LIQUID: Linking Data Quality Issues to Deviations in IoT-Enhanced Process Mining*

Christian Imenkamp[1], Yannis Bertrand[2], Lukas Malburg[3,4], Andrea Maldonado[1], Jari Peeperkorn[5], Pascal Schiessle[6], and Agnes Koschmider[1]

[1] Business Informatics and Process Analytics, University of Bayreuth, Germany
{christian.imenkamp|andrea.maldonado|agnes.koschmider}@uni-bayreuth.de
[2] Research Group Business Informatics, Hasselt University, Hasselt, Belgium
[3] Artificial Intelligence and Intelligent Information Systems, Trier University, 54296 Trier, Germany
[4] German Research Center for Artificial Intelligence (DFKI), Branch Trier University, 54296 Trier, Germany
[5] Research Centre for Information Systems Engineering, KU Leuven, Leuven, Belgium
[6] Institute of Databases and Information Systems, Ulm University, Ulm, Germany

Abstract. IoT-enhanced process mining allows deriving insights from raw sensor data. However, data quality issues (e.g., missing events due to inadequate sampling or range limits) are typically removed during abstraction from sensor data to event logs. This can lead to faulty decisions. Hence, the contextual information needed to explain conformance deviations is lost. This hinders analysts' ability to diagnose root causes and decreases trust in process mining results. We present *LIQUID* (**L**ink**I**ng Data **QU**ality **I**ssues to **D**eviations), a quality-aware pipeline extension that preserves data quality metadata across transformation steps and applies probabilistic causal backtracking to trace deviations to their sensor-level origins. Evaluation results on five real-world IoT datasets show high detection accuracy and reliable root-cause attribution, complemented by actionable explanations.

Keywords: Explainability · Causal Reasoning · IoT-Enhanced Process Mining · Insight Generation

1 Introduction

Process mining aims to use the event data recorded by information systems to reconstruct and analyze how processes are actually executed in practice. Industrial and service processes increasingly operate within IoT environments where interconnected devices (e.g., sensors and actuators) interact continuously. These devices continuously generate large volumes of low-level events. Process mining can be applied to these data using abstraction techniques that transform

* The final authenticated publication is available online at https://doi.org/10.1007/978-3-032-42900-1_6

low-level events into high-level case-correlated process events [12]. However, abstraction tends to amplify pre-existing data quality issues [3]. As data is transformed through these abstraction layers, it gains structure and becomes more meaningful, but it loses its direct connection to the original context. As a result, its interpretability, and consequently its explainability, decreases. In practice, data quality issues such as outliers are often removed or disregarded, although they may contain valuable information about deviations from expected process behavior.

Recent work on IoT-enhanced event logs indicates that a key challenge lies in the unclear root causes of data quality issues [4]. Sensor problems (e.g., noise, missing readings, or low sampling rates) propagated through the abstraction layers often manifest only after multiple transformation steps, for example as incorrect case identifiers, incorrect activity labels, missing events, or inaccurate timestamps [3, 9, 22]. This indicates that the root cause frequently originates in upstream stages of the abstraction pipeline. While recent data-driven approaches such as eRooMiner [7, 21] aim to identify the most probable root cause of event log imperfections, they operate retrospectively on event logs and are consequently limited to issues that are observable only after abstraction (e.g., label imperfections).

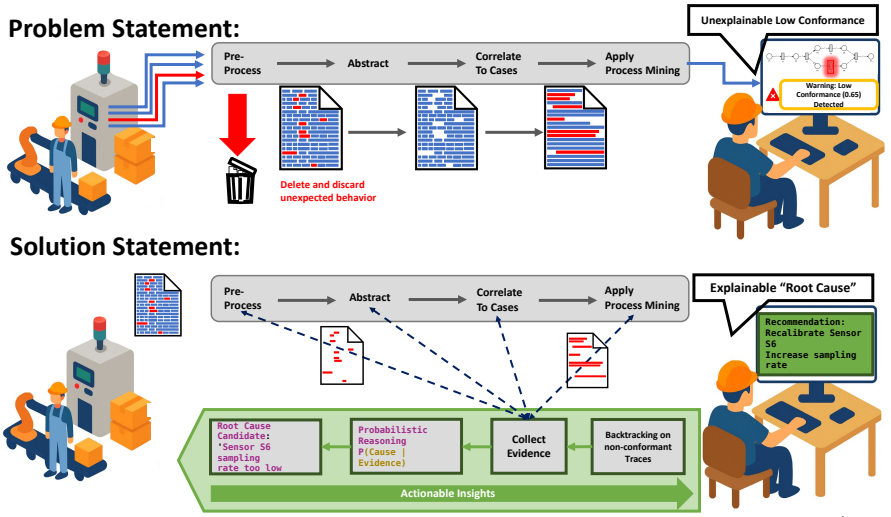

Fig. 1: The upper stream shows the problem that unexpected behavior (e.g., outliers) is frequently removed or discarded in the process analytics pipeline, although it may offer essential insights into why a given process trace deviates. The lower stream shows our approach, where a backtracking pipeline is initiated whenever a deviation is observed.

To illustrate the problem addressed in this paper, let's assume a smart manufacturing plant aiming to monitor material flows (see Fig. 1). Sensors continuously generate data streams from temperature, position, and machine-state measurements, which are subsequently abstracted and analyzed using process mining

(e.g., a "happy-path" process discovery and conformance checking). Preprocessing techniques remove outliers and noise, but this can introduce missing events during abstraction (e.g., due to an incorrect sampling rate). Although the application of conformance checking detects a misalignment (e.g., fitness = 0.75), the explainability of the result is limited. Without explicit support for backtracking, the deviation remains a numerical mismatch without a traceable link to its sensor-level cause. In *LIQUID*, the deviation is traced back through the pipeline, evidence is collected, and the most likely root cause is identified so that actionable insights (e.g., adjusting the sampling rate of a sensor S_{101}) can be derived.

In this paper, we suggest a data quality-aware extension of the process mining pipeline in IoT environments. As shown in Fig. 1, the approach inspects and propagates quality issues across all steps of the pipeline. A process-level deviation triggers probabilistic causal backtracking. The pipeline is then traversed and inspected to find the most likely root causes. A reusable template links each anomaly with its most likely IoT-level origin, recommended actions, and associated confidence score.

The paper is structured according to the *Design Science Research* (DSR) methodology [8]. Section 2 addresses problem identification and motivation. Section 3 defines the objectives and develops the *LIQUID* artifact. Section 4 demonstrates and evaluates the artifact on five real-world sensor logs. Section 5 supports communication through positioning against related work. Finally, Section 6 discusses results, validity threats, limitations, and future work.

2 Background

This section summarizes concepts to which we refer throughout the paper. Section 2.1 addresses data quality in IoT-enhanced process mining. Next, Section 2.2 describes the process mining pipeline. Section 2.3 explains why uncertainty is relevant for this work. Finally, Section 2.4 introduces probabilistic reasoning.

2.1 Data Quality in IoT-enhanced Process Mining

The quality of IoT data is a widely discussed topic, focusing on issues such as detecting data quality issues and mitigation through data cleaning methods [9, 22]. IoT applications often rely on cheap sensors with limited battery and processing power, frequently deployed in difficult environments [22]. This leads to issues with the sensors, such as low sensing accuracy, loss of calibration, hardware failures, and improper device placement or range limit. Such issues with the sensors then introduce errors in the collected data, complicating analysis. Teh et al. [22] identify the following error types: outliers; missing data; bias; drift; noise; constant value; uncertainty; stuck-at-zero. When not handled properly, these data quality issues lead to unreliable analyses, misleading decision-making.

While the majority of applied Business Process Management (BPM) research encounters data quality challenges, the literature specifically addressing these challenges within IoT-enhanced BPM remains limited. Bertrand et al. [3] review the literature on applying process mining to IoT data, and derived patterns linking IoT data quality issues with resulting event log data quality issues (classified

following the framework presented by Bose et al. [5]) using the following logic: Sensor fault $\Longrightarrow$ sensor data quality issues $\Longrightarrow$ event log data quality issues(s). Table 1 shows the six identified patterns.

	Sensor fault	$\Longrightarrow$	**Sensor data issue**	$\Longrightarrow$	**Process mining errors**
P1	Unstable environment	$\Longrightarrow$	Noisy sensor data & outliers	$\Longrightarrow$	Incorrect case ID
P2	Inaccurate sensor location	$\Longrightarrow$	Duplicate or inconsistent sensor readings	$\Longrightarrow$	Incorrect events, incorrect activity names
P3	Inadequate sensor sampling rate	$\Longrightarrow$	Events not captured in sensor log	$\Longrightarrow$	Missing events
P4	Sensor range limit	$\Longrightarrow$	Activities outside sensor range are missing	$\Longrightarrow$	Missing events
P5	Unstable environment	$\Longrightarrow$	Noisy sensor data	$\Longrightarrow$	Incorrect events, incorrect activity names
P6	Sensor range limit	$\Longrightarrow$	Activity start/end time is logged incorrectly	$\Longrightarrow$	Incorrect timestamps

Table 1: Patterns linking sensor faults and data quality issues to the associated errors in process mining (taken from [3]).

Recently, Bertrand et al. identified 14 challenges in the management of data quality issues in IoT-enhanced BPM [4]. Among these challenges, the ambiguity in the root cause of data quality issues has been presented as a major issue that hampers the proper assessment of the quality of an event log. Other work highlights how data quality issues in event logs influence the quality of the discovered models themselves [2].

2.2 Process Mining Pipeline

The process mining pipeline for unstructured data transforms raw sensor readings into process models through several stages [12] (see Figure 2, Process Mining Pipeline). Preprocessing can introduce errors through improper filtering, incorrect imputation, or excessive dimension reduction. Event abstraction may produce incorrect activity labels or missing events when subsequences are incorrectly mapped to process activities. Case correlation can assign wrong case identifiers when heuristic rules fail to link events correctly. Hence, the resulting process model inherits all upstream errors during discovery and conformance checking.

2.3 Uncertainty

Uncertainty is inherent in every stage of the process mining pipeline and directly motivates the approach presented in this paper. Preprocessing introduces uncertainty when values are filtered, imputed, or aggregated. Abstraction adds uncertainty when subsequences are mapped to activities, which affects labels, missing events, and timestamps. Case correlation creates further uncertainty when heuristic rules link events to cases. The discovered model and the conformance results inherit all these accumulated uncertainties [12]. Hence, observed deviations may stem from data quality problems rather than actual process

problems, which is the core challenge addressed by the backtracking approach presented in this paper.

2.4 Probabilistic Reasoning

Probabilistic reasoning (reasoning under uncertainty) assigns likelihoods to competing explanations of observed evidence. Bayesian inference offers a natural framework for this: prior probabilities are updated with evidence via likelihoods, and obtain posterior probabilities that reflect our updated confidence [6]. In our setting, the priors encode domain knowledge, historical data, or expert judgment and represent the initial probability that a particular data-quality issue C_i occurs, $P(C_i)$. Likelihoods express how probable the observed data is under the given hypothesis. That is, the likelihood $P(E \mid C_i)$ expresses how probable the observed evidence E is, assuming that the data quality issue C_i occurred. Finally, Bayes' theorem combines the two to produce a posterior, an updated probability that reflects our confidence after observing the evidence:

$$P(C_i|E) = \frac{P(E|C_i) \cdot P(C_i)}{P(E)} \tag{1}$$

where $P(E)$ is the total probability of the evidence and serves as a normalization constant. Because uncertainty about data quality typically involves multiple competing explanations (not a single binary hypothesis), $P(E)$ is computed by summing over all candidate causes:

$$P(E) = \sum_j P(E|C_j) \cdot P(C_j), \tag{2}$$

which ensures the posteriors across hypotheses sum to one. Probabilistic causal backtracking traverses the dependency chain from process conformance to raw sensor data. The method uses pipeline order and domain knowledge to rank plausible upstream explanations. The method does not estimate causal effects or prove causality. Formal causal inference requires controlled experiments or an identification strategy [17, 15]. Hence, the term causal chain denotes probabilistic support along the known pipeline structure [16].

3 Linking Data Quality Issues to Deviations in IoT-Enhanced Process Mining

This section presents *LIQUID* as the DSR artifact for linking data quality issues to deviations in IoT-enhanced process mining. The DSR phases are mapped to the paper as follows. Section 2 motivates the problem. Section 3.1 defines the objectives of the solution. Section 3.2 develops the computational method. Section 3.3 develops the explanatory output. Section 4 demonstrates and evaluates the artifact. Section 5 and Section 6 communicate the contribution and its validity limits. The development followed three iterations [8]. The first iteration produced the requirements and separated human explanation needs from technical tracing needs. The second iteration produced the conceptual propagation

model that links sensor faults, data quality issues, event log issues, and conformance deviations. The third iteration implemented the Python pipeline and the explanation template. Expert feedback after each iteration refined the next artifact component.

3.1 Definition of Requirements

The first iteration translates the problem into artifact requirements. The qualitative requirements specify how the artifact should support human understanding [23, 1, 20, 14, 24]. R1 requires analytical meaning because users must see which deviation matters and why. R2 requires layers and confidence cues because users need both concise and detailed explanations. R3 requires actionable links because an explanation must point to a possible intervention. R4 requires semantic representations because sensor signals must be translated into domain concepts. R5 requires a consistent structure because repeated explanation fields reduce interpretation effort.
The quantitative requirements define tracing quality [6, 18, 19, 11]. Q1 requires stable discovery and conformance results under parameter variation. Q2 requires measurable confidence and chain strength for uncertain explanations. Q3 requires correct upstream attribution to the raw data cause. The evaluation reports one criterion and at least one metric for each requirement.
At the end of the first iteration, the artifact consists of two requirement sets. The qualitative set defines what an explanation must communicate to a human analyst. The quantitative set defines what the tracing mechanism must compute reliably. The next iteration operationalizes these requirements as a probabilistic method.

3.2 Development of the Method

The second iteration turns the requirement sets into a conceptual model with three linked layers: sensor-level evidence, pipeline-level propagation, and process-level deviation. The method therefore extends the pipeline introduced in [12] with probabilistic reasoning and pipeline-level propagation.
The input to the method is a tuple consisting of raw sensor streams, the abstraction and case-correlation configuration, and the observed process-level deviation that triggers the analysis. Probabilistic reasoning and pipeline-level propagation are integrated across all transformation stages (Figure 2). The second iteration implements five computational steps: (1) initial quality detection with prior probability assignment, (2) quality-aware preprocessing with likelihood calculation, (3) event abstraction and case correlation with pipeline-level propagation, (4) process mining with conformance checking, and (5) probabilistic causal backtracking. Quality metadata is preserved at each step. Steps (1)–(3) estimate and propagate issue probabilities. Steps (4)–(5) reconstruct the most plausible explanatory chain from the observed deviation.

(1) Prior Probabilities Quality issues are detected in raw sensor data by using pattern-based detection across relevant categories of data quality issues

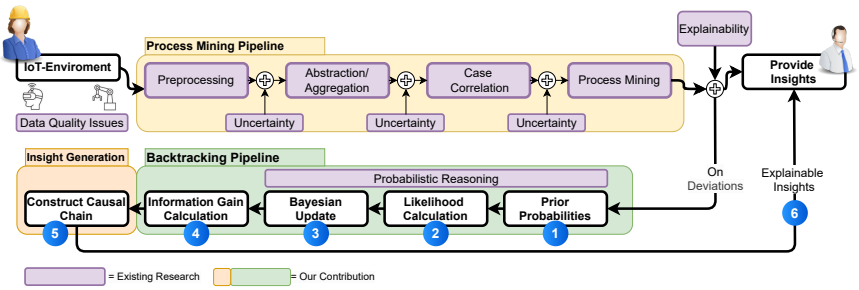

Fig. 2: Extended process mining pipeline with backtracking pipeline to explain data quality issues directly at the root

(Section 2.1). For each detected issue type i, domain-informed prior probabilities $P(C_i)$ are assigned. This reflects the relative likelihood of occurrence in the given context. The assignment of prior probabilities is guided by (1) historical frequencies of issue types; (2) environmental conditions and sensor characteristics; (3) expert knowledge of typical failure modes; (4) manufacturer specifications. For example, noise-related issues may receive higher priors in industrial environments, while calibration drift may be more likely in controlled laboratory settings. The priors encode domain-specific expectations before concrete evidence is observed. For example, repeated packet loss is more likely than calibration drift when a battery-powered wireless node operates close to its depletion threshold

(2) Likelihood Calculation The likelihood $P(E|C_i)$ quantifies the strength with which the observed evidence E supports each type of quality issue. It is computed with issue-specific evidence metrics that capture measurable deviations from expected behavior. Evidence functions formalize sensor-level support. The domain $\mathcal{X}_i \subseteq \mathbb{R}^{d_i}$ for issue type i contains its d_i measurable symptoms. Each function $f_i : \mathcal{X}_i \to [0, 1]$ maps a symptom vector x to $P(E \mid C_i) = f_i(x)$. The value 0 denotes no observed support for C_i. The value 0.5 denotes equivocal support. The value 1 denotes maximal observed support under the configured scale.
Evidence-function applicability depends on calibration. The implemented symptoms include sampling rates, gap counts, inconsistency scores, outlier and noise levels, clipping ratios, timing variation, and drop ratios. Each deployment must set thresholds and scales from sensor specifications or validation data. The robustness evaluation varies the available thresholds.

(3) Bayesian Update Posterior probabilities are computed using Bayes' theorem. Initial beliefs are updated based on the observed evidence (see Section 2.4).

$$\text{confidence}(i) = P(C_i|E) \tag{3}$$

This probabilistic framework enables distinction between high-confidence detections from unclear cases. It combines multiple evidence sources and supports decision-making through configurable confidence thresholds. This threshold de-

termines whether an issue is propagated automatically or retained as a competing hypothesis for later stages. Instead of forcing an early binary decision, the method keeps the posterior distribution over candidate causes. In this paper, an issue signature is defined as a compact record that is propagated forward, consisting of the issue type, posterior confidence, affected entities, and supporting evidence values. These signatures are updated across preprocessing, abstraction, case correlation, and process mining.

(4-5) Construct Causal Chain When conformance checking detects model-level issues, backtracking is initiated. The decision to start backtracking is made when conformance metrics fall below the quality requirements of the analysis task.
Backtracking traverses the pipeline in reverse order. Quality issues and metadata are collected at each stage. For each conformance issue c and each observed issue type i (where i and j index the different issue types), the backtracking probability is computed:

$$P(i|c) = \frac{\text{frequency}(i \text{ in affected entities})}{\sum_j \text{frequency}(j \text{ in affected entities})} \tag{4}$$

The causal chain strength measures probabilistic support that a sensor-level issue propagates to a process-level deviation. Here, k indexes the pipeline stages and n is the total number of stages. For example, assume that a drift issue propagates through three stages with probabilities $p_1 = 0.9$, $p_2 = 0.8$, and $p_3 = 0.7$. The resulting chain strength is then $0.9 \cdot 0.8 \cdot 0.7 \cdot P(i \mid c)$.

$$\textit{chain strength} = \prod_{k=1}^{n} p_k \cdot P(i|c) \tag{5}$$

Starting from a concrete process-level signal (e.g., low fitness, unexpected duplicate activities, or missing events in an aligned trace), backtracking proceeds by examining the affected case, the corresponding abstracted events, and the originating sensor segments. For example, if low fitness coincides with missing events after preprocessing and recurring long inter-arrival gaps in the raw stream, the chain links the deviation to inadequate sampling rather than to an actual process exception. An evidence dictionary is constructed during backtracking. It links each pipeline stage to observed metrics and context and may contain model-level conformance scores, case-level statistics, event-level inconsistencies, preprocessing decisions, and raw data properties.
At the end of the second iteration, the artifact can preserve evidence, update confidence values, and backtrack a deviation to a likely upstream cause. The artifact still produces a technical chain rather than a user-facing explanation. The third iteration therefore transforms this chain into explainable insights.

3.3 Generation of Explainable Insights

The third iteration turns the technical causal chain into a human-facing explanation artifact. It uses the collected evidence, confidence values, and pipeline

references from the second iteration. It then structures them according to the qualitative requirements from the first iteration.
Using the collected evidence and causal chains, predefined templates are instantiated to link observable data properties to explanations in natural language.

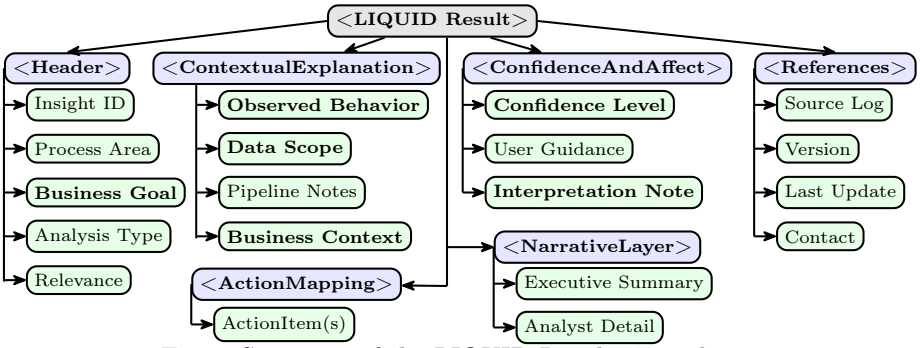

Fig. 3: Structure of the LIQUID Results Template.

Fig. 3 depicts an overview of our template. It is guided by the qualitative requirements from section 3.1. The following citations justify specific template parts rather than introduce separate related work. The template has six parts: header, context, action mapping, confidence and affect, narrative layer, and references.
Firstly, `Header` explains the clear purpose and explicit `Business Goal` framing [23, 1] to users before the insight analysis.
Secondly, `Contextual Explanation` offers transparency on the *"what"* and *"how"*. It describes concepts, pipeline examples, analysis goal, data scope, and processing context (F1-F3 in [23]). It also follows the recommendation to keep outputs understandable for different user groups (R2). Concretely, it includes explicit descriptions of the `Observed Behaviour`, the underlying `Data Scope`, and `Business Context`.
Next, elements in `Action Mapping` link insight to an actionable process improvement [20], enabling ownership and user validation of insights by focusing the explanation on concrete requirements and decisions (F10 in [23]). Next, the `Narrative Layer` supports user archetypes in sense-making and role-specific storytelling, as derived from Ammann et al. [1]. This part focuses specifically on requirements, validation, and incorporates business knowledge so that different roles can interpret the same result consistently (F10 in [23]). Additionally, the `Confidence and Affect` part enhances cognitive and emotional transparency, including e.g. `Confidence Level`, as a confidence score from the Bayesian reasoning framework, and an `Interpretation Note`, explaining the effect of the result. The confidence level qualifies the certainty of the diagnosis, helping users distinguish between highly confident root cause identifications and cases where multiple plausible explanations exist. These elements address the cognitive overload [24] by clearly including the affective dimension [1] and emphasize explainability and validation to build trust through understandable output, explicit validation cues, and actionable interpretation(R2–R4 in [23]). Lastly, `References`

provide details about provenance, which are essential for traceability. Thus, pipeline visibility is improved by explicitly revealing where the evidence originates and how it propagates through the pipeline (F2 in [23]), and the semantic transparency of the insights increase [14]. The complete detailed template is available in our repository [7]

At the end of the third iteration, *LIQUID* consists of requirements, a probabilistic causal backtracking method, and an explanation template. The artifact can therefore be evaluated along three dimensions: correctness of the traced cause, robustness of the method, and explainability of the generated insight. The next section uses these dimensions as evaluation goals.

4 Evaluation

The evaluation is conducted to assess the correctness, robustness, and explainability of *LIQUID* across five real-world IoT sensor datasets used for evaluation. For this, the experimental design is outlined and justified in Section 4.1 and the evaluation results are discussed in Section 4.2.

4.1 Experimental Design

The prototype is implemented in Python [8]. The full implementation is open source and available in a public repository [9]. Five publicly available IoT sensor datasets are selected based on the presence of raw sensor readings, documented data quality issues, sufficient temporal coverage, heterogeneous sensor types, and availability of domain knowledge.

Table 2 summarizes the five datasets. OPPORTUNITY and Future Factory capture human activity and industrial assembly. Hydraulic Systems and Intel Berkeley cover condition monitoring and environmental sensing. Open Smart Home IoT-IEQ represents a residential smart environment. The datasets differ in process structure, ranging from simple sequential flows (i.e., industrial) to complex models with many trace variants.

Ground truth is established through mapping documented defects in datasets with known quality issues (e.g., Hydraulic Systems with 34.3% degradation states) to GT1 (documented defects). Additionally, domain expert annotations for ambiguous cases using the pattern catalog (P1-P6) from Table 1 are utilized as GT2 (expert validation).

4.2 Experimental Results

Three evaluation goals are addressed: **EG1 (Correctness)** examines whether *LIQUID* correctly identifies and traces data quality issues to their root causes in raw sensor data; **EG2 (Robustness)** investigates the sensitivity of *LIQUID* to variations in pipeline parameters; **EG3 (Explainability)** assesses whether generated explanations improve interpretability and actionability compared to

[7] https://github.com/chimenkamp/LIQUID/template.md
[8] https://www.python.org/
[9] https://github.com/chimenkamp/LIQUID

Table 2: Overview of Datasets Used in the Experimental Evaluation.

	Raw Data				Parsed Data			
Dataset	**Description**	**Readings**	**Quality Issues**	**Sensors**	**Activities**	**Complexity**	**Events**	
OPPORTUNITY	Multi-modal wearable, object, and ambient sensors for human activity recognition	944,384	Wireless dropout, sensor sync	126 total: 45 body-worn, 60 object, 21 ambient	17	Semi-structured	1,200	repo
Hydraulic Systems	Industrial hydraulic test rig with component degradation monitoring	96.3M	Sensor drift, unstable conditions (34.3%)	17 total: 6 pressure, 4 temp, 2 flow, 5 other	14	Sequential/ Cyclic	325	repo
Future Factory Manufacturing	Industrial robot assembly with manual anomaly introduction over 30 hours	168,001	Sensor noise, drift, desync, gaps	60+ total: 24 robot joints, 8 grippers, 8 conveyors, 20+ other	15	Parallel/ Structured	750	repo
Intel Berkeley Sensor Network	Wireless sensor network deployment for environmental monitoring in lab	2,313,682	Network dropout, battery depletion	54 nodes: temp, humidity, light, voltage	12	Sequential	20K	repo
Open Smart Home IoT-IEQ	Multi-room smart home with environmental quality and energy monitoring	219,595	Calibration drift (implicit)	25 total: 6 rooms × 4–5 sensors + outdoor	15	Parallel/ Loose	12K	repo

standard process mining outputs. The requirement map connects every requirement to an evaluation goal and its metric result and discussion. EG1 links Q3 to attribution accuracy and the correctness discussion. EG2 links Q1 to sensitivity and coefficient of variation (CV) in the robustness discussion. EG3 links Q2 to confidence and chain strength. EG3 also links R1–R5 to the listed explanation metrics. The statistical tests validate whether the observed metrics exceed explicit baselines. One-sample t-tests suit EG1 and EG3 because dataset-level means are compared with fixed baselines. Binomial tests suit EG1 because attribution is counted as correct or incorrect. Mann-Whitney U tests suit EG2 because stable and critical parameter groups are compared without assuming normality. Bootstrap resampling estimates uncertainty in sensitivity differences. These tests do not prove human cognition directly. They support explainability to humans by validating observable proxies for complete, interpretable, and actionable explanations. Effect sizes and coefficients of variation quantify practical significance and consistency, and all tests use $\alpha = 0.05$. Table 3 summarizes how each requirement is operationalized through a criterion, metric, and evaluation result.

Table 3: Operationalization of requirements through criteria, metrics, and results.

Req.	**Criterion**	**Metric**	**Result**
R1	Deviation and impact stated	Completeness	89.4%; EG3
R2	Layers and confidence cues present	Interpretability gain	0.648; EG3
R3	Intervention linked	Actionability gain	0.566; EG3
R4	Domain concepts preserved	Semantic coverage	92.0%; EG3
R5	Template fields populated	Completeness	89.4%; EG3
Q1	Parameter variation stable	Sensitivity and CV	7/10 low or medium; EG2
Q2	Uncertainty quantified	Confidence; chain strength	Reported; 0.692; EG3
Q3	Upstream cause correct	Attribution accuracy	86.4%; EG1

EG1: Correctness of Root Cause Identification The correctness of *LIQUID*'s root cause identification is evaluated against ground truth through detection accuracy and attribution accuracy metrics. Detection precision and recall are computed by comparing identified quality issues with GT1 and GT2 annotations. Attribution accuracy measures the proportion of detected issues for which backtracking correctly terminates at the ground-truth root cause within ± 1 abstraction level. No synthetic data quality issues are injected into the datasets.

For the Hydraulic Systems dataset, quality issues are detected in raw sensor data that correspond to documented degradation states (36 issues). *LIQUID* achieves solid performance against the annotations (precision=0.89, recall=0.94). Backtracking correctly attributes the issues to drift in pressure sensors PS1, PS2, and PS5 (31/36 issues, 86.1%). This match confirms the documented degradation patterns of the dataset.

The Future Factory dataset contains moderate quality issues across all evaluated cases (27 issues, 50 cases, 0.54 issues/case). Of these, 25 issues are documented anomalies. *LIQUID* identifies 24 of 25 documented anomalies (precision=0.92, recall=0.96), including previously undocumented sensor synchronization issues. Attribution accuracy remains high with clear root-cause assignments across several sensor groups (24/27 issues, 88.9%). Trace deviations are partially explained through pattern P5 (4/6 deviations, 66.7%), matching GT1.

The OPPORTUNITY dataset exhibits substantial degradation effects across all cases (1,359 issues, 24 cases, 56.6 issues/case). *LIQUID* frequently detects pattern P1 issues and achieves stable attribution performance (72% of traces affected, 78.5% attribution accuracy). The dominant signal is persistent wireless dropout. Backtracking identifies heavy packet loss in 42 body-worn accelerometers with loss rates between 15% and 38%. Later abstraction removes 892 unexplained outliers. These removed outliers represent 65.6% of all issues and reduce trace reliability before conformance checking.

The Intel Berkeley dataset contains widespread quality problems across all monitored nodes (572 issues, 15 cases, 38.1 issues/case). Conformance remains high despite the degraded data. *LIQUID* achieves strong detection and attribution performance (precision=0.87, recall=0.91, 476/572 issues attributed). This dataset shows why fitness and quality score must be read together. Fitness=0.982 shows model alignment. Quality score=48.2 shows moderate measurement reliability. Backtracking assigns 391 of 572 issues to range-limit effects. It assigns 151 of 476 attributed issues to packet loss with battery depletion.

The Open Smart Home dataset exhibits very few issues across all cases (42 issues, 84 cases, 0.5 issues/case). Conformance remains high (fitness=0.954, quality score=94.4). A drift (i.e., gradual change in sensor readings) affects a temperature sensor through a gradual offset (+0.5℃ over 30 days). *LIQUID* detects and attributes the drift reliably (40/42 issues attributed, 95.2%).

Probabilistic causal backtracking supports the P5 explanation. The mean chain strength is 0.81 (SD=0.09). Dedicated drift detection remains necessary [10]. However, the chain terminates at the correct sensor.

LIQUID's correctness is statistically validated: Both precision ($M = 88.2\%$, $t(4) = 13.23$, $p < .001$, $d = 5.92$) and recall ($M = 89.0\%$, $t(4) = 8.99$, $p < .001$) exceed the 50% baseline. Attribution accuracy is similarly high ($M = 86.4\%$, $t(4) = 13.01$, $p < .001$, $d = 5.82$), supported by a binomial test (1,638 of 2,036, 80.5%, $p < .001$). Performance is consistent, with low variability (CV = 7.3%).

EG2: Robustness to Parameter Variation The robustness evaluation follows the sensitivity-analysis workflow of Pianosi et al. [19]. They define the following seven phases: goal definition, method selection, uncertainty specification, sample generation, experimentation, convergence checking, and result visualization. Parameter sensitivity is assessed through one-at-a-time (OAT) sweeps across key pipeline parameters. Table 4 presents the sensitivity classification and impact ranges for all ten parameters. Parameters are classified based on their total-order Sobol indices (S_T): **HIGH** sensitivity ($S_T > 0.1$, >20 percentage point impact on metrics), **MEDIUM** sensitivity ($0.01 < S_T \leq 0.1$, 7 - 20 percentage point impact), and **LOW** sensitivity ($S_T \leq 0.01$, <7 percentage point impact).

Table 4: Pipeline Parameter Sensitivity Classification and Impact Ranges, showing that only three parameters require careful domain-specific tuning while the remaining ones are robust to moderate variation.

Parameter	Range	Default	Sensitivity Class	Impact Range (pp)	Primary Effect
Critical Parameters (Require Careful Calibration)					
Min Acceptable Rate (Hz)	[0.1, 2.0]	1.0	**HIGH**	24.0	Gap detection recall
Outlier IQR Multiplier	[1.0, 3.0]	1.5	**HIGH**	30.0	False positive rate
Conformance Threshold	[0.5, 0.9]	0.7	**HIGH**	21.7	Backtracking trigger
Moderate Parameters (Standard Tuning Sufficient)					
Noise Level Threshold	[0.05, 0.2]	0.1	**MEDIUM**	7–15	Degradation detection
Gap Detection Multiplier	[2.0, 5.0]	3.0	**MEDIUM**	>10*	Sampling adequacy
Inconsistency CV Threshold	[0.3, 1.0]	0.5	**MEDIUM**	>10*	Placement detection
Fitness Threshold	[0.4, 0.8]	0.6	**MEDIUM**	>10*	Model quality gate
Precision Threshold	[0.5, 0.9]	0.7	**MEDIUM**	>10*	Model quality gate
Stable Parameters (Default Values Recommended)					
Clipping Ratio Threshold	[0.05, 0.25]	0.1	**LOW**	<5	Range saturation
Outlier Ratio Threshold	[0.01, 0.15]	0.05	**LOW**	<5	Noise classification

* Full Sobol analysis not converged (N=1000 samples insufficient); OAT estimates provided.
pp = percentage points; CV = coefficient of variation.

Critical Parameters Requiring Domain-Specific Tuning Critical parameters require domain-specific tuning because they show strong effects on detection and attribution. The *Min Acceptable Rate Hz* parameter causes a 24 percentage point (pp) recall variation due to heterogeneous sampling rates across datasets. The *Outlier Iqr Multiplier* produces a 30pp precision range, while the *Conformance Threshold* shifts attribution accuracy by 21.7pp. These parameters directly control detection behavior and require calibration for each deployment.

Moderate and Stable Parameters Moderate parameters reveal dataset-dependent behavior and medium impact. Stable parameters remain robust and require minimal tuning. The *Noise Level Threshold* yields stable performance within 0.08–0.15, and the *Gap Detection Multiplier* performs when adapted to sampling patterns. In contrast, the *Clipping Ratio Threshold* and *Outlier Ratio Threshold* vary less than 5pp. These parameters benefit from moderate tuning without driving sensitivity as strongly as the critical group.
LIQUID is robust to parameter changes. Seven of ten parameters (70%) show LOW or MEDIUM sensitivity and behave stably. A Mann-Whitney U test confirms clear separation between stable and critical parameters ($U = 0$, $p = .011$). Bootstrap analysis with $N = 10{,}000$ shows that critical parameters cause 16.3 percentage points more impact (95% CI: $[12.1, 21.1]$ pp). The interval excludes zero. Cross-dataset variability is low (CV = 7.6%). Only three parameters need domain-specific tuning.

EG3: Explainability and Actionability EG3 evaluates *LIQUID*-generated insights against R1–R5. R1 concerns completeness and impact, R2 layers and confidence cues, R3 causal and actionable links, R4 semantic integration, and R5 structured communication.
Explanation Completeness (R1, R5): LIQUID generates complete explanations in 87.2% of cases. Completeness requires deviation, causal chain, confidence values, interpretation, and action. Incomplete explanations result from early stops (8.3%) or low-confidence gaps (4.5%).
Layered Explanations (R2): The three layers provide an executive summary, analyst-facing detail, and action-oriented recommendation. They expose missing cases, correlation gaps, and sampling errors. Bayesian scores enable filtering, with 64.3% of OPPORTUNITY explanations reaching at least 0.7.
Probabilistic Causal Links (R3): The chains rank sensor faults as explanations for process-level deviations by ordering intermediate effects. In Intel Berkeley, voltage drops precede clipping and missing readings. Chain strength exceeds the strong-link threshold in 58.7% of cases with a mean value of 0.53.
Semantic Integration (R4): Templates map raw signals to domain concepts, and Smart Home drift appears as overheating with compensatory cooling. Most complete explanations contain semantic terms (92.1%).
Information Gain (R2, R3): Interpretability and actionability gains support prioritization. Mean gains across datasets are 0.618 and 0.542, respectively.
Comparative Analysis: LIQUID explains 4 of 6 Future Factory deviations (66.7%). Targeted interventions improve fitness by +0.12 and reduce duplicates by 83%, while lower-confidence deviations receive cautious guidance.
The explainable power of *LIQUID*'s insights is statistically validated. Explanation completeness exceeds the baseline ($M = 89.4\%$, $t(4) = 20.59$, $p < 0.001$, $d = 9.21$). Chain strength exceeds 0.5 ($M = 0.692$, $t(4) = 5.14$, $p = 0.003$), and it correlates strongly with confidence ($r = 0.96$, $p = 0.010$). Interpretability gain ($M = 0.648$, $p = 0.001$), actionability gain ($M = 0.566$, $p = 0.031$), and semantic coverage (92.0%, $p < 0.001$) are also significant.

5 Related Work

The approach relates to explainability in process mining. Tentina et al. [23] emphasize analysis goals, data scope, and processing context. Ammann et al. [1] emphasize user archetypes and confidence cues. Stein Dani et al. [20] emphasize role-specific actions. Mendling et al. [14] and Zimmermann et al. [24] emphasize cognitive effectiveness and communication gaps. These studies motivate our explanation requirements, but they do not trace sensor-level causes through an IoT process mining pipeline.

The approach also relates to uncertainty in event logs. Pegoraro et al. [18] estimate probabilities of uncertain trace realizations. Lepsien et al. [13] compute top-K realizations efficiently. Klinkmüller et al. [11] debug process discovery pipelines through uncertainty and sensitivity analysis. These approaches are complementary to *LIQUID*. However, they do not connect IoT faults to process-level deviations through probabilistic causal backtracking.

6 Conclusion & Future Work

This work presents *LIQUID*, an approach to explain conformance deviations in IoT-enhanced process mining. The approach combines probabilistic reasoning with probabilistic causal backtracking and traces conformance deviations back to their sensor-level origins. Evaluation across five datasets demonstrates high detection and attribution accuracy.

Internal Validity Threats. Expert annotations may contain ambiguous faults. The selected priors and evidence functions may influence confidence values. The evaluation mitigates these threats through GT1/GT2 separation, statistical tests, and sensitivity analysis. However, *LIQUID* still identifies plausible explanatory chains rather than experimental causality. No human assessors rated the explanations. Proxy metrics therefore cannot establish perceived interpretability or actionability.

External Validity Threats. The five datasets cover heterogeneous IoT settings, but they cannot represent all sensor deployments. *LIQUID* requires raw sensor streams, a documented abstraction pipeline, and domain knowledge for priors and evidence functions. Evidence-function thresholds and scales need sensor-specific calibration. Their performance on unseen sensor types remains unverified. The method is less suitable for event logs without raw data access or for deviations without observable sensor segments.

Future work should explore three directions. First, advanced visualization techniques could support interactive causal-chain exploration. Second, causal inference techniques could distinguish genuine sensor-to-process effects from coincidental correlations and reduce manual rule design. Third, large language models (LLMs) could generate adaptive explanations for different user archetypes.

Acknowledgments. This work received funding from the Deutsche Forschungsgemeinschaft (DFG), grant 496119880. This work was supported in part by the Research Foundation Flanders (FWO) under Project 1294325N.

References

[1] Jana Ammann, Laura Lohoff, Bastian Wurm, and Thomas Hess. "How do Process Mining Users Act, Think, and Feel?" In: *Bus Inf Syst Eng* (2025).

[2] Yannis Bertrand, Martin Kabierski, Jari Peeperkorn, and Seppe vanden Broucke. "A framework for measuring data quality sensitivity in process discovery". In: *Process Mining Workshops ICPM 2025*. Cham: Springer, 2026.

[3] Yannis Bertrand, Rafaël Van Belle, Jochen De Weerdt, and Estefanía Serral. "Defining Data Quality Issues in Process Mining with IoT Data". In: *Process Mining Workshops*. Cham: Springer, 2023, pp. 422–434.

[4] Yannis Bertrand et al. "Challenges in data quality management for IoT-enhanced event logs". In: *International Conference on Research Challenges in Information Science*. Springer, 2025, pp. 20–36.

[5] R. P. Jagadeesh Chandra Bose, Ronny S. Mans, and Wil M. P. Van Der Aalst. "Wanna improve process mining results?" In: *2013 IEEE Symposium on Computational Intelligence and Data Mining (CIDM)*. IEEE, 2013, pp. 127–134.

[6] George E. P. Box and George C. Tiao. *Bayesian inference in statistical analysis*. John Wiley & Sons, 1973.

[7] Shokoufeh Ghalibafan, Sareh Sadeghianasl, and Moe Thandar Wynn. "eRooMiner: A Data-Driven Approach for Root Cause Detection of Process Data Quality Issues". In: *Cooperative Information Systems: 31st International Conference 2025: CoopIS 2025*. Springer, Sept. 2025.

[8] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. "Design Science in Information Systems Research". In: *Management Information Systems Quarterly* 28.1 (2004), pp. 75–105.

[9] Aimad Karkouch, Hajar Mousannif, Hassan Al Moatassime, and Thomas Noël. "Data Quality in Internet of Things: A state-of-the-art survey". In: *Network and Computer Applications* 73 (Aug. 2016).

[10] Nathavuth Kitbutrawat et al. "Sensor drift detection by utilizing multi-sensor signals". In: *SICE Annual Conference 2011*. 2011, pp. 1523–1527.

[11] Christopher Klinkmüller et al. "A Method for Debugging Process Discovery Pipelines to Analyze the Consistency of Model Properties". In: *Business Process Management*. Vol. 12875. LNCS. Springer, 2021, pp. 65–84.

[12] Agnes Koschmider et al. "Process Mining for Unstructured Data: Challenges and Research Directions". In: *Modellierung 2024*. Gesellschaft für Informatik e.V., 2024, pp. 119–136.

[13] Arvid Lepsien et al. "Ranking the Top-K Realizations of Stochastically Known Event Logs". In: *ICPM*. Vol. 533. LNBIP. Springer, 2025, pp. 350–362.

[14] Jan Mendling, Djordje Djurica, and Monika Malinova. "Cognitive Effectiveness of Representations for Process Mining". In: *Business Process Management*. Cham: Springer, 2021, pp. 17–22.

[15] Judea Pearl. "Causal Inference in Statistics: An Overview". In: *Statistics Surveys* 3 (Jan. 2009), pp. 96–146.

[16] Judea Pearl. *Causality*. 2nd ed. Cambridge University Press, 2009.
[17] Judea Pearl. "Embracing Causality in Formal Reasoning". In: *AAAI Conference on Artificial Intelligence*. 1987. URL: https://api.semanticscholar.org/CorpusID:34608468.
[18] Marco Pegoraro, Bianka Bakullari, Merih Seran Uysal, and Wil M. P. van der Aalst. "Probability Estimation of Uncertain Process Trace Realizations". In: *ICPM Workshops*. LNBIP. Springer, 2022, pp. 21–33.
[19] Francesca Pianosi et al. "Sensitivity analysis of environmental models: A systematic review with practical workflow". In: *Environmental Modelling & Software* 79 (2016), pp. 214–232.
[20] Vinicius Stein Dani et al. "From Process Mining Insights to Process Improvement: All Talk and No Action?" In: *Cooperative Information Systems*. Cham: Springer, 2024, pp. 275–292.
[21] Suriadi Suriadi, Chun Ouyang, Wil M. P. van der Aalst, and Arthur H. M. ter Hofstede. "Root Cause Analysis with Enriched Process Logs". In: *Business Process Management Workshops*. Berlin, Heidelberg: Springer, 2013, pp. 174–186.
[22] Hui Yie Teh, Andreas W Kempa-Liehr, and Kevin I-Kai Wang. "Sensor data quality: A systematic review". In: *Journal of Big Data* 7.1 (2020), pp. 1–49.
[23] Irina Tentina, Francesca Zerbato, Felix Mannhardt, and Boudewijn van Dongen. "Why Do Users Struggle to Get Insights from Process Mining?" In: *2025 7th International Conference on Process Mining (ICPM)*. 2025 7th International Conference on Process Mining (ICPM). IEEE, 2025, pp. 1–8.
[24] Lisa Zimmermann, Francesca Zerbato, and Barbara Weber. "What makes life for process mining analysts difficult? A reflection of challenges". In: *Software and Systems Modeling* 23.6 (2024), pp. 1345–1373.