On the Use of Logic-Based Learning for Detecting Chemical Process Failures: a Case Study — Supplementary Material

Anonymous Author(s)
Submission Id: 86

ACM Reference Format:
Anonymous Author(s). 2026. On the Use of Logic-Based Learning for Detecting Chemical Process Failures: a Case Study — Supplementary Material. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 3 pages.

A DATASET

The datasets we have generated are tabular and have the process variables as features, along with the fault being simulated and the index of the simulation run. In the dynamic case, the timepoint being observed is an additional feature. The static and dynamic datasets have respective dimensions of 1501x27 and 27000x28. In order to generate the training data for T_{static}, we also have a table containing the ordering of components in the process (i.e., a set of tuples, each of specifying a component and one of its downstream components). Extracts from these three tables are shown in Figure 1.

failure	run	m1_pv	e2_tti	...
missingOxygen_src	0	150.0	399.3	...
missingOxygen_src	1	150.0	400.3	...
missingOxygen_src	2	150.0	395.0	...
...	...	...	...	...

(a) Dataset for T_{static}

failure	run	t	m1_pv	e2_tti	...
missingOxygen_src	0	0	151.1	295.2	...
missingOxygen_src	0	2	151.5	293.5	...
missingOxygen_src	0	4	153.5	291.9	...
...	...	...	...	...	...

(b) Dataset for $T_{dynamic}$

from_component	to_component
srcr1	xv3
xv3	m2
xc2	xv3
...	...

(c) The ordering of components in the process

Figure 1: Extracts from the three tables used to generate training and validation data for our work

B GENERATED RULES

The rules generated by our approach for each of the three main sets of failures considered are shown in Figures 2, 3 and 4. In these figures, "unchanged" means that the value of a given process parameter has remained approximately constant between the initial state (when the failure occurs) and the considered t_{short_term} timepoint at which the measurements are taken, while ↗ and ↘ indicate that the value of the process parameter being referred to has either increased or decreased, respectively. As expected, the complexity of the rules increases and the average rule probability decreases as the task becomes more difficult.

C CONFUSION MATRICES

We have generated confusion matrices to evaluate our learned models and compare their classification abilities to the baselines. Assuming the default learning parameter settings but using all 125 runs for each of the six nontrivial failures used by default, we end up with the confusion matrices shown in Figure 5. Looking at the results, we can see that there is disparity between the baselines, and that they do not consistently identify every failure, though at the cost of uncertainty in our models when there is ambiguity in the data.

D DECISION TREES

Two of the baselines, random forest (RF) and adaptive boosting (AB), provide explainability in the form of decision trees generated by each of the 100 estimators in these baselines. These decision trees are used to make predictions, with the final output being a combination of the result from each estimator. In Figure 6, we provide examples of decision trees used as estimators by RF and AB. Looking at these decision trees more closely, we have observed that many of the process variables used at the branching points of these decisions trees are those that are used in the rules generated by our rule-based learning approach.

E ADDITIONAL INFORMATION

More details can be found in the provided anonymous repository, in which the top-level README.md explains what each directory contains.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.

1: failure(source,missingEthylene) :– m1_pv(T), T $\geq$ 154.
0.4: failure(tempControlLoop,lowSetpoint) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 152.
0.4: failure(coolingWater,lowPressure) :– unchanged(e2_tti), m1_pv(T), T $\geq$ 152, T $\leq$ 152.
0.4: failure(coolingWaterOutValve,stuckClosed) :– unchanged(e2_tti), m1_pv(T), T $\geq$ 152, T $\leq$ 152.
1: failure(flowControlLoop,lowSetpoint) :– unchanged(m3_pv), m2_pv$\searrow$(_).
0.1: failure(coolingWater,lowPressure) :– unchanged(m1_pv), e2_tti(T), T $\leq$ 295.
0.1: failure(coolingWaterOutValve,stuckClosed) :– unchanged(m1_pv), e2_tti(T), T $\leq$ 295.
1: failure(beforeCompressor,leak) :– unchanged(m2_pv), e2_tti$\nearrow$(_).

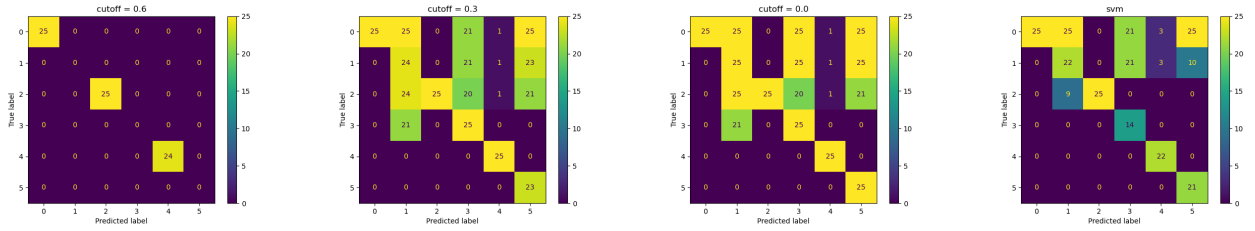
Figure 2: Rules generated by DisPLAS$^+$ for $T_{dynamic}$ using the default learning parameters (m1_pv and e2_tti are in °C)

1: failure(tempControlValve,stuckClosed) :– unchanged(e2_tti), m1_pv(T), T $\geq$ 153, T $\leq$ 153.
0.4: failure(tempControlValve,stuckClosed) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 153.
0.4: failure(source,missingOxygen) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 152.
0.4: failure(coolingWaterOutValve,stuckClosed) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 152.
0.4: failure(tempControlLoop,lowSetpoint) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 152.
0.4: failure(tempControlLoop,highSetpoint) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 152.
1: failure(source,missingEthylene) :– m1_pv(T), T $\geq$ 160.
1: failure(beforeCompressor,leak) :– unchanged(m2_pv), e2_tti$\nearrow$(_).
0.4: failure(tempControlValve,stuckOpen) :– unchanged(e2_tti), unchanged(m2_pv), m1_pv(T), T $\leq$ 152.
1: failure(source,missingEthylene) :– unchanged(e2_tti), m1_pv$\nearrow$(T), T $\geq$ 3.
1: failure(flowControlLoop,lowSetpoint) :– unchanged(m3_pv), m2_pv$\searrow$(_).
1: failure(source,missingOxygen) :– unchanged(m2_pv), e2_tti$\searrow$(_).
0.4: failure(source,missingOxygen) :– e2_tti(T), T $\leq$ 289.
1: failure(source,missingOxygen) :– e2_tti(T), T $\leq$ 288.
1: failure(flowControlLoop,highSetpoint) :– m1_pv(T), m2_pv$\nearrow$(_), T $\leq$ 155.
1: failure(source,missingOxygen) :– unchanged(snk1_p), e2_tti(T), T $\leq$ 290.
1: failure(flowControlLoop,highSetpoint) :– m2_pv$\nearrow$(F), F $\geq$ 6.

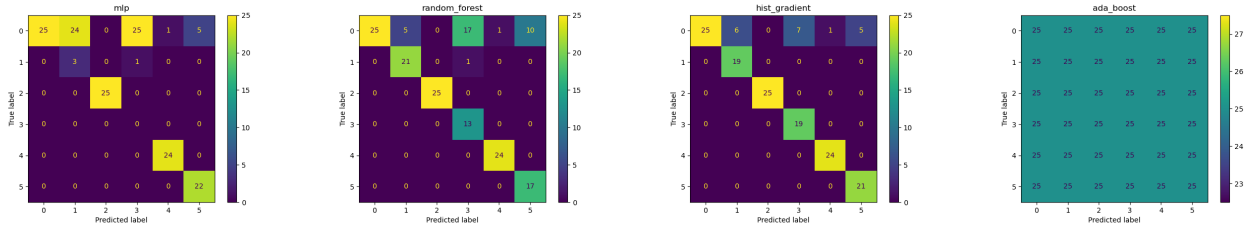
Figure 3: Rules generated by DisPLAS$^+$ for $T_{dynamic}$ using the set of 10 nontrivial failures, and otherwise default learning parameters (m1_pv and e2_tti are in °C, and m2_pv in kg/s)

0.1: failure(source,lowPressure) :– unchanged(srcr1_t), unchanged(e2_tsi), m1_pv(T), T $\leq$ 151.
1: failure(source,lowPressure) :– unchanged(srcr1_t), unchanged(e2_tsi), e2_tti(T), T $\geq$ 296.
0.1: failure(source,lowPressure) :– unchanged(srcr1_t), unchanged(e2_tsi), m1_pv$\nearrow$(_).
1: failure(source,highTemp) :– srcr1_t$\nearrow$(_).
1: failure(source,lowTemp) :– srcr1_t$\searrow$(_).
1: failure(coolingWater,highTemp) :– e2_tsi$\nearrow$(_).
1: failure(coolingWater,lowTemp) :– e2_tsi$\searrow$(_).

Figure 4: Rules generated by DisPLAS$^+$ for $T_{dynamic}$ using the set of (5) trivial failures, and otherwise default learning parameters



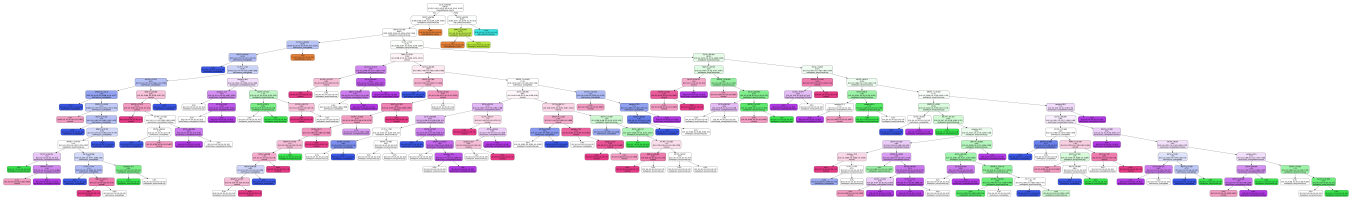
(a) $T_{dynamic}$**, excluding rules with probability** < 0.6 **(b)** $T_{dynamic}$**, excluding rules with probability** < 0.3 **(c)** $T_{dynamic}$**, with all rules** **(d) SVM**



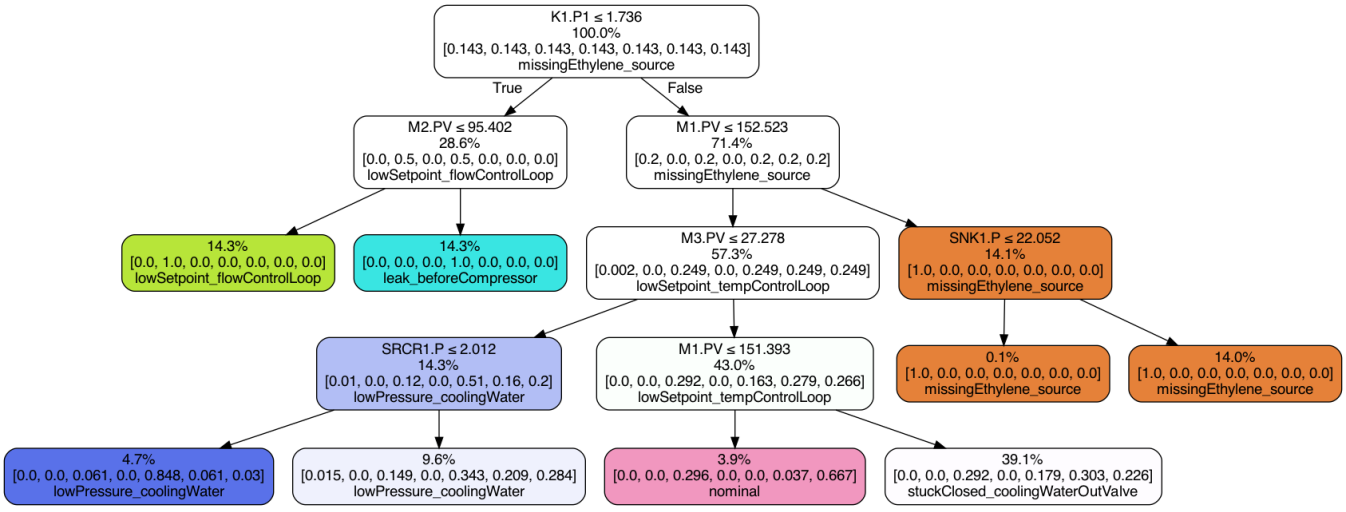
(e) MLP **(f) RF** **(g) HGB** **(h) AB**

Figure 5: Confusion matrices generated for $T_{dynamic}$ **and its baselines, using data from all 125 runs for each of the six nontrivial failures**[1]

[1] {0: "missingEthylene_source", 1: "lowSetpoint_flowControlLoop", 2: "lowSetpoint_tempControlLoop", 3: "leak_beforeCompressor", 4: "lowPressure_coolingWater", 5: "stuckClosed_coolingWaterOutValve"}



(a) RF, using the default maximum depth of 15



(b) AB, using the default maximum depth of 4

Figure 6: Examples of decision trees used as estimators by the baselines that provide explainability, for the six nontrivial failures used by default