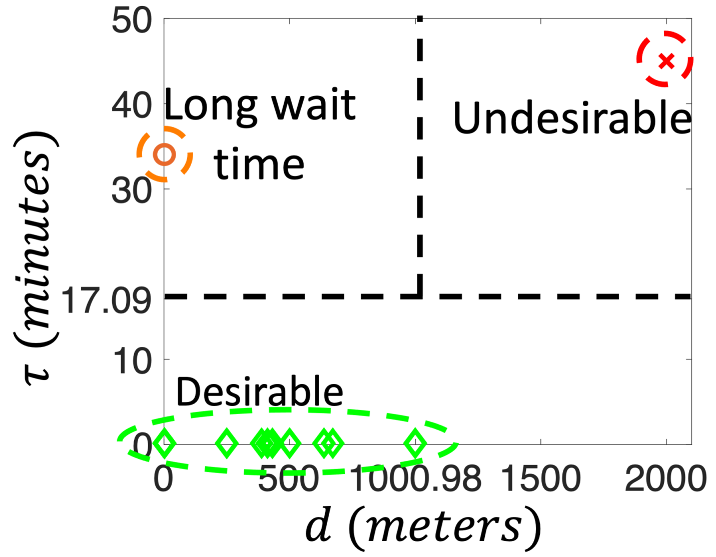


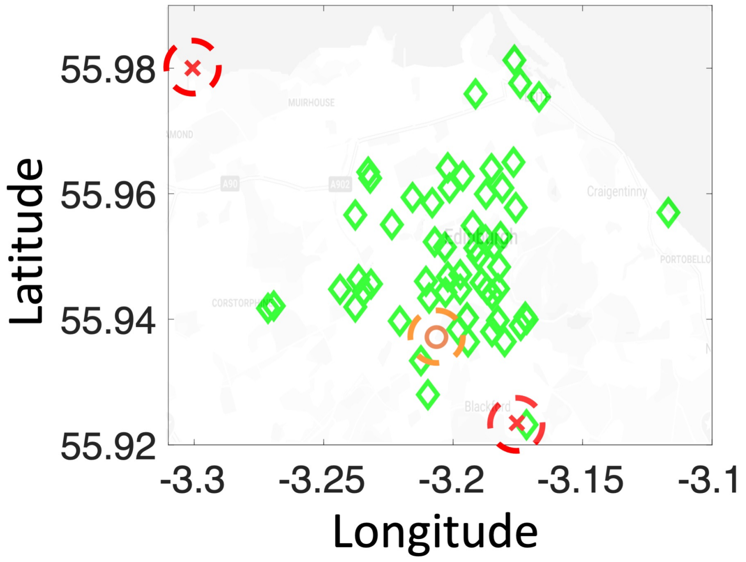
$$\varphi_{red} = \Diamond_{[0,4691.29]}\{F_{[0,10]}(x \geq 3181)\} \vee \Diamond_{[0,15000]}\{F_{[0,10]}(x \geq 5612)\}$$

BSS data from the city of Edinburgh

PSTREL formula: $\varphi(\tau, d) = G_{[0,3]}\big(\varphi_{wait}(\tau) \vee \varphi_{walk}(d)\big)$

$\varphi_{wait}(\tau) = F_{[0,\tau]}(B \geq 1) \wedge F_{[0,\tau]}(S \geq 1),$
$\varphi_{walk}(d) = \diamond_{[0,d]}(B \geq 1) \wedge \diamond_{[0,d]}(S \geq 1)$



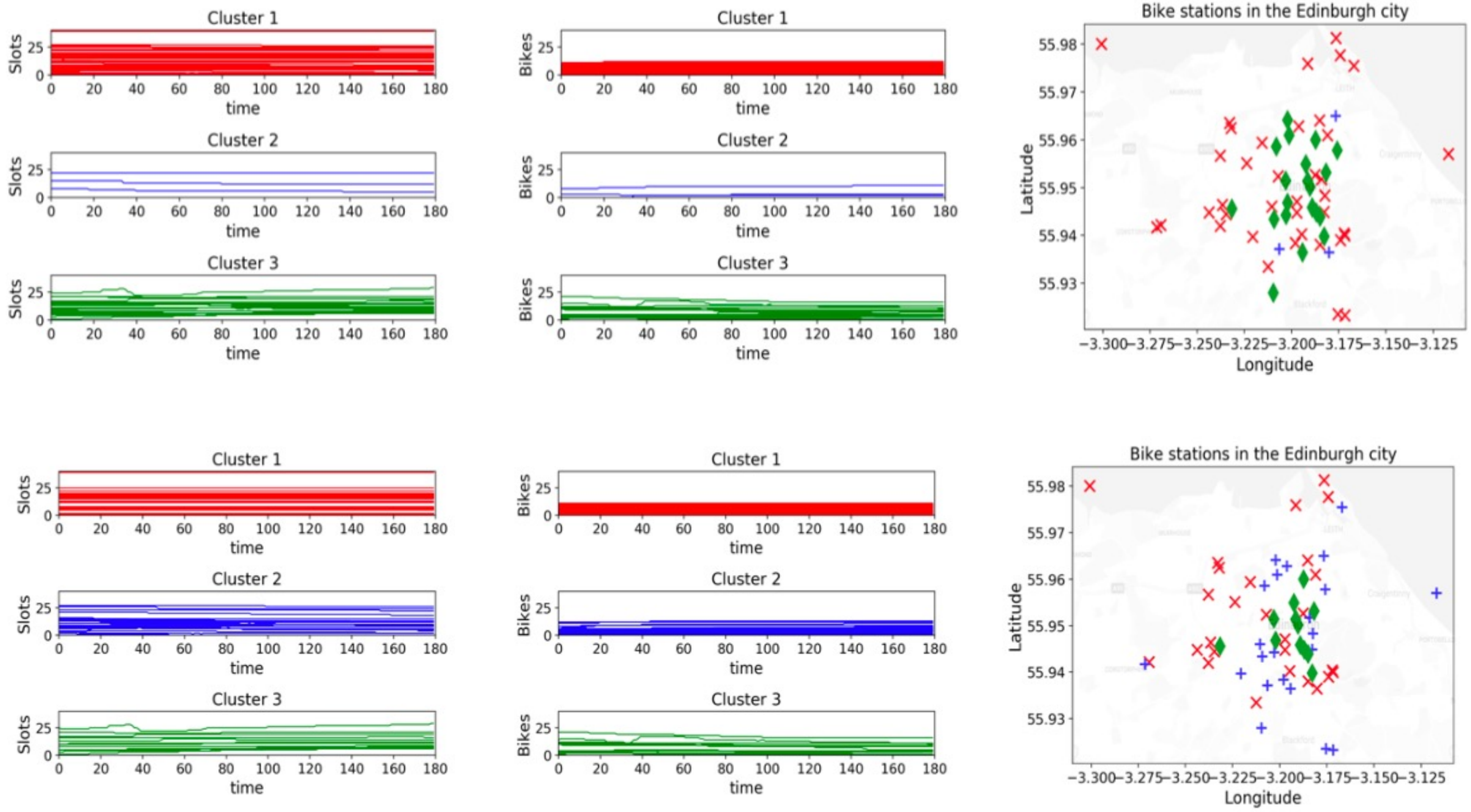


$\varphi_{red} = \neg G_{[0,3]}\big(\varphi_{wait}(17.09) \vee \varphi_{walk}(2100)\big) \wedge \neg G_{[0,3]}(\varphi_{wait}(50) \vee \varphi_{walk}(1000.98))$

Traditional ML approaches

Kshape approach from tslearn library

KMeans approach from tslearn library



(a) Clusters learned from BSS data

(b) Clusters learned from BSS data

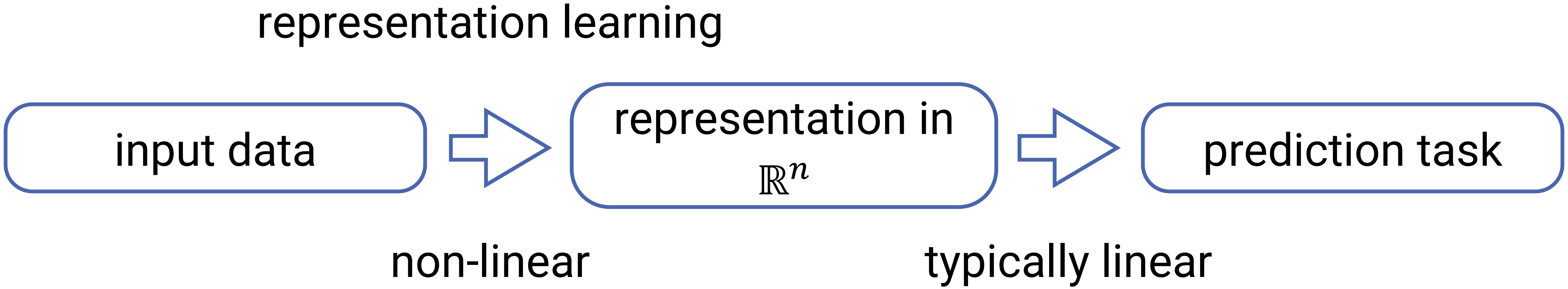
(c) Regions in Edinburgh city associated with the learned clusters.

Dessert

Related Works

- Bartocci et all: Survey on mining signal temporal logic specifications. Inf. Comput., 2022
- Template-Free:
  - Bombara, G et all, A Decision Tree Approach to Data Classification Using Signal Temporal Logic. In: Proc. of HSCC, 2016
  - Bombara, G. and Belta, C. (2021). Offline and Online Learning of Signal Temporal Logic Formulae Using Decision Trees.
  - Mohammadinejad, S., Deshmukh, J. V., Puranic, A. G., Vazquez-Chanlatte, M., and Donze ́, A. (2020b). Interpretable classification of time-series data using efficient enumerative techniques. Proceedings of the 23rd International Conference on Hybrid Systems: Computation and Control.
  - Andrea Brunello, Dario Della Monica, Angelo Montanari, Nicola Saccomanno, Andrea Urgolo: Monitors That Learn From Failures: Pairing STL and Genetic Programming. IEEE Access 11:
- Only-positive Example:
  - S. Jha, A. Tiwari, S. A. Seshia, T. Sahai, N. Shankar. TeLEx: learning signal temporal logic from positive examples using tightness metric, Formal Methods in System Design
- Clustering
  - Marcell Vazquez-Chanlatte, Jyotirmoy V. Deshmukh, Xiaoqing Jin, Sanjit A. Seshia: Logical Clustering and Learning for Time-Series Data. CAV (1) 2017: 305-325
- Exploiting Monotonicity
  - Marcell Vazquez-Chanlatte, Shromona Ghosh, Jyotirmoy V. Deshmukh, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia: Time-Series Learning Using Monotonic Logical Properties. RV 2018: 389-405

The heavy cake: Can we learn formulae in a continuous space?

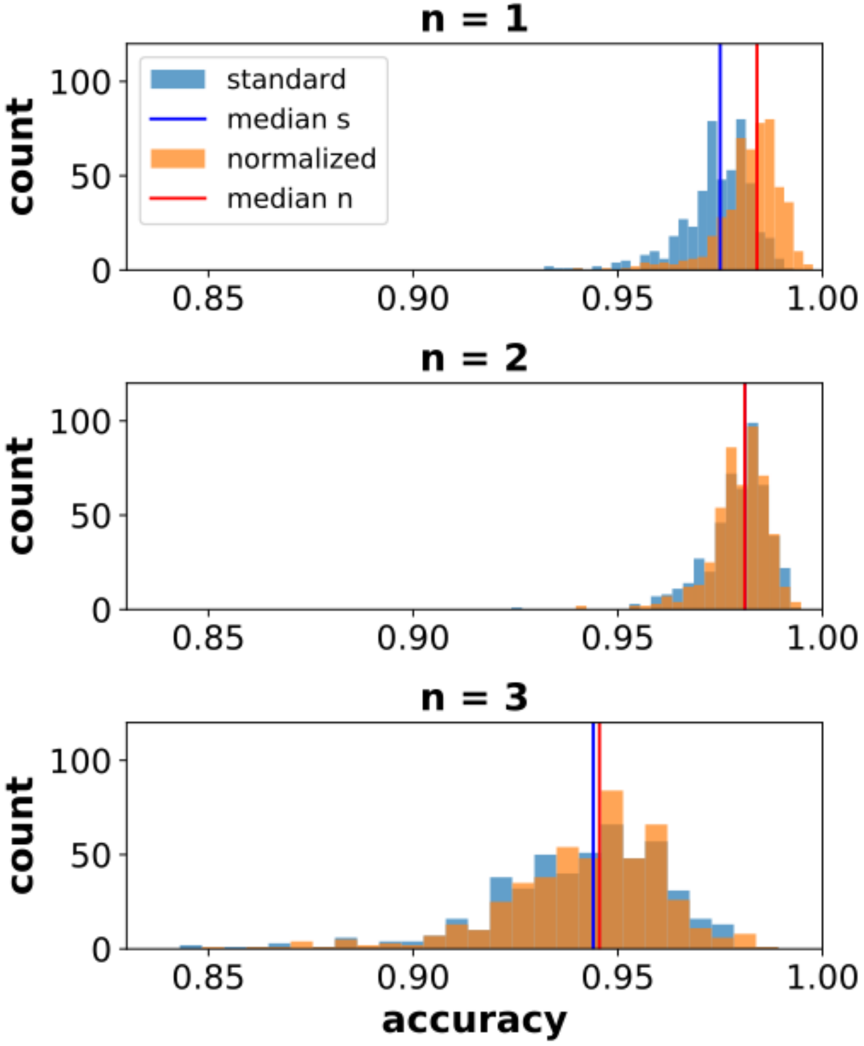


Main Idea: define an embedding of STL formulae in continuous space implicitly by defining a kernel for STL (semantic embedding)

Very brief overview

- Using **kernels-based method** we can construct an embeddings
- STL **kernels regression**: given $p(\psi_j | M)$ for randomly chosen formulae $\psi_1, \dots, \psi_n,$ we predict $p(\varphi | M)$ without knowing or executing the system M
- Use **Kernel PCA** to reduce the dimensionality of the embedded space
- **Inverting the embedding**: learn invertible encodings using Graph Neural Networks (GNN)
- Combine syntax and semantic based embeddings to get invertible mappings from formulae to real vector spaces and use the framework for STL requirement mining

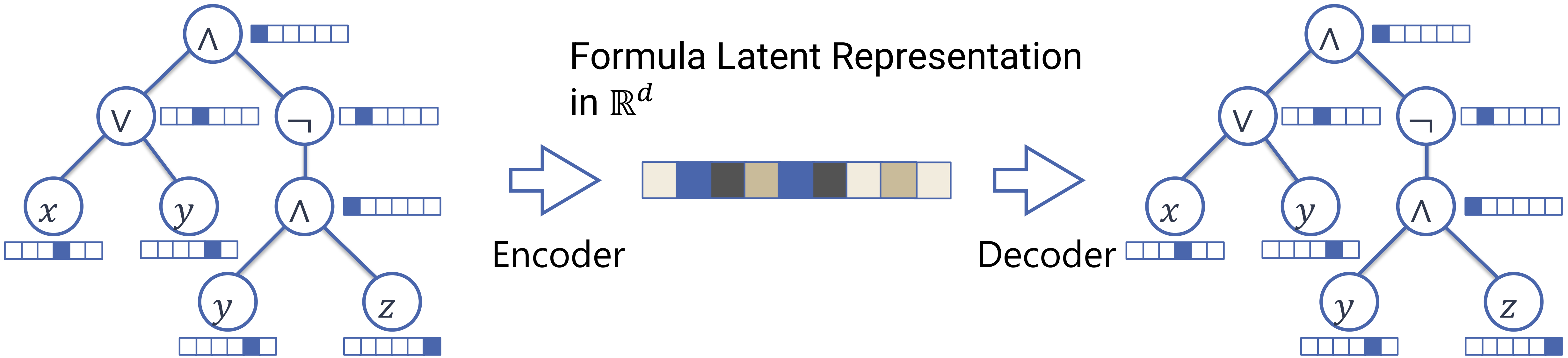
Accuracy of satisfiability prediction



[Bortolussi, L., Gallo, G. M., Křetínský, J., & Nenzi, L. Learning model checking and the kernel trick for signal temporal logic on stochastic processes. In: TACAS, 2022]

Inverting the embedding

Problem with kernel embeddings: non-invertibility → **encoding-decoding architecture**



Learn invertible encodings using Graph Neural Networks (GNN):

- Encode parse tree of the formula into the latent space
- Decode latent vectors to syntactic trees, ideally with the same semantic meaning of the input formula

Thank you for listening!

Robustness function $\rho(\varphi, \mathbf{x}, t)$

$$
\begin{aligned}
\rho(\top, \mathbf{x}, t) &= +\infty \\
\rho(\mu, \mathbf{x}, t) &= y(\mathbf{x}(t)) \textit{ where } \mu \equiv y(\mathbf{x}(t)) \geq 0 \\
\rho(\neg\varphi, \mathbf{x}, t) &= -\rho(\varphi, \mathbf{x}, t) \\
\rho(\varphi_1 \wedge \varphi_2, \mathbf{x}, t) &= \min(\rho(\varphi_1, \mathbf{x}, t), \rho(\varphi_2, \mathbf{x}, t)) \\
\rho(\varphi_1 \, \mathcal{U}_{[a,b]} \varphi_2, \mathbf{x}, t) &= \sup_{t' \in t+[a,b]} (\min(\rho(\varphi_2, \mathbf{x}, t'), \inf_{t'' \in [t,t')} (\rho(\varphi_1, \mathbf{x}, t''))))
\end{aligned}
$$

Learning the Parameters

Problem

Given a PSTL formula φ, a parameter space K, find Θ^* that maximises the discrimination function $f_{opt}(\varphi_\Theta)$

Methodology

1. Sample $\{(\theta_{(i)}, y_{(i)}), i = 1,\ldots,n\}$
2. Emulate (**GP Regression**): $G[R_\varphi] \sim GP(\mu,k)$
3. Optimize the emulation via **GP-UCB algorithm**, new $\theta_{(n+1)}$

ACC, FPR and FNR

$$\mathrm{Acc}(\hat{\varphi}; X_{\text{test}}^{+}, X_{\text{test}}^{-}; \epsilon) = \frac{\left|\{\boldsymbol{x} \in X_{\text{test}}^{+} : \rho(\hat{\varphi}, \boldsymbol{x}) > \epsilon\}\right|}{\left|X_{\text{test}}^{+}\right| + \left|X_{\text{test}}^{-}\right|} + \frac{\left|\{\boldsymbol{x} \in X_{\text{test}}^{-} : \rho(\hat{\varphi}, \boldsymbol{x}) \leq \epsilon\}\right|}{\left|X_{\text{test}}^{+}\right| + \left|X_{\text{test}}^{-}\right|}$$

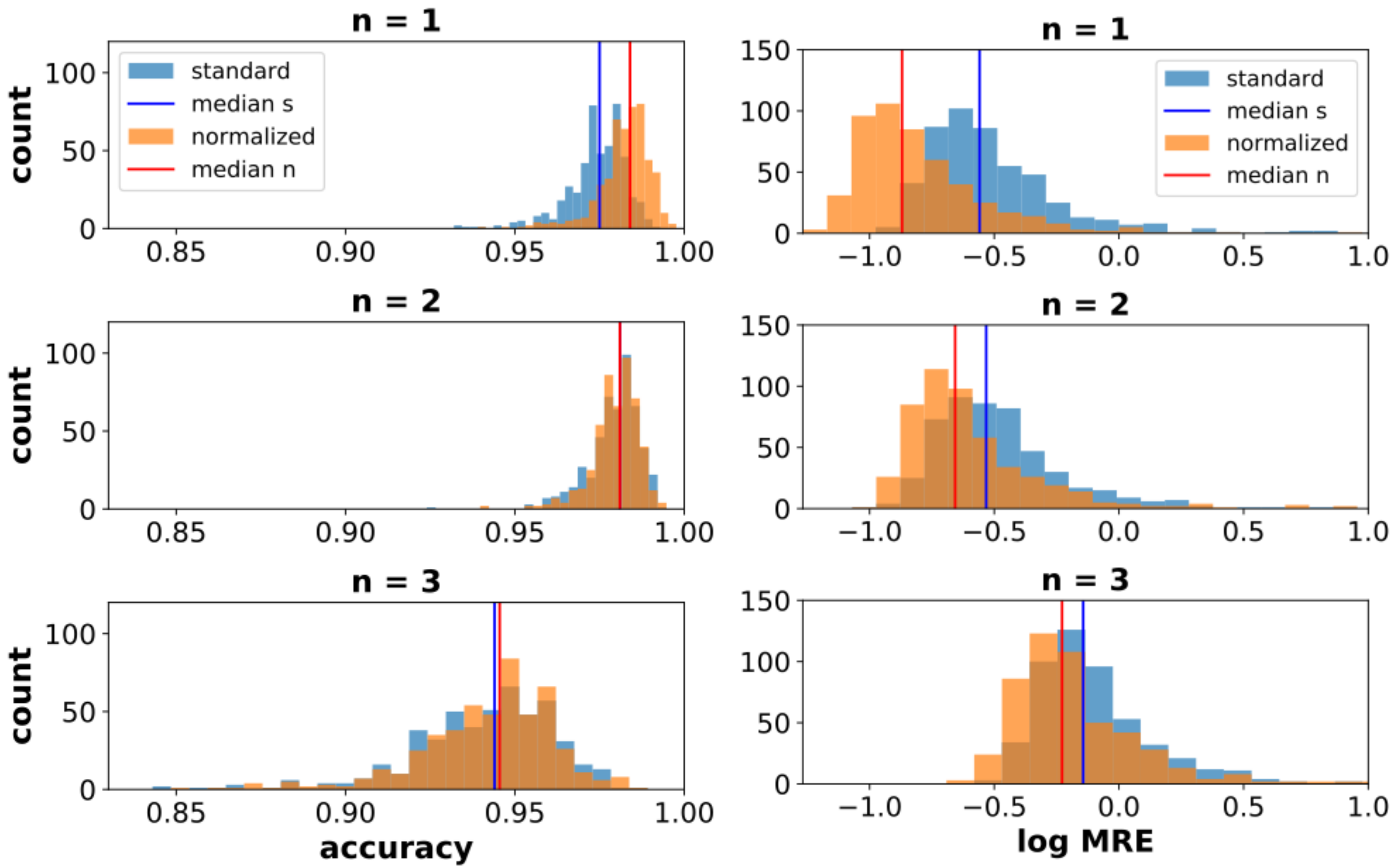
$$\mathrm{FPR}(\hat{\varphi}; X_{\text{test}}^{-}; \epsilon) = \frac{\left|\{\boldsymbol{x} \in X_{\text{test}}^{-} : \rho(\hat{\varphi}, \boldsymbol{x}) > \epsilon\}\right|}{\left|X_{\text{test}}^{-}\right|}$$

$$\mathrm{FNR}(\hat{\varphi}; X_{\text{test}}^{+}; \epsilon) = \frac{\left|\{\boldsymbol{x} \in X_{\text{test}}^{+} : \rho(\hat{\varphi}, \boldsymbol{x}) \leq \epsilon\}\right|}{\left|X_{\text{test}}^{+}\right|}$$

Results summary:

Case	$\|L\|$	$\|W\|$	$runtime(secs)$	numC	$\|\varphi_{cluster}\|$
COVID-19	235	427	813.65	3	$3.\|\varphi\| + 4$
BSS	61	91	681.78	3	$2.\|\varphi\| + 4$
Air Quality	107	60	136.02	8	$5.\|\varphi\| + 7$
Food Court	20	35	78.24	8	$3.\|\varphi\| + 4$

Experimental Results on the stochastic models



(left) Accuracy of satisfiability prediction and (right) MRE of robustness prediction

Immigration (1d)
Isomerization (2d)
Transcription (3d)