QSIM-Guided Weak Supervision for Fault Detection in Cyber-Physical Systems

Ankita Das[1]
Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Roxane Koitz-Hristov
Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Franz Wotawa[2]
Institute of Software Engineering and Artificial Intelligence, Graz University of Technology, Austria

Abstract

This paper presents an initial investigation into the combination of qualitative simulation and machine learning for fault detection in cyber-physical systems. In recent years, the detection and diagnosis of faults in cyber-physical systems has become increasingly dependent on machine learning. However, supervised machine learning approaches rely on large labelled datasets, which are rarely available during the initial deployment phase. In systems such as smart buildings, faults occur rarely, and even the nominal behaviour of a system can vary across structurally identical installations. We therefore propose a framework that addresses this issue by generating weak supervision labels using qualitative reasoning. Given a qualitative model that captures the system's nominal behaviour, we use a conformance check to determine whether the system traces are likely to be faulty or not. These binary pseudo-labels are then used to train a downstream classifier. Unlike statistical anomaly detection or self-supervised techniques, the generated supervision signal is based on physical consistency rather than deviations from a learned baseline. We evaluate our approach on four benchmarks relating to electrical, fluid, mechanical and thermal dynamics, comparing our methodology with unsupervised anomaly detection and supervised learning with limited labels. Our initial experiments suggest that, while qualitative conformance checks can provide usable training signals when labelled data is unavailable, their effectiveness depends on the discriminative structure of the qualitative model.

2012 ACM Subject Classification Computing methodologies → Causal reasoning and diagnostics; Computing methodologies → Spatial and physical reasoning; Computing methodologies → Modeling methodologies

Keywords and phrases Qualitative Simulation, Fault Detection, Outlier Detection

Digital Object Identifier 10.4230/OASIcs.DX.2026.11

Funding The work presented in this paper has been supported by the FFG project Artificial Intelligence for Smart Diagnosis in Building Automation (ALFA) under grant 914932.
Ankita Das: Supported by by the Christian-Doppler Forschungsgesellschaft (CDG) project AI-based Diagnosis for Energy Transition and the Circular Economy (AID4ETCE).

Acknowledgements The following AI-based tools were used for language editing and polishing: ChatGPT 5.5 and Claude 4.7.

1 Introduction

Cyber-physical systems (CPS) tightly couple physical processes and computing components through the use of sensors, actuators, and feedback loops to enable smart monitoring and control in buildings, industrial automation, or transportation. Fault detection and diagnosis

[1] Authors are listed in alphabetical order.
[2] Corresponding author.

(FDD) of CPS is becoming increasingly data-driven with the goal of improving the operational efficiency, reliability, and safety of CPS [15]. Machine learning (ML) and deep learning approaches have shown promising results for anomaly detection [27, 21] and predictive maintenance [26]. However, these methods typically require large operational datasets and reliable fault annotations to sufficiently capture the system behavior and generalize across operating conditions.

In practical CPS settings, this requirement is often difficult to satisfy. In many CPS domains, fault data is rare, unavailable at the beginning of the system's life cycle, or expensive to acquire [7]. Let us consider smart buildings, a domain that suffers strongly from the cold-start problem, i.e., the absence of sufficient representative training data during early deployment phases. Newly commissioned buildings do not provide an extensive amount of operation data and faults occur rarely. This leads to little to no training data being available. Moreover, even nominal behavior may vary substantially across structurally identical systems due to installation specific-parameters, environmental conditions, or occupant behavior. Thus, ML models trained on one building generalize poorly to others [20, 31].

Several research areas address this issue by reducing the data and annotation requirements of learning-based diagnosis, e.g., transfer learning [13, 10], self-supervised learning [14, 32], and anomaly detection [4]. Yet, these approaches typically depend on sufficiently representative operational data, assume the availability of an initial fault-free operational phase, or infer abnormal behavior from deviations from a learned statistical baseline rather than from violations of physical consistency. Weak supervision is a complementary line of research that addresses label scarcity by replacing expert-provided annotations with programmatic labeling functions encoding heuristics, rules, or external knowledge sources [24, 33]. The quality of weak supervision, however, depends on the quality of these labeling sources. For CPS, the most reliable source is the system's: a fault is, by definition, a behavior that the underlying physical model cannot produce when it functions correctly. Thus, symbolic reasoning from first principle, e.g., realized through qualitative reasoning, presents itself as a natural supervision source for diagnosis.

Qualitative reasoning provides a way to formalize physically plausible system behavior and structure without the need for precise numeric values [8, 11, 18, 28]. In contrast to quantitative approaches, such as digital twins or physics-informed neural networks, qualitative models describe the system behavior through qualitative variables that encompass qualitative magnitudes (e.g., positive, negative, or zero) and qualitative directions of change (e.g., increasing, decreasing, or steady) [11]. In our previous work [5, 6], we have shown that qualitative conformance checking, i.e., comparing observed traces against the set of behaviors permitted by a qualitative simulation model, can detect faults in CPS without requiring precise parameterization.

While qualitative conformance checking provides interpretable and physically grounded diagnosis results, purely symbolic reasoning approaches may become computationally expensive for continuous online monitoring and large-scale deployment. In contrast, ML-based classifiers provide efficient runtime inference and scalable deployment, but face the aforementioned cold-start problem. This motivates us to combine both paradigms by using qualitative reasoning as a symbolic supervision mechanism for training ML fault detectors.

In this paper, we propose a framework that exploits qualitative conformance outcomes between a qualitative model and observed CPS traces to generate weak supervision labels for fault detection. In essence our approach can be divided into two stages. First, we exploit a QSIM model of the nominal system behavior to detect deviations in system observation traces by checking the trace's conformance with the admissible system behavior generated

by the qualitative simulation. In case the system behavior does not conform to any of the qualitative simulation traces, we label the trace as *non-conforming*, i.e., faulty; otherwise the trace is labeled *conforming*, i.e., nominal. Second, we use those pseudo-labels to train a binary ML classifier. To determine the feasibility and effectiveness of our approach, we evaluate our framework on multiple benchmarks and a real-world system under cold-start conditions and compare our method against off-the-shelf anomaly detection methods.

The remainder of this paper is structured as follows. In Section 2, we discuss related work. Subsequently, we give background information on qualitative simulation and our conformance checking approach in Section 3. Section 4 describes how we exploit qualitative simulation and conformance checking to obtain weak labels, while Section 5 and Section 6 present our initial empirical study and its results, respectively. Lastly, we conclude the paper and provide directions for future work in Section 7.

2 Related Work

A substantial body of research addresses FDD in CPS in low-label environments. For instance, transfer learning approaches reuse models across installations or operating conditions to compensate for limited data availability. Genkin and McArthur [13] combine transfer learning with reinforcement learning to transfer optimal control policies from an existing building to a newly commissioned structure. While reducing the warm-up period and overall mean variance, their approach requires a transfer building with a wide range of sensors to work effectively. Dou et al. [10] exploit simulation data from a source building to pre-train neural network models that are then fine-tuned on the measurements of the target installation with promising results.

Self-supervised learning (SSL) is another widely used strategy for low-label environments. SSL exploits unlabeled data and a pretext task to learn structure and relationships in the data to generate supervisory signals without the need for human annotations [14]. Senanayaka et al. [25] combine one-class support vector machine (SVM) for health estimation with supervised convolutional neural network (CNN) classification for online fault diagnosis in electric powertrains. First, the one-class SVM is trained on available nominal data to define the machine health status; second, a CNN classifier is trained on those health classes. Although explicit labeling is no longer a prerequisite, such methods still require a sufficiently representative operational data and may not be able to generalize across heterogeneous system configurations [32].

Anomaly and outlier detection methods, such as one-class SVMs, are also commonly used when labeled fault data is unavailable. Those methods train exclusively on nominal system behavior, avoiding the need for fault data altogether. While requiring less data than traditional supervised learning methods, these methods assume an initial fault-free start-up phase and rely on statistical methods rather than physical inconsistencies to determine abnormal behavior [4].

A complementary line of research addresses low-label environments by changing the source of supervision rather than the data requirements. Weak supervision is one such technique that reduces dependence on human-provided annotations through programmatic labeling functions that encode heuristics, rules, or other external knowledge sources to produce training labels [33, 24]. Frameworks, such as Snorkel [24], demonstrate that programmatically generated labels can be used successfully for training models in domains where manual annotations are infeasible or expensive. Snorkel combines multiple labeling functions through a generative model to produce probabilistic training labels at scale. Weakly supervised learning has increasingly been explored in predictive maintenance and CPS settings to address incomplete, inexact, or inaccurate supervision [22].

In contrast to existing weak supervision and SSL approaches, our work derives supervisory signals not from statistical anomalies but from symbolic physical reasoning. Building on our previous work on qualitative simulation and conformance checking [5, 6], we use qualitative conformance outcomes as weak labels for downstream supervised learning. Similar to prior pseudo-labeling approaches, as the one presented by Senanayaka et al. [25], our framework employs an intermediate diagnosis mechanism to generate labels for a supervised learning classifier. Our proposed approach generates labels from violations of physically admissible system behavior encoded in a qualitative model.

3 Qualitative Simulation (QSIM)

Qualitative Simulation (QSIM) provides an abstract representation of dynamical system behavior by replacing precise numerical values with symbolic categories and qualitative relations [11, 9, 19, 18]. Instead of modeling variables through exact measurements, QSIM represents them using qualitative values such as *low*, *medium*, *high*, *on*, and *off*, together with constraints that capture structural and physical dependencies between variables. This abstraction is particularly useful for CPS, where the qualitative structure of the system is often known even when exact parameter values or detailed numerical models vary.

A key advantage of qualitative simulation is that it can describe families of behaviors rather than a single numerical trajectory. For example, changes in resistance, capacitance, inflow rate, damping, or thermal resistance may alter the exact numerical trajectory of a system, while preserving its qualitative structure. QSIM exploits this property by reasoning over qualitative magnitudes, landmarks, intervals, and qualitative trends. As a result, the same qualitative model can often be reused across different parameterizations of a benchmark, provided that the qualitative relationships between variables remain unchanged.

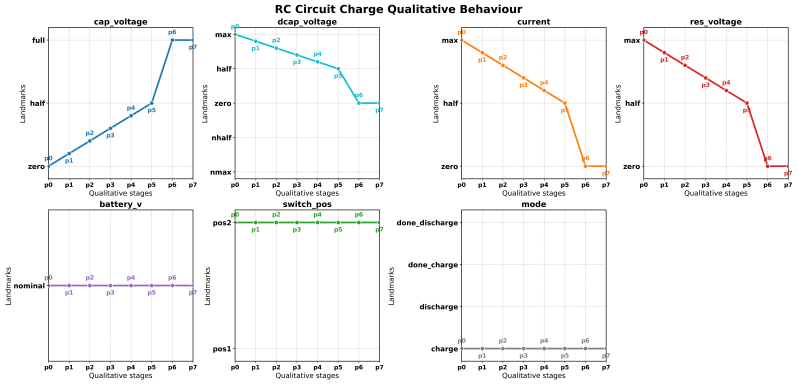
In this work, we use ASP-QSIM, an Answer Set Programming (ASP) formulation of qualitative simulation [30]. In ASP-QSIM, system states, transitions, and qualitative constraints are encoded declaratively as logic predicates. Solving the resulting ASP program yields one or more answer sets, where each answer set corresponds to one admissible qualitative evolution of the modeled system over time.

A qualitative *trace* denotes one such admissible system evolution. It consists of:

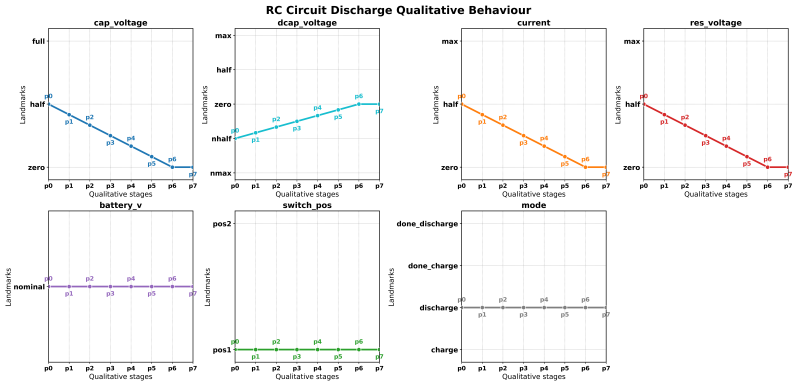
- *time points*, such as $p(0), p(1), \ldots$, representing qualitative system states,
- *intervals*, such as $i(0, 1)$, representing transitions between successive states, and
- *qualitative assignments* of system variables at time points and intervals, for example $v_{heating\_request}(p(0)) = on$ or $v_{supply\_temp}(p(1)) = high$.

Within the ASP-QSIM encoding, predicates such as `holds/4` represent qualitative variable values and trends at particular time points or intervals, while predicates such as `holds_constraint/4` encode qualitative relations that must hold between variables. These relations may express structural dependencies, monotonic influences, qualitative proportionalities, or admissible state transitions. Hence, a trace captures not only the qualitative state of the system, but also how this state evolves over time.

Since qualitative simulation generally admits multiple valid evolutions, the model defines a set of possible qualitative behaviors rather than a single deterministic trajectory [19, 18]. This is important for our setting because the exact numerical trajectory of a cyber-physical system may vary across parameter settings, initial conditions, and operating regimes. Instead of requiring one exact numerical prediction, the qualitative model specifies which classes of behavior are physically admissible. However, while local qualitative simulation algorithms,



(a) RC circuit charging behavior.



(b) RC circuit discharging behavior.

Figure 1 Example qualitative behaviors generated by the ASP-QSIM model for the RC circuit benchmark. The charging and discharging cases illustrate admissible qualitative evolutions of the nominal system, which are used as reference behaviors for downstream conformance checking.

such as QSIM, are able to produce every physically possible behavior, they may also generate impossible spurious traces. This follows from the inherently local nature of state expansion: each successor is derived from the information in the current state alone [17].

Figure 1 illustrates this idea for a simple RC circuit. The charging and discharging regimes are represented by separate ASP-QSIM reference behaviors. In the charging case, the capacitor voltage increases toward the supply voltage, while in the discharging case the capacitor voltage decreases. These figures illustrate that the QSIM model captures qualitative evolution patterns rather than exact numerical curves. Such admissible qualitative behaviors form the reference set used for conformance checking.

This property is central to our framework. We use the set of admissible qualitative traces generated by the nominal ASP-QSIM model as a reference model of expected system behavior. Observed CPS traces are first abstracted into the same qualitative representation using benchmark-specific landmarks and intervals. The abstracted observations are then checked for conformance against the admissible behaviors of the nominal model. Intuitively, an observed trace is conforming if there exists at least one admissible ASP-QSIM behavior that is consistent with the observed qualitative assignments. If no such behavior exists, the observation is considered non-conforming (see Das et al. [5]).

Operationally, conformance checking is performed by adding the observed qualitative assignments as constraints to the ASP-QSIM program. If the resulting program has at least one answer set, then the observation is compatible with the nominal qualitative model. If the program becomes unsatisfiable, then the observed behavior cannot be explained by the nominal model and is treated as a potential fault. Thus, ASP-QSIM provides a model-based mechanism for converting qualitative physical consistency into binary conformance outcomes.

In the proposed weak-supervision framework, these conformance outcomes are used as pseudo-labels for a downstream ML task. The next section describes our approach.

4 Approach

We address the cold-start problem by combining a qualitative model of nominal system behavior with conformance checking [1, 23], using qualitative simulation as a source of weak supervision for ML. The central assumption is that, in many CPS, expert knowledge about nominal behavior is available before large amounts of labeled fault data can be collected. We encode this knowledge as a nominal QSIM model [19]. Observed quantitative traces are then abstracted and checked for conformance against the admissible behaviors generated by this model. The resulting conformance outcomes are interpreted as pseudo-labels and used to train downstream classifiers.

Figure 2 summarizes the weak-label generation pipeline. Each quantitative time-series trace, obtained either from simulation or system operation, is divided into fixed-length temporal windows. These windows define the quantitative feature vectors used by the ML models. In parallel, each window is abstracted into qualitative states using the benchmark-specific landmarks and intervals. The abstracted window is then checked for conformance against the nominal ASP-QSIM model. If the window is consistent with at least one admissible qualitative behavior, it is considered conforming; otherwise, it is considered non-conforming.

The window-level conformance outcomes are aggregated per original trace to obtain a trace-level pseudo-label. This step is necessary because the benchmark ground-truth labels are defined at the trace level, while the ML instances are defined at the window level. The resulting QSIM-derived pseudo-label is assigned to the quantitative windows from the same trace, producing a weakly labeled dataset for classifier training. Thus, the classifier learns from concrete quantitative features, while the supervision signal is derived from qualitative physical consistency.

This design has three main advantages. First, it reduces the need for manually labeled fault data, which is particularly important in cold-start settings. Second, windowing increases the number of training instances compared with labeling only complete traces. Third, the learned classifier can capture numerical regularities in the concrete system behavior that are not explicitly represented in the qualitative abstraction. The trade-off is that window-level conformance provides less temporal context than complete-trace conformance and can therefore be sensitive to abstraction thresholds, window boundaries, and initial-state assumptions. We therefore treat the resulting labels as weak labels rather than ground truth. The binary weak labels that result from these compliance results are then used to train ML models downstream. The framework eliminates the need for laborious human labeling by transforming model-based behavioral consistency into a learning signal. In order to determine whether qualitative conformance can facilitate efficient fault detection under sparse annotation, the final models are assessed on held-out traces and in low-label circumstances.

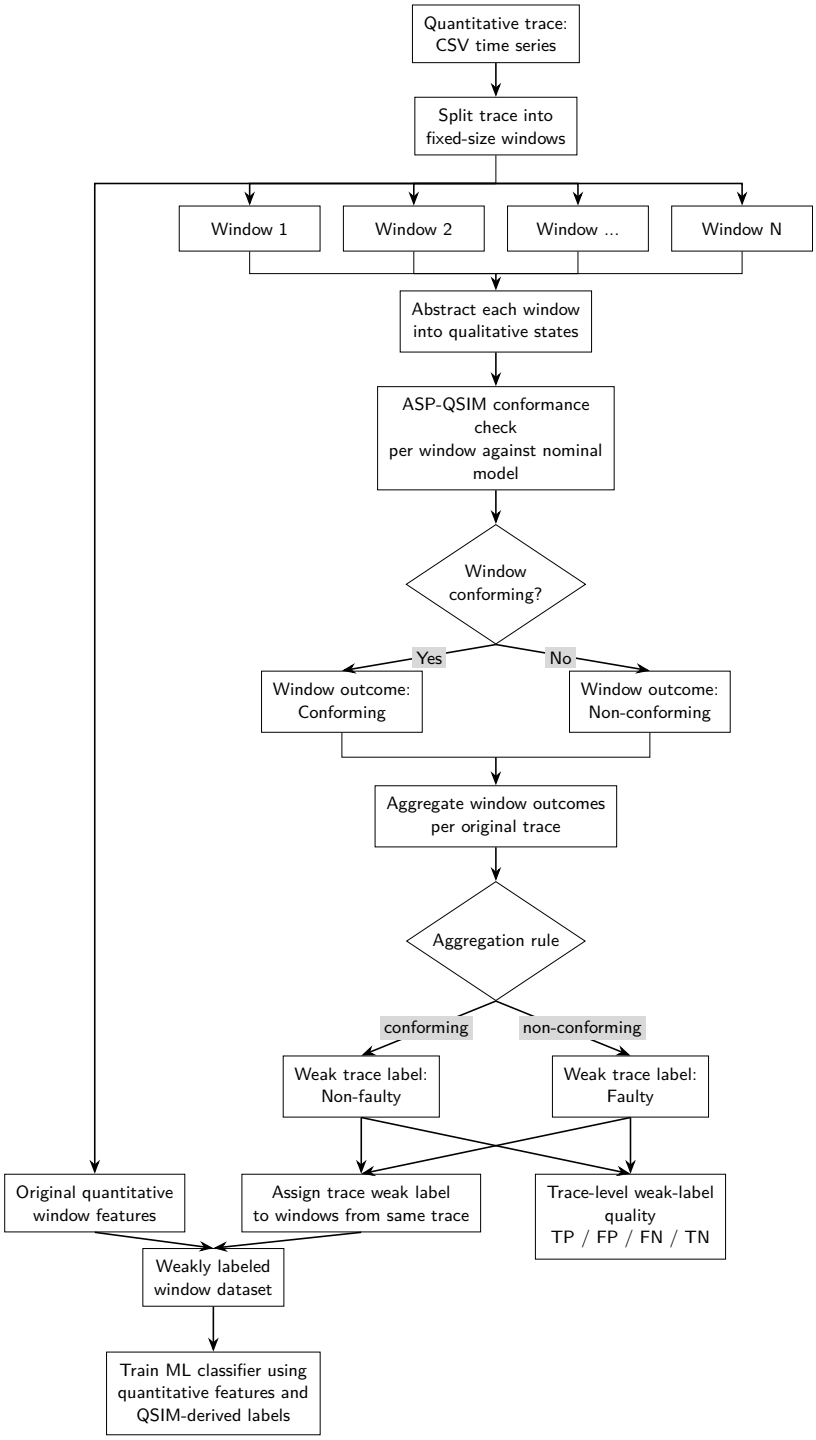


Figure 2 Window-level QSIM weak-label generation pipeline. Quantitative traces are segmented into windows, abstracted into qualitative states, and checked for conformance against the nominal ASP-QSIM model. Window-level conformance outcomes are aggregated into trace-level pseudo-labels, which are assigned to quantitative window features for downstream ML training and trace-level evaluation.

5 Evaluation

In this section, we outline our initial experiments. First, we describe the four benchmark systems, before detailing the experimental setup.

5.1 Benchmark Description

We examine four benchmarks that vary in terms of both system complexity and the kind of qualitative dynamics they display in order to assess the suggested framework. From a qualitative perspective, the benchmarks cover distinct classes of system dynamics. Together, these benchmarks allow us to study the proposed approach on both compact illustrative systems and a practically relevant control application.

5.1.1 RC circuit

The RC circuit, as shown in Figure 3a, serves as a compact benchmark for transient qualitative behavior. Its dynamics are governed by the interaction between switch position, voltage source, capacitor voltage, current, resistor voltage, and the derivative of the capacitor voltage. Qualitatively, the benchmark exhibits structured charging and discharging processes with clear landmark transitions and trend changes. This makes it particularly suitable for analyzing how nominal and faulty behaviors can be distinguished through qualitative traces and how such distinctions can be converted into weak labels.

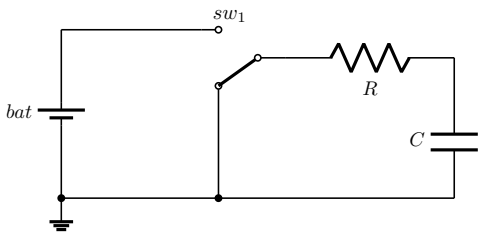
5.1.2 Bathtub

The bathtub benchmark provides a simple but expressive example of accumulation-driven behavior [19, 11]. As can be seen in Figure 3b, the system consists of a reservoir with controlled inflow and gravity-driven outflow, such that the water level evolves according to the balance between incoming and outgoing flow. In qualitative terms, this benchmark captures filling, draining, and equilibrium behavior, making it well suited for examining how deviations in flow conditions affect trace evolution and conformance. For this benchmark, we used the public ASP-QSIM implementation of Wiley et al. [30] as a starting point and adapted it to our experimental setting.[3]

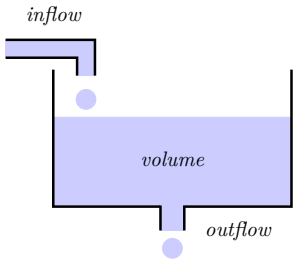
5.1.3 Spring–mass–damper

The spring–mass–damper benchmark represents a classical mechanical system with oscillatory behavior and our ASP-QSIM implementation follows the qualitative formulation introduced by Coghill et al. [3]. It consists of a mass connected to a fixed support through a spring and a viscous damper operating in parallel, and is subject to an external force (see Figure 3c). The resulting dynamics are characterized by restoring forces, dissipation, and damped oscillations. From a qualitative perspective, this benchmark is useful because it introduces non-monotonic behavior and repeated changes in trend, which are more challenging than purely charging or accumulation-based dynamics. It therefore provides a complementary test case for assessing whether conformance-based weak supervision can handle mechanically induced temporal patterns.

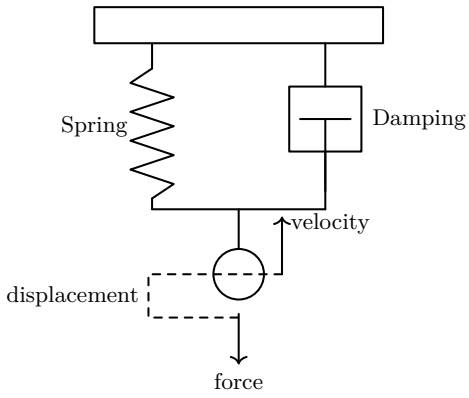
[3] `https://github.com/timothy-wiley/aspqsim`



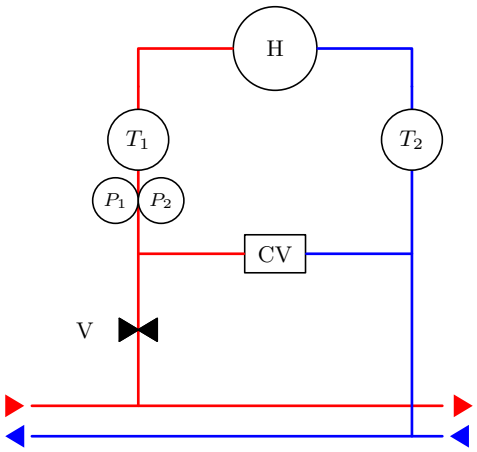
(a) Simple RC circuit.



(b) Bathtub.



(c) Spring–mass–damper system [3].



(d) Heating circuit.

Figure 3 The four benchmark systems used in this study: (a) an RC circuit with resistor R, capacitor C, switch sw_1, and battery *bat*. (b) a bathtub with inflow, outflow, and volume; (c) a spring–mass–damper with displacement, velocity, and external force; (d) a heating circuit with heat exchanger H, two alternating pumps P_1 and P_2, a globe valve V, a check valve CV, and temperature sensors T_1 and T_2.

5.1.4 Heating circuit

The heating circuit, shown in Figure 3d [16], constitutes the most application-oriented benchmark in our study as it is the only system where real-world traces are available to us. It models a hydraulic heating loop that supplies heat from a district heating station to a commercial building. District heating systems are an important component of sustainable urban energy infrastructure [29], and faults in such systems can lead to substantial energy losses [2].

For this benchmark, we use a simplified QSIM model to evaluate whether compact qualitative knowledge can provide useful weak labels on real data. The model covers active heating operation under two pump regimes, `pump1_only` and `pump2_only`, with one regime-specific nominal model for each configuration. In the first regime, pump P_1 is active and pump P_2 is inactive; in the second regime, pump P_2 is active and pump P_1 is inactive. A stable-regime constraint prevents unintended switching between pump regimes during conformance checking.

The qualitative variables considered in the simplified heating model include outdoor temperature, heating request, pump states, heat-exchanger flow, valve position, supply temperature and return temperature. Outdoor temperature and valve position are treated as

contextual variables to tolerate delays and feedback effects in the real data. Thermal behavior is captured coarsely through a monotonic relation between supply and return temperature during active circulation. The model is therefore intentionally conservative: it abstracts away the full low-level controller and tests whether compact nominal knowledge can still provide useful weak supervision for fault detection.

5.2 Experimental Setup

For each benchmark except the heating system, we generated simulated traces under both nominal and faulty conditions using Modelica models [12]. In order to assess our approach on various system configurations, we used parameter sweeps over operating regimes and initial conditions. Table 1 gives an overview of the parameter ranges considered. Ground-truth labels were obtained directly from the simulation setup: traces generated under nominal conditions were treated as *non-faulty* whereas traces generated under injected fault conditions were treated as *faulty*.

Table 1 Benchmark parameter ranges and conformance results between the qualitative models and the corresponding nominal quantitative simulation behaviors.

Benchmark	Parameter	Range
RC Circuit	R resistance	$\{9, 10, 11\}$
	C capacitance	$\{0.0008, 0.001, 0.0012\}$
	V_{bat} voltage	$\{4.0, 4.5, 5.0\}$
Bathtub	q_{in} inflow	$\{0.04, 0.08, 0.12\}$
	A_d drain area	$\{0.002, 0.005, 0.010\}$
	$V(0)$ initial volume	$\{0.10, 0.20, 0.40\}$
Spring–Mass–Damper	k spring constant	$\{0.5, 1.0, 2.0\}$
	c damping coefficient	$\{0.3, 1.0, 3.0\}$
	$x(0)$ initial position	$\{-0.5, 0, 0.5\}$
	$\dot{x}(0)$ initial velocity	$\{-0.5, 0, 0.5\}$

Across the simulated benchmarks, faults were injected as separate Modelica model variants targeting the main physical components of each system. In the RC circuit, faults include resistor short and open circuits, capacitor faults, switch faults, intermittent contacts and battery-supply faults, which affect current flow and voltage dynamics. In the bathtub benchmark, faults modify the water-balance behavior through stuck or excessive inflow, drain blockage, leakage, altered tub geometry, overflow-threshold changes, and intermittent blockage. In the Spring–Mass–Damper benchmark, faults affect actuator force, spring stiffness, damping, mass and friction thereby changing the oscillatory dynamics. For the real heating circuit, the faulty traces correspond to two valve faults: a leaking valve and a stuck valve. Table 2 summarizes these fault families.

Across all benchmarks, quantitative system variables are abstracted into qualitative levels and linked through qualitative constraints. Derivatives are included when they are part of the benchmark model or can be robustly estimated from the trace; otherwise, the abstraction relies on qualitative magnitudes only. The nominal QSIM model defines the admissible qualitative behavior of the system. Observed nominal and faulty simulation traces are then abstracted and checked against this nominal behavior model to derive conformance outcomes for weak-label generation.

Table 2 Summary of fault types used in each benchmark.

Benchmark	#Faults	Fault types
RC circuit	10	Resistor, capacitor, battery and switch faults including short, open, degraded and intermittent behaviors.
Bathtub	10	Inflow, drain, overflow, tub area error and intermittent blockage faults.
Spring–Mass–Damper	10	Actuator, spring, damper, mass and friction faults.
Heating circuit	2	Real valve faults: leaking valve and stuck valve.

To obtain ML instances, each continuous trace was segmented into fixed-length windows. Unless otherwise stated, we used windows of 100 consecutive samples, e.g., samples 0–99, 100–199, and so on. These windows were used to construct the quantitative feature vectors for downstream learning. To ensure comparability across methods, the amount of raw operational data, trace duration, and number of windows were kept as consistent as possible across the anomaly detection, weak-supervision, and low-label supervision settings.

Weak labels were generated using window-level qualitative conformance checking. Each window was abstracted into qualitative states and checked against the corresponding nominal QSIM model. The resulting window-level conformance outcomes were aggregated per original trace to obtain QSIM-derived pseudo-labels. These pseudo-labels were then assigned to the quantitative windows from the same trace and used as binary labels for our weakly supervised training regime.

For the simulated benchmarks with parameter ranges, windows from all parameter configurations were pooled after applying a temporal split within each trace. The first 70% of windows from each trace were used for training, and the remaining 30% were used for testing. As a result, both portions contain examples from all simulated parameter configurations. Thus, both splits contain examples from all simulated configurations, allowing us to evaluate performance when the operating range is represented during training.

We considered five supervision regimes:

- **Full supervision**, where classifiers were trained using ground-truth labels on the first 70% of windows from each trace, including both nominal and faulty traces. The remaining 30% of windows from each trace were used for testing. As applied logistic regression, random forests and decision trees for supervised learning.
- **Weak supervision**, where classifiers were trained on quantitative window features using QSIM-derived pseudo-labels instead of manually provided labels. The same temporal split as in full supervision was used and performance was evaluated against the ground-truth labels.
- **Low-label supervision**,where classifiers were trained using only a small subset of ground-truth labels from the training split. This setup represents a cold-start scenario in which only a few manually labeled examples are available. The selected labeled windows were balanced at the binary class level between *non-faulty* and *faulty* examples. Thus, LL-5 used 3 non-faulty and 2 faulty labeled windows, LL-10 used 5 non-faulty and 5 faulty labeled windows, and LL-20 used 10 non-faulty and 10 faulty labeled windows.
- **Unsupervised anomaly detection**, where models were trained only on nominal windows. This setting reflects the assumption that some fault-free operation may be available during system start-up or commissioning, while fault labels are unavailable. We trained One-Class SVM and Isolation Forest on the first 70% of nominal windows and tested them on the remaining nominal windows together with faulty windows.

- **Full Supervision across configurations**, where models are trained on one configuration, i.e., parameter setting, and evaluated on held-out configurations, emulating the transferal of a model learned on one instance to another instance with varying parameters. As with the supervised case we use the ground truth, however we train on 100% of the samples of the source configuration and test on 100% of the other configurations.

To assess the quality of the weak-label signal itself, we additionally compare QSIM-derived labels against the simulation ground truth using precision, i.e., Precision $= \frac{\mathrm{TP}}{\mathrm{TP+FP}}$, recall, i.e., Recall $= \frac{\mathrm{TP}}{\mathrm{TP+FN}}$, F1 score, i.e., F1 $= 2 \cdot \frac{\mathrm{Precision \cdot Recall}}{\mathrm{Precision+Recall}}$, and the underlying classification statistics (i.e., true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN)).

We report accuracy, i.e., Accuracy $= \frac{\mathrm{TP+TN}}{\mathrm{TP+TN+FP+FN}}$, and Macro-F1 as the main predictive metrics for downstream classification. Macro-F1 is the unweighted average of the classwise F1-scores, so that faulty and non-faulty classes contribute equally even under class imbalance, i.e., Macro-F1 $= \frac{\mathrm{F_1+F_2+\ldots+F_N}}{\mathrm{N}}$, where $\mathrm{F_1}$ is the F1 score for class 1 etc. and N is the total number of classes.

All benchmarks use the same general weak-supervision idea, with benchmark-specific modifications where necessary. Bathtub and Spring–Mass–Damper use the generic window-level conformance pipeline. For the RC circuit, we additionally use phase-aware aggregation and trace-level representations because RC traces contain structured charging and discharging phases separated by switching transitions. These extensions allow the weak-labeling procedure to account for stable phases, transition windows and settling behavior more explicitly than a plain fixed-window approach.

For the heating system benchmark, the raw operational CSV traces were first transformed into the same window-level format used for the other benchmarks. Since the real data contain multiple operating regimes, each window was assigned to a pump regime based on the dominant pump state within that window. Windows dominated by `pump1_only` behavior were checked against the pump1 nominal model, while windows dominated by `pump2_only` behavior were checked against the pump2 nominal model. Mixed or inactive windows were treated conservatively as non-conforming, since they do not match either active nominal regime model.

6 Results

In this section, we analyze the results of our initial study. First, we discuss the accuracy of our labeling approach, before elaborating on the classification results of the various supervision regimes.

In particular, we first explain the results on the fully supervised, weakly supervised and low-label approaches, before elaborating on the results of the unsupervised learning models. In addition, we report on cross-configuration results in the full supervision case.

6.1 Weak Labeling Accuracy

Table 3 reports the quality of the QSIM-derived weak labels before any downstream classifier is trained. Faulty behavior is treated as the positive class. Thus, TP are faulty examples correctly assigned faulty/non-conforming weak labels by QSIM, TN are nominal examples correctly assigned non-faulty/conforming weak labels, FP are nominal examples incorrectly rejected by the QSIM-based weak-labeling procedure and FN are faulty examples incorrectly

Table 3 Quality of QSIM weak labels per benchmark, evaluated against ground-truth trace labels. TP/FP/FN/TN counts treat "Faulty" as the positive class. Precision (Prec.), recall (Rec.), and F1-score characterize the weak-label signal before any classifier is trained.

	Ground truth		Counts				Quality		
Benchmark	Faulty	Non-Faulty	TP	FP	FN	TN	Prec.	Rec.	F1
RC circuit	270	27	243	0	27	27	1.000	0.900	0.947
Bathtub	270	27	198	21	72	6	0.904	0.733	0.810
Spring–mass–damper	810	81	635	57	175	24	0.918	0.784	0.846
Heating	920	805	198	53	722	752	0.789	0.215	0.338

TP: QSIM = Faulty, Ground truth = Faulty; FP: QSIM = Faulty, Ground truth = Not faulty; FN: QSIM = Not faulty, Ground truth = Faulty; TN: QSIM = Not faulty, Ground truth = Not faulty.

assigned conforming weak labels. Precision measures how reliable a QSIM-generated fault label is, whereas recall measures how many ground-truth faulty examples are recovered by the conformance check.

For the RC circuit, the weak labels are highly reliable. The absence of false positives ($FP = 0$) shows that nominal traces are consistently accepted by the qualitative model, while the high number of true positives ($TP = 243$) indicates that most injected faults produce qualitative deviations. The remaining errors are false negatives ($FN = 27$), meaning that some faulty traces are not rejected by the QSIM-based labeling procedure. This is consistent with the fault-level analysis: most capacitor, resistor, supply, and switch faults are detected, whereas the fault representing a loose connection or failing contact at the resistor, remains often undetected.

The bathtub benchmark shows a noisier but still useful weak-label signal. QSIM detects many faulty traces ($TP = 198$), resulting in good precision (0.904), but it also misses a non-negligible number of faults ($FN = 72$). The false positives ($FP = 21$) indicate that some nominal traces are over-rejected by the operational labeling pipeline. This does not necessarily mean that the qualitative model fails to describe the intended nominal behavior. Faults that strongly alter the qualitative structure, such as the fault where the effective area is scaled, are detected well, while several inflow-related faults like `InflowStuckClosed` and `InflowStuckOpenHigh` remain closer to nominal behavior and are therefore more frequently missed.

For the Spring–Mass–Damper benchmark, the weak labels also provide a useful signal, with high precision (0.918) and reasonably high recall (0.784). However, the model still produces both false positives and false negatives. The false positives indicate occasional over-rejection of nominal traces while the false negatives show that some faulty traces remain difficult to distinguish from nominal behavior under the current qualitative abstraction. This behavior is expected for faults that mainly affect quantitative magnitudes without producing a clear qualitative change in the observed variables. In the fault-level analysis, actuator-related faults and several structural faults are captured well, whereas an increased spring stiffness and some friction-related cases are more difficult.

The most challenging example is the heating benchmark. It is based on actual operational data and includes controller delays, operational variability, and regime shifts that the simplified qualitative model is unable to completely reflect, in contrast to the simulated benchmarks. The heating model treats mixed or inactive windows conservatively as non-conforming and uses distinct locked nominal models for the `pump1_only` and `pump2_only` regimes. The weak-label statistics reveal low recall (0.215) and moderate precision (0.789),

with many faulty windows obtaining conforming labels ($FN = 722$). This occurs because the valve faults do not always produce an immediate qualitative change in the observed pump state, flow, heating request or temperature variables.According to these findings, under the existing abstraction leaking-valve and stuck-valve faults frequently remain qualitatively close to nominal active heating behavior. This is plausible because the observed variables mainly capture coarse pump-regime, flow, request, and temperature states. Valve faults may change the quantitative efficiency or timing of heat transfer without necessarily changing these variables enough to cross qualitative landmarks within a window. Consequently, leaking-valve and stuck-valve faults can remain qualitatively close to nominal active heating behavior under the current abstraction.

Overall, the results show that QSIM weak labels are useful but imperfect supervision signals. They are strongest when faults cause clear qualitative deviations from nominal behavior, as in the RC circuit and weaker when faults primarily induce subtle quantitative changes, as in the heating benchmark. This motivates the combination with ML: QSIM provides physically grounded labels without manual annotation, while the downstream classifier learns from the original quantitative window features and can capture numerical regularities not explicitly represented in the qualitative abstraction.

6.2 Fault Detection Accuracy

Table 4 summarizes the predictive performance across the four benchmarks under full supervision, QSIM-based weak supervision, and low-label supervision. We report both accuracy and Macro-F1, and focus primarily on Macro-F1 in the discussion because it gives equal weight to faulty and non-faulty classes and is therefore more informative under class imbalance.

Overall, the results show that QSIM-derived weak labels can provide a useful supervisory signal when manually labeled data are scarce although the strength of this signal varies across systems. The clearest case is the RC circuit benchmark. Here, weak supervision retains a substantial fraction of the fully supervised performance across all three classifiers especially for logistic regression, where weak QSIM supervision slightly exceeds the fully supervised baseline. This reflects the strong alignment between the qualitative RC model and the simulated quantitative behavior: most injected faults produce clear qualitative deviations and nominal behavior is consistently accepted by the model.

The bathtub benchmark is more challenging. Full supervision performs strongly, especially for Random Forest and Decision Tree classifiers but QSIM-based weak supervision is less accurate. This is consistent with the weak-label analysis in Table 3, where the bathtub model produced both false positives and false negatives. In this benchmark, several faults, particularly inflow-related faults remain qualitatively close to nominal behavior at the window level. Consequently, the QSIM weak labels are less discriminative for downstream learning.The LL-5 Decision Tree result further highlights the need for Macro-F1 under imbalance: despite high accuracy, the substantially lower Macro-F1 indicates that the model does not achieve balanced performance across both classes when trained from only five labels. It is also worth noting that the low-label setting can outperform QSIM weak supervision for this benchmark, especially for Decision Tree and Logistic Regression at larger label budgets. This suggests that a small number of accurate ground-truth labels may be more informative than a larger set of noisy QSIM-derived labels when the qualitative abstraction does not separate nominal and faulty behavior well.

For the Spring–Mass–Damper benchmark, QSIM-based weak supervision remains competitive with full supervision, but the effect depends on the classifier. Weak supervision is below full supervision for Random Forest, approximately matches or slightly exceeds full supervision

Table 4 Accuracy (Acc) and Macro F1 by supervision regime across all benchmarks. Full and Weak QSIM use all training samples; LL-k uses only k labeled samples.

	Full		Weak QSIM		LL-5		LL-10		LL-20	
Model	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
RC circuit (train: 1242, test: 594)										
Random Forest	0.965	**0.894**	0.889	0.759	0.226	0.224	0.485	0.431	0.800	0.594
Decision Tree	0.939	**0.849**	0.837	0.714	0.131	0.129	0.689	0.565	0.848	0.577
Logistic Regression	0.736	0.619	0.758	0.637	0.219	0.218	0.495	0.440	0.783	**0.660**
Bathtub (train: 4158, test: 1782)										
Random Forest	0.937	**0.853**	0.493	0.400	0.357	0.331	0.433	0.367	0.552	0.461
Decision Tree	0.888	**0.776**	0.385	0.339	0.852	0.460	0.659	0.470	0.574	0.486
Logistic Regression	0.540	0.473	0.477	0.391	0.447	0.382	0.478	0.392	0.586	**0.495**
Spring–mass–damper (train: 12474, test: 5346)										
Random Forest	0.869	**0.550**	0.698	0.491	0.438	0.364	0.558	0.429	0.361	0.340
Decision Tree	0.581	**0.490**	0.698	0.491	0.304	0.282	0.495	0.388	0.372	0.348
Logistic Regression	0.456	0.406	0.701	**0.493**	0.368	0.331	0.670	0.478	0.414	0.375
Heating (train: 1207 , test: 518)										
Random Forest	0.983	**0.983**	0.558	0.508	0.597	0.595	0.705	0.675	0.888	0.888
Decision Tree	0.888	**0.888**	0.573	0.538	0.446	0.433	0.542	0.445	0.788	0.787
Logistic Regression	0.927	0.927	0.577	**0.551**	0.687	0.686	0.672	0.668	0.807	**0.806**

for Decision Tree and clearly improves over full supervision for Logistic Regression. This is an important result because the system exhibits oscillatory continuous dynamics, making purely window-level classification difficult. The QSIM-derived labels appear to provide a useful qualitative smoothing of the dynamics: rather than relying only on local numerical values, they encode whether a window is compatible with the qualitative structure of nominal oscillatory behavior. At the same time, the moderate Macro-F1 values show that some faults remain close to nominal oscillation patterns and are therefore difficult to separate.

The heating benchmark is the most difficult real-world setting. Full supervision achieves high performance, indicating that the quantitative window features contain useful information about valve faults. In contrast, QSIM weak supervision is more conservative, with Macro-F1 between 0.508 and 0.551. This matches the weak-label results: the final heating model was designed to preserve nominal conformance over active pump-regime segments, and therefore many faulty windows remain conforming under the qualitative abstraction. Nevertheless, the weakly supervised classifiers still learn a non-trivial signal from the QSIM labels, despite the absence of manually supplied fault labels during training. Low-label supervision generally improves with larger label budgets, with LL-20 achieving strong performance for all classifiers.

Taken together, these results support the main hypothesis of the paper. QSIM-based weak supervision is not a replacement for full supervision when many accurate labels are available. Instead, it provides a physically grounded source of supervision in cold-start settings, where labeled fault data are limited or unavailable. Its effectiveness depends on how strongly faults alter the qualitative behavior of the system. When faults create clear qualitative deviations, as in the RC circuit, weak supervision is highly effective. When faults mainly induce subtle quantitative changes, as in the heating benchmark, the QSIM signal is weaker but still useful as an initial labeling mechanism.

Table 5 Unsupervised anomaly detection across benchmarks. Models are trained on the first 70% of each nominal trace and tested on the remaining nominal windows together with the faulty windows from the same configurations.

	RC circuit		Bathtub		Spring–mass–damper		Heating	
Model	Acc	F1	Acc	F1	Acc	F1	Acc	F1
Isolation Forest	0.325	0.269	0.386	0.312	0.348	0.289	0.419	0.417
One-Class SVM	0.840	0.544	0.826	0.555	0.703	0.461	0.786	0.722

6.3 Unsupervised Anomaly Detection

The results of the unsupervised anomaly-detection baselines are shown in Table 5. Only nominal windows are used to train the models in this configuration, and they are assessed using both held-out nominal windows and faulty windows from the same benchmark configuration. This is consistent with a frequent cold-start assumption: nominal data may be available during early system operation or commissioning, while labeled fault examples are not.

One-Class SVM consistently performs better than Isolation Forest across all benchmarks, especially with respect to Macro-F1. This suggests that, for these window-level feature distributions a boundary based description of nominal behavior is more effective than Isolation Forest's isolation-based criterion. One-Class SVM performs strongly on the bathtub and heating benchmarks, where many faulty windows appear to deviate from the nominal feature distribution, while QSIM weak supervision is more competitive on the RC circuit and Spring–Mass–Damper benchmarks. This indicates that the two approaches capture different aspects of the data: unsupervised anomaly detection identifies distributional deviations from nominal examples, whereas QSIM-based weak supervision encodes qualitative physical consistency with a nominal behavioral model.

The Spring–Mass–Damper benchmark remains difficult for both approaches, most likely because oscillatory and regime dependent behavior creates substantial overlap between nominal and faulty windows. The heating result should also be interpreted carefully because the real operational data contain structured nominal variability and some valve-related faults can remain near nominal under particular operating regimes. Overall, these results show that unsupervised anomaly detection is a competitive cold-start baseline but not a replacement for model-based weak supervision. The distinction between these baselines and supervised or QSIM-based weakly supervised models motivates the employment of qualitative conformance as an additional physically grounded supervision signal.

6.4 Cross-configuration Generalization

Table 6 reports cross-configuration generalization under full supervision for the simulated benchmarks. In the *Same* setting, classifiers are trained and tested on windows from the same parameter configuration using ground-truth labels and the standard temporal split. In the *Cross* setting, classifiers are trained using ground-truth labels from one parameter configuration and evaluated on held-out parameter configurations.Overall, the findings indicate a low level of cross-configuration transfer. Random Forest and Decision Tree models provide very good same-configuration performance for the bathtub benchmark. However, Macro-F1 significantly declines under cross-configuration testing. This suggests that when inflow, drain area or beginning volume vary, these models learn configuration specific numerical patterns that are poorly transferable. The Spring-Mass-Damper benchmark exhibits a similar pattern: Macro-F1 remains low, indicating that the classifiers do not consistently distinguish

Table 6 Cross-configuration generalization across benchmarks. "Same" trains and tests on the same parameter configuration; "Cross" tests on held-out configurations. Macro F1 drops sharply under Cross across all benchmarks, indicating limited transferability.

		RC		Bathtub		Spring	
Model	Setting	Acc	F1	Acc	F1	Acc	F1
Random Forest	Same	0.909	0.476	0.999	0.997	0.882	0.514
	Cross	0.909	0.476	0.898	0.511	0.894	0.477
Decision Tree	Same	0.173	0.162	0.997	0.992	0.850	0.514
	Cross	0.164	0.154	0.898	0.511	0.885	0.481
Logistic Regression	Same	0.315	0.238	0.693	0.590	0.576	0.482
	Cross	0.304	0.237	0.719	0.453	0.696	0.470

between nominal and defective behavior across several oscillatory regimes, while accuracy remains quite high in some cross-configuration scenarios. Additionally, the RC circuit benchmark exhibits weak Macro-F1 in both settings suggesting that it is challenging to capture the specific trace-level or phase-dependent structure of this benchmark with a straightforward cross-configuration window-level classifier.

The heating data do not offer numerous systematically varied parameter configurations such as various resistances, inflows, or spring constants, relative to the simulated benchmarks. Therefore, there is no direct analogue of training on one parameter configuration and testing on held-out configurations.

7 Conclusion

Our initial study explores whether qualitative simulation can serve as a weak supervision source for fault detection in CPS. We propose a framework that exploits qualitative conformance checking to generate binary pseudo-labels for observed traces, which are then used to train a downstream binary classifier. With this approach, we address the cold-start problem by replacing manually labeled training data with annotation derived from a QSIM model of the nominal behavior. To determine the feasibility of our approach, we evaluate the technique on four benchmarks and compared our method to full supervision, unsupervised anomaly detection, supervised learning with limited labels, and transferring of a supervised learning model to another system configuration. Across the benchmarks our method was able to produce labels, although the quality of the resulting weak labels varied with the discriminative structure of the underlying qualitative model. On systems, such as the RC circuit, with clearly distinguishable phases, our trained classifier approached the performance of the fully supervised method. In the other case, the weak labels were less informative and thus our approach could not outperform the fully supervised classifier.

A natural question is why a classifier is needed at all if qualitative conformance already yields a label and thus could be used directly for fault identification. Conformance checking is costly, scales poorly with model size (and possibly model number in case fault models are also considered), and is less intuitive to deploy. This makes the QSIM approach less suitable for continuous online monitoring. An ML classifier is trained once, inference then is efficient and independent of the foundational qualitative model. Yet, the classifier still benefits from the physical grounding of the qualitative model. Additionally, beyond runtime efficiency, the trained classifier captures the concrete behavior of the system rather than

only its qualitative abstraction. Hence, once trained on QSIM-derived labels, the classifier may detect faults (e.g., large deviations between the observed and the concrete behavior), even when such faults remain invisible to the QSIM model itself. However, the classifier may also learn spurious quantitative patterns that the qualitative model cannot detect, e.g., unrepresentative operating regimes.

Our approach is subject to several limitations. First, it presumes the availability of a QSIM model of the nominal behavior. While qualitative models are generally less demanding to construct than precise numerical ones, they still require domain expertise. Hence, in domains where this expert knowledge is scarce, the development of the model might be a barrier. Second, the qualitative model must adequately capture the admissible nominal behavior. On the one hand, if the model is too permissive, faulty traces are not detected. On the other hand, if it is too restrictive, nominal traces may be rejected. Our initial study has shown that achieving an appropriate level of discriminative rigor can be challenging, particularly for systems with multiple operating regimes. Thus, the quality of the resulting pseudo-labels is dependent on the fidelity of the qualitative model.

Several directions for future research remain. First, we currently only consider fault detection, thus extending this to multi-classification is the next logical step. This would require richer labels, which we initially plan to generate by considering qualitative models that encompass typical system faults. Second it would be to formalize the weak-supervision setting more explicitly. Sensitivity to window length, stride, abstraction thresholds and aggregation techniques should be particularly studied in future research, along with uncertainty estimates for labels generated using QSIM. Third, to counteract the weak label quality on some systems we plan on combining symbolic supervision with self-supervised learning. The classifier first learns a representation from unlabeled operational data and qualitative conformance labels are then used only for fine-tuning the fault-detection head.

References

1 Bernhard K. Aichernig, Harald Brandl, and Franz Wotawa. Conformance testing of hybrid systems with qualitative reasoning models. *Electronic Notes in Theoretical Computer Science*, 253(2):53–69, 2009. doi:10.1016/j.entcs.2009.09.051.

2 Ivan Bellanco, Elena Fuentes, Manel Vallès, and Jaume Salom. A review of the fault behavior of heat pumps and measurements, detection and diagnosis methods including virtual sensors. *Journal of Building Engineering*, 39:102254, 2021. doi:10.1016/j.jobe.2021.102254.

3 George M. Coghill, Ashwin Srinivasan, and Ross D. King. Qualitative system identification from imperfect data. *Journal of Artificial Intelligence Research*, 32:825–877, 2008. doi:10.1613/jair.2374.

4 Marco Cook, Cory Paterson, Angelos K Marnerides, and Dimitrios Pezaros. Anomaly diagnosis in cyber-physical systems. In *ICC 2022-IEEE International Conference on Communications*, pages 5445–5450. IEEE, 2022. doi:10.1109/ICC45855.2022.9838968.

5 Ankita Das, Roxane Koitz-Hristov, and Franz Wotawa. Using qualitative simulation models for monitoring and diagnosis. In Marcos Quiñones-Grueiro, Gautam Biswas, and Ingo Pill, editors, *36th International Conference on Principles of Diagnosis and Resilient Systems, DX 2025, Nashville, TN, USA, September 22-24, 2025*, volume 136 of *OASIcs*, pages 4:1–4:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/OASIcs.DX.2025.4.

6 Ankita Das, Roxane Koitz-Hristov, and Franz Wotawa. Towards parameter-independent fault detection using qualitative simulation. Under review for Qualitative Reasoning 2026 at IJCAI, Bremen, Germany, 2026.

7 Swagat Das, Devin Drake, Harindra S. Mavikumbure, Victor Cobilean, and Milos Manic. Learning beyond labels: Self-supervised methods for anomaly detection in cyber-physical systems. In *IECON 2025 – 51st Annual Conference of the IEEE Industrial Electronics Society*, pages 1–8, 2025. doi:10.1109/IECON58223.2025.11221272.

8 Johan De Kleer and John Seely Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24(1):7–83, 1984.

9 Johan de Kleer and John Seely Brown. A qualitative physics based on confluences. *Artificial Intelligence*, 24(1-3):7–83, 1984. doi:10.1016/0004-3702(84)90037-7.

10 Hongwen Dou and Kun Zhang. Transfer learning for cross-building forecasting of building energy and indoor air temperature in model predictive control applications. *Journal of Building Engineering*, 111:113341, 2025. doi:10.1016/j.jobe.2025.113341.

11 Kenneth D. Forbus. Qualitative process theory. *Artificial Intelligence*, 24(1-3):85–168, December 1984. doi:10.1016/0004-3702(84)90038-9.

12 Peter Fritzson. Modelica - a language for equation-based physical modeling and high performance simulation. In *Proceedings of the 4th International Workshop on Applied Parallel Computing, Large Scale Scientific and Industrial Problems*, PARA '98, pages 149–160, London, UK, UK, 1998. Springer-Verlag. doi:10.1007/BFB0095332.

13 Mikhail Genkin and J. J. McArthur. Using reinforcement learning with transfer learning to overcome smart building cold start. In *International Conference on Computational Science and Computational Intelligence, CSCI 2022, Las Vegas, NV, USA, December 14-16, 2022*, pages 713–718. IEEE, 2022. doi:10.1109/CSCI58124.2022.00130.

14 Jie Gui, Tuo Chen, Jing Zhang, Qiong Cao, Zhenan Sun, Hao Luo, and Dacheng Tao. A survey on self-supervised learning: Algorithms, applications, and future trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):9052–9071, 2024. doi:10.1109/TPAMI.2024.3415112.

15 Srinivas Katipamula and Michael R. Brambley. Review article: Methods for fault detection, diagnostics, and prognostics for building systems—a review, part i. *HVAC&R Research*, 11(1):3–25, 2005. doi:10.1080/10789669.2005.10391123.

16 Roxane Koitz-Hristov, Liliana Marie Prikler, and Franz Wotawa. Beyond Static Diagnosis: A Temporal ASP Framework for HVAC Fault Detection. In Marcos Quinones-Grueiro, Gautam Biswas, and Ingo Pill, editors, *36th International Conference on Principles of Diagnosis and Resilient Systems (DX 2025)*, volume 136 of *Open Access Series in Informatics (OASIcs)*, pages 1:1–1:20, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/OASIcs.DX.2025.1.

17 Benjamin Kuipers. The limits of qualitative simulation. In Aravind K. Joshi, editor, *Proceedings of the 9th International Joint Conference on Artificial Intelligence. Los Angeles, CA, USA, August 1985*, pages 128–136. Morgan Kaufmann, 1985. URL: http://ijcai.org/Proceedings/85-1/Papers/023.pdf.

18 Benjamin Kuipers. Qualitative reasoning: Modeling and simulation with incomplete knowledge. *Automatica*, 25(4):571–585, 1989. doi:10.1016/0005-1098(89)90099-X.

19 Benjamin J. Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–338, 1986. doi:10.1016/0004-3702(86)90073-1.

20 Jie Lu, Anjin Liu, Fan Dong, Feng Gu, João Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019. doi:10.1109/TKDE.2018.2876857.

21 Yuan Luo, Ya Xiao, Long Cheng, Guojun Peng, and Danfeng (Daphne) Yao. Deep learning-based anomaly detection in cyber-physical systems: Progress and opportunities. *ACM Comput. Surv.*, 54(5), May 2021. doi:10.1145/3453155.

22 Antonio M. Martínez-Heredia and Sebastián Ventura. Weak supervision: A survey on predictive maintenance. *WIREs Data Mining and Knowledge Discovery*, 15(2):e70022, 2025. doi:10.1002/widm.70022.

23 Morteza Mohaqeqi, Mohammad Reza Mousavi, and Walid Taha. Conformance testing of cyber-physical systems: A comparative study. *Electron. Commun. Eur. Assoc. Softw. Sci. Technol.*, 70, 2014. doi:10.14279/TUJ.ECEASST.70.982.

24 Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. Snorkel: Rapid training data creation with weak supervision. *The VLDB Journal*, 29(2):709–730, 2020. doi:10.1007/s00778-019-00552-1.

25 Jagath Sri Lal Senanayaka, Huynh Van Khang, and Kjell G. Robbersmyr. Toward self-supervised feature learning for online diagnosis of multiple faults in electric powertrains. *IEEE Transactions on Industrial Informatics*, 17(6):3772–3781, 2021. doi:10.1109/TII.2020.3014422.

26 Maxim Shcherbakov and Cuong Sai. A hybrid deep learning framework for intelligent predictive maintenance of cyber-physical systems. *ACM Trans. Cyber-Phys. Syst.*, 6(2), May 2022. doi:10.1145/3486252.

27 Georgios Tertytchny, Nicolas Nicolaou, and Maria K. Michael. Classifying network abnormalities into faults and attacks in iot-based cyber physical systems using machine learning. *Microprocessors and Microsystems*, 77:103121, 2020. doi:10.1016/j.micpro.2020.103121.

28 Louise Travé-Massuyès and Robert Milne. Application oriented qualitative reasoning. *The Knowledge Engineering Review*, 10(2):181–204, 1995. doi:10.1017/S0269888900008146.

29 Sven Werner. International review of district heating and cooling. *Energy*, 137:617–631, 2017. doi:10.1016/j.energy.2017.04.045.

30 Timothy Wiley, Claude Sammut, and Ivan Bratko. Qualitative simulation with answer set programming. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, pages 915–920, 2014. doi:10.3233/978-1-61499-419-0-915.

31 Shichao Xu, Yixuan Wang, Yanzhi Wang, Zheng O'Neill, and Qi Zhu. One for many: Transfer learning for building hvac control. In *Proceedings of the 7th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*, BuildSys '20, pages 230–239, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3408308.3427617.

32 Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. doi:10.1609/aaai.v37i9.26317.

33 Zhi-Hua Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, January 2018. doi:10.1093/nsr/nwx106.