UNIVERSITÄT ZU LÜBECK
INSTITUTE FOR SOFTWARE ENGINEERING
AND PROGRAMMING LANGUAGES

isp

Machine Learning for Stream-based Diagnosis

Raik Hipler Martin Leucker

University of Lübeck, Lübeck, Germany

AIPV 2026

May 18–19, 2026

Setting

- Safety-critical systems must stay resilient during runtime.
- Faults must be detected and localized automatically.
- Runtime Verification for fault detection.
- Model-Based Diagnosis for fault localization.

Our work combines Runtime Verification and Model-Based Diagnosis in a unified stream-based framework using stream specification language LOLA.

Motivation

Bottleneck in diagnosis:

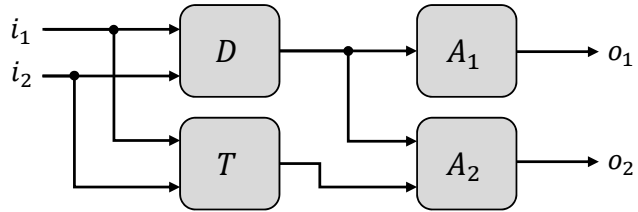
- How to precisely model the expected system behavior?
- Especially difficult for continuous data in cyber-physical systems.
- Machine Learning can bridge this gap by learning symbolic representations of normal behavior.

Goal: Integrate Machine Learning into our LOLA framework to improve applicability.

Model-Based Diagnosis

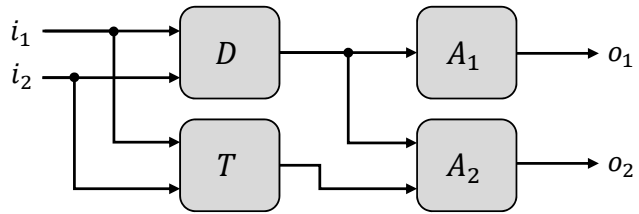
- Technique for localizing faults in compositional systems.
- Components can be normal or abnormal.
- Normal behaviors given by a *system description*.
- Task: Given some observation, identify components that, when assumed to be abnormal and others normal, explain observation.

Model-based Diagnosis



- Inputs: temperature sensors i_1, i_2.
- Outputs: alarms o_1, o_2.
- Components $\text{COMPS} = \{D, T, A_1, A_2\}$.
- System description $\text{SD} = \{$
  $\neg\text{AB}(D) \rightarrow out_D = |i_1 - i_2|,$
  $\neg\text{AB}(T) \rightarrow out_T = (i_1 + i_2)/2,$
  $\neg\text{AB}(A_1) \rightarrow o_1 = (out_D \geq 1),$
  $\neg\text{AB}(A_2) \rightarrow o_2 = (out_T - out_D \geq 5)$
  $\}$

Model-based Diagnosis



Given system description SD and observation OBS:

- Diagnosis is subset $\Delta \subseteq \text{COMPS}$, such that

$$\text{SD} \cup \text{OBS} \cup \{\text{AB}(c) \mid c \in \Delta\} \cup \{\neg\text{AB}(c) \mid c \in \text{COMPS} \setminus \Delta\} \text{ is consistent.}$$

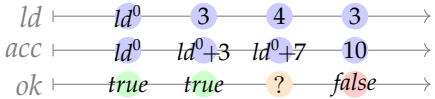
- Here, e.g., $\text{OBS} = \{i_1 = 5, i_2 = 7, o_1 = \mathit{false}, o_2 = \mathit{true}\}$ leading to diagnoses: $\Delta_1 = \{D\}, \quad \Delta_2 = \{T, A_1\}, \quad \Delta_3 = \{A_1, A_2\}$.

LOLA

- Synchronous stream specification language used in Runtime Verification.
- Specification maps input streams to defined streams.

Example:

```
in ld : ℝ
def acc := acc[−1|0] + ld − ld[−3|0]
def ok := acc ≤ 8
ASSUMPTION : 0 ≤ ld ≤ 5
```



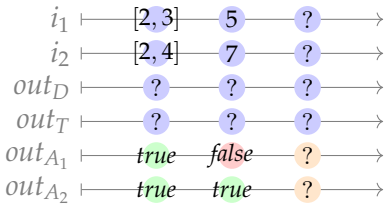
- Input values may be exact, noisy (ranges), or unknown.
- Uncertain values are encoded symbolically.
- Assumption encodes background knowledge.
- Reasoning via SMT solving.
- Additionally, internal streams only implicitly defined via assumption.

Diagnosis in LOLA

- Encode system description as assumption stream.
- Encode observable variables as input streams.
- Encode unobservable variables as internal streams.
- For each component c, encode its health state as Boolean internal stream indicating that c is abnormal at a specific time instant.

Diagnosis in LOLA

$$
\begin{aligned}
&\texttt{internal}\ D, T, A_1, A_2 \ :\ \mathbb{B}\\
&\texttt{in}\ i_1, i_2 \ :\ \mathbb{R}\\
&\texttt{in}\ o_1, o_2 \ :\ \mathbb{B}\\
&\texttt{internal}\ out_D, out_T \ :\ \mathbb{R}\\
&\texttt{ASSUMPTION}\ :\\
&\quad (\neg D \rightarrow (out_D = |i_1 - i_2|))\\
&\wedge\, (\neg T \rightarrow (out_T = (i_1 + i_2)/2))\\
&\wedge\, (\neg A_1 \rightarrow (o_1 \leftrightarrow out_D \geq 1))\\
&\wedge\, (\neg A_2 \rightarrow (o_2 \leftrightarrow out_T - out_D \geq 5))
\end{aligned}
$$



- Outputs of D and T are unobserved.
- Instants 1 and 2 are inconsistent $\Rightarrow$ faults detected.
- Non-empty minimal diagnoses for instants 1 and 2.

Diagnosis in LOLA – Practical Considerations

- System descriptions may be temporal.
- Ideally, diagnoses explain *all* time instants
  (components normal/abnormal over all instants).
- But trace-length independent complexity is usually a requirement for monitoring.
- To guarantee this, it might be necessary to restrict diagnosis to constant-sized recent history (health states assumed only for sliding window), at the cost of diagnostic precision.

How to Obtain System Descriptions?

- Precisely and formally modelling the expected behavior is a major challenge.
- Where should thresholds for sensor readings lie?
- How to interpret multiple sensor readings jointly?

Learning System Descriptions via Machine Learning

Basic idea of approach by Moddemann et al.:

- A model (here a CatVAE) is trained unsupervised on the normal behavior of some components $C \subseteq \text{COMPS}$.
- The model estimates and dicretizes
  - (i) a *state* s_i (from a finite domain) the (sub)system is likely to be in (e.g., *heating* or *cooling*)
  - (ii) a Boolean *residual* r as an indicator of inconsistency
- System description consists of formulas obtained via association-rule mining of the form

$$\left(\bigwedge_{c \in C} \neg \text{AB}(c)\right) \rightarrow (s_i \rightarrow r)$$

- If state is s_i and $r = \mathit{false}$, then at least one $c \in C$ must be abnormal.

Preliminary Work: Integration in the LOLA Framework

General idea:

- Use obtained formulas as assumption.
- Give observation to model, and pipe state and residual to input streams.

```
internal D, T, A1, A2 : 𝔹
in s : State
internal out_D, out_T : ℝ
ASSUMPTION :
```

$$((\neg D \wedge \neg T \wedge \neg A_1 \wedge \neg A_2) \rightarrow (s = s_1 \rightarrow r))$$
$$\wedge\,((\neg D \wedge \neg T \wedge \neg A_1 \wedge \neg A_2) \rightarrow (s = s_2 \rightarrow r))$$
$$\wedge\,\ldots$$

- Enables diagnosis in otherwise infeasible domains at the cost of formal soundness guarantees.

Considerations and Open Questions

Multiple models:

- One model can only construct rules for a fixed $C \subseteq \text{COMPS}$ where the $c \in C$ are indistinguishable from each other.
- Multiple models for partially overlapping subsystems.
- Requires more observable values.

Considerations and Open Questions

Incorporating temporal aspects:

- Moddemann et al. produce only one state and residual for entire observations (i.e., in the LOLA setting over an entire trace).
- Can the models be trained, such that the temporal evolution of the system is taken into account?
- If yes, is this also possible for the sliding windows?

Considerations and Open Questions

Mixing symbolic and subsymbolic system descriptions:

- Can precise descriptions for subsystems be combined with ML-derived descriptions?
- How to take advantage of overlapping components?
- Additional rules that reason over state transitions possible, but states are unlabeled (symbol grounding problem).

Conclusion

- Machine learning can help to derive system descriptions for model-based diagnosis.
- The approach can in principle be applied in the LOLA-based framework.
- However, there are many challenges and open questions that require future research.