

(REVIEW ARTICLE)



BUSINESS PROCESS MONITORING INTELLIGENCE IN SAP CLOUD ALM: A FRAMEWORK FOR DRIFT DETECTION, PREDICTIVE ANALYTICS, AND EXPLAINABLE INSIGHTS

Karthik Paramasivam *

PSG College of Technology, Peelamedu, Coimbatore, Tamil Nadu 641004, India.

* Corresponding Author

ORCID Details

Karthik Paramasivam: <https://orcid.org/0009-0009-7437-1975>

World Journal of Advanced Engineering Technology and Sciences, 2026, 20(03), 147–156

Publication history: Received on 02 June 2026; revised on 16 September 2026; accepted on 18 September 2026

Article DOI: <https://doi.org/10.30574/wjaets.2026.20.3.0358>

Copyright © 2026 Author(s) retain the copyright of this article. This article is published under the terms of the Creative Commons Attribution License 4.0

ABSTRACT

Process owners in SAP-managed enterprises face a recurring operational gap: threshold-based alerting indicates that a KPI has crossed a boundary but provides no explanation of whether that movement reflects a gradual trend, expected seasonal variation, or a correlated pattern emerging across several related KPIs simultaneously. Acting on a single threshold breach without this context leads to delayed response, misrouted escalations, and missed early-warning signals that never individually crossed a boundary yet collectively represent a developing process risk. This paper proposes a Business Process Monitoring Intelligence framework that extends SAP Cloud ALM BPMon with category-aware drift detection and LLM-assisted predictive explanation. The framework maps standard and custom KPIs into process hierarchy nodes, validates data readiness and retention conditions, extracts KPI status and history through the Business Process Monitoring Analytics API, and classifies drift through four patterns: threshold breach, trend slope, seasonal deviation, and multi-KPI co-drift. A bounded LLM-assisted explanation layer then produces a process-owner-readable narrative, a short-horizon forecast with explicit invalidating conditions, evidence references, and a recommended action class routed through existing access group ownership. The framework draws on current SAP documentation for BPMon standard content, custom KPI instrumentation for both ABAP and BTP, custom process structures, data readiness prerequisites, and the BPMon Analytics API. No empirical results against live BPMon KPI data are reported; all drift detection and forecasting claims are design propositions for future empirical validation.

Keywords: SAP Cloud ALM; Business Process Monitoring; process drift detection; predictive analytics; LLM-assisted explanation; BPMon Analytics API; co-drift

1 INTRODUCTION

When a KPI in SAP Cloud ALM Business Process Monitoring breaches a threshold, the system alerts the team responsible for the process. What does not automatically tell you is whether the breach was anticipated, for example, whether it reflects a quarter-end surge or a slowdown that began a few monitoring periods earlier. It also does not indicate when three KPIs on a process node in the same region all move adversely while remaining below their respective alert

thresholds, collectively indicating a growing problem that none of the individual alerts would detect. The issue this paper tackles is the gap between threshold alerting and process intelligence.

SAP Cloud ALM BPMon provides a strong operational foundation: standard end-to-end process content (Lead to Cash, Source to Pay, Design to Operate, Recruit to Retire); custom KPI support for ABAP-based and BTP-based applications; custom process structures beyond the standard taxonomy; and a documented Business Process Monitoring Analytics API for extracting KPI status and history [1,2]. This foundation lacks the ability to interpret KPI movement over time, compare it with a seasonal baseline, correlate movement across process nodes, and provide a meaningful, process-owner-readable explanation together with a short-horizon forecast.

This paper explores how SAP Cloud ALM BPMon KPIs, presented in the traditional and custom process hierarchy, can be analyzed for process drift beyond point-in-time threshold alerts, and how a bounded LLM-assisted explanation layer can transform detected process drift into a grounded root-cause story and short-term forecast delivered to the accountable process owner. One novelty of the approach proposed here is the integration of the category-aware drift detection, the seasonal deviation analysis and the multi-KPI co-drift scoring into a bounded, grounded explanation architecture within a process-hierarchy and access-group model that routes narratives to accountable owners.

2 SAP CLOUD ALM BPMON ARCHITECTURE

SAP Cloud ALM Business Process Monitoring provides visibility into end-to-end process health via standard process areas and predefined KPIs. With BPMon, line-of-business and IT users can gain insight into disruptions throughout the system landscape, and take corrective measures before a critical situation occurs, thereby following a detect-to-correct lifecycle for SAP solution landscapes [1,9]. The target personas (Process Manager, Application Owner and operational business users) indicate that BPMon explanations should be business-oriented and not just technical alerts.

The five areas covered by the standard KPI content are Lead to Cash (order, delivery, billing, payment); Source to Pay (procurement, supplier invoice); Design to Operate (manufacturing, logistics); Recruit to Retire (HR process health); and Custom KPIs (indicators defined by the organization). The standard KPI coverage relies on the current ABAP-based system (ST-A/PI) [2,10]. Once established, custom KPIs are part of the same monitoring process as standard KPIs and up to fifteen custom KPIs can be added to standard content [5,7].

For ABAP-based components, the discovery and activation process registers the KPI for the monitoring cycle [3,12], and each custom KPI must be a class implementing the methods GET_KPI_ID and GET_DATA. For SAP BTP applications, data records are sent using a push model and include KPI identifiers and UTC timestamps [4]. Through the custom process structures, organizations can design, import and track their own process hierarchies in addition to standard content, with standard and custom KPIs [11]. Access groups represent process ownership and control who receives drift narratives. Threshold ratings are based on categories: higher values indicate poorer performance for backlog, lead-time, and exception KPIs but better performance for throughput KPIs [6,13]. The BPMon Analytics API provides programmatic access to KPI status, rating, dimensions and historical values [14,15].

Table 1: BPMon process areas and research relevance

BPMon Process Area	Research Relevance for Drift Intelligence
Lead to Cash	Revenue, order, delivery, billing, and fulfillment drift including cross-KPI co-drift within order-to-cash flows
Source to Pay	Procurement, goods receipt, supplier invoice, and payment exception drift useful for supplier and procurement risk
Design to Operate	Planning, manufacturing, logistics, and execution drift useful for supply chain continuity monitoring
Recruit to Retire	Workforce, hiring, onboarding, and HR process drift useful for HR process health and workforce operations
Custom KPIs / Custom Processes	Company-specific process intelligence beyond standard content enables industry-specific controls and local ownership

3 RESEARCH METHODOLOGY

This paper adopts a design-science research approach [16]. The purpose here is not to report a controlled field experiment but to propose reusable architecture for a recurrent enterprise problem: business-process monitoring should not rely solely on threshold alerts when gradual, seasonal, or multi-KPI drift patterns are present. The design process consists of the identification of the problem domain, the grounding of the SAP technical design, the specification of the evidence model, the proposal of a drift taxonomy, the specification of the LLM Explanation Layer and the proposal of governance and evaluation controls. The paper does not use any specific KPI values, thresholds, process structures or tenant names. Practitioner-grounded reasoning is used in the paper.

Table 2: Design-science mapping from problem to artifact

Design Element	Operational Issue	Artifact Response
Problem identification	Threshold alerts capture point-in-time violations but miss gradual or multi-KPI drift	Define a drift detection layer beyond single-KPI thresholds
Evidence definition	Standard/custom KPIs, process hierarchy, and KPI history reviewed independently	Normalize KPI history into category-aware drift features
Readiness model	Poor component readiness, short retention, or inactive collection can mimic stability or drift	Add BPMon readiness and data quality gates before forecasting
API strategy	Custom dashboards need a stable extraction layer for KPI status and trend	Use the BPMon Analytics API as the primary extraction source
Reasoning model	Process owners want predicted, not only current, performance	Build drift classifier plus LLM-assisted forecast and explanation
Governance	Explanations can reach business users who cannot validate technical assumptions	Require evidence-grounded, role-appropriate, human-confirmed narratives

4 BPMON INTELLIGENCE FRAMEWORK

The proposed architecture includes seven layers, which are operational and do not replace SAP Cloud ALM BPMon. BPMon is still the system of record for setting up KPIs, configuring thresholds, assigning processes and monitoring the status. The end-to-end architecture is shown in Figure 1.

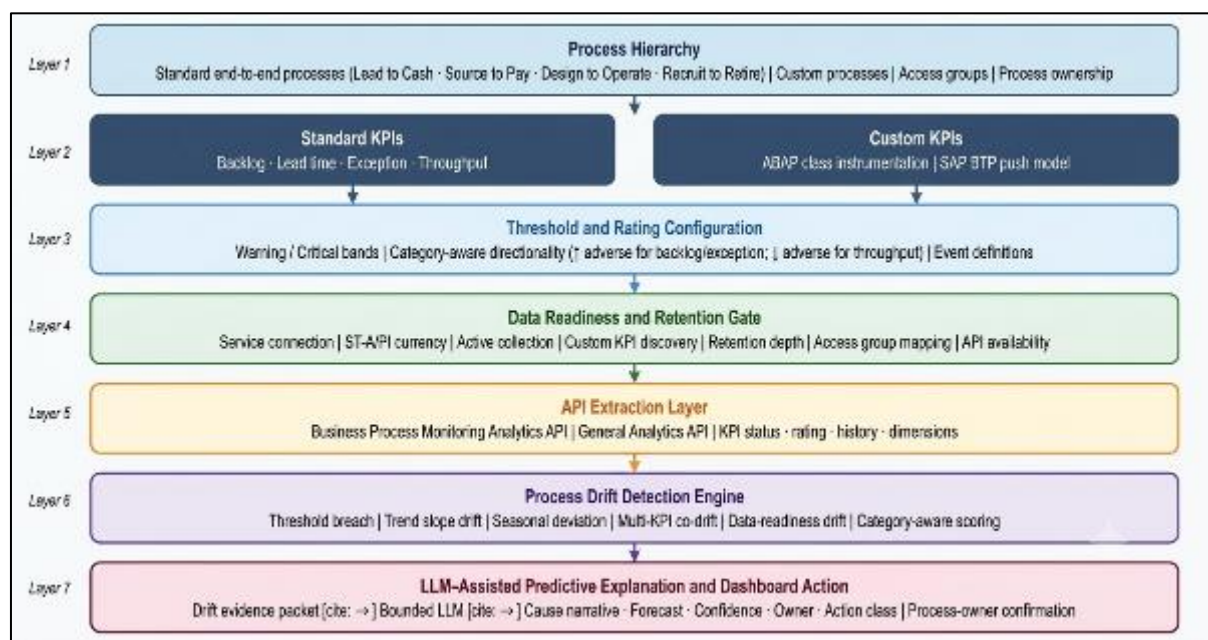


Figure 1: Business Process Monitoring Intelligence Framework Architecture

Table 3: Framework layers and operational responsibilities

Layer	Key Inputs	Output
Process hierarchy	Standard and custom processes, sub-processes, access groups	Process structure with owner context
KPI layer	Standard KPIs, custom ABAP/BTP KPIs	KPI values by process node and category
Threshold and rating	Warning/Critical bands, filters, event definitions	Threshold alerts and KPI rating
Readiness and retention	Component connection, ST-A/PI, active collection, retention settings	Confidence gates for drift scoring
API extraction	BPMon Analytics API and general Analytics API	KPI status, rating, and history feed
Process drift detection	KPI history, seasonal baseline, process context	Scored drift candidates by pattern
LLM predictive explanation	Drift evidence packet and governance prompt	Process-owner narrative, forecast, action class

The framework also clearly delineates between the native BPMon features and the proposed intelligence extensions. The native BPMon configuration (home views, KPI details, event definitions, alert inbox, saved views, custom KPI cards, etc.) is preserved, and the analytics layer is built on top of the same evidence already exposed by the BPMon Analytics API. The intelligence extensions provide drift severity indicators, historical baseline comparisons, co-drift explanations, forecast confidence and owner-specific action recommendations without the need for a second monitoring system.

5 CUSTOM KPI LIFECYCLE AND PROCESS HIERARCHY INTEGRATION

The possible KPI states and their interpretations are important for drift intelligence because there are a variety of states that appear the same on a dashboard but have very different implications for model confidence. The framework considers the lifecycle as a state machine with the following states: defined, instrumented and discovered, collecting, assigned, threshold-configured, API-extractable and drift-ready. A KPI that is technically detected but has not yet been assigned to a process node provides a context-free explanation. Defects in the ABAP class that supplies KPI values can create the impression of process drift even when the underlying process has not changed.

Table 4: Custom KPI lifecycle stages and framework touchpoints

Lifecycle Stage	SAP Mechanism	Framework Touchpoint
KPI definition	KPI Management in BPMon; category selection	Establishes category directionality for drift scoring
Source instrumentation	ABAP class (GET_KPI_ID / GET_DATA) or BTP push payload	Defines expected data shape and collection frequency
Auto discovery and activation	Scheduled discovery makes KPI available for monitoring cycle	Distinguishes activation gap from business drift
Process node assignment	KPI mapped to standard or custom process hierarchy node	Provides business-process context for explanation routing
Access group mapping	Process structure reflected in access groups	Routes drift explanation to accountable process owner
Threshold configuration	Warning/Critical rating, filters, alert recipients	Supplies point-in-time baseline signal for co-drift comparison
API extraction readiness	BPMon Analytics API response validated and history confirmed	Enables reproducible feature extraction and drift scoring

Custom processes enhance the relevance of each drift story. A global Source-to-Pay process view may be too broad for a regional procurement team. If drift is mapped to a custom regional process node with its own access group/ownership, the LLM explanation can be routed to the person who can take action rather than to a generic shared inbox. Access

groups also serve a data-governance function because KPI data showing backlog volumes, credit holds, or supplier exceptions may be commercially sensitive and must not be shared beyond the group responsible.

6 PROCESS DRIFT DETECTION

Threshold-based alerting answers the binary question of whether a KPI has currently exceeded a threshold. Drift detection instead addresses whether the process is changing in a significant manner even when no individual threshold has been crossed. Four drift patterns, in addition to a data-readiness class, are proposed for the new model. All scoring is category-aware: higher values indicate worse performance for backlog, lead-time, and exception KPIs, whereas lower values indicate worse performance for throughput KPIs. If a model treats all slopes in the same way, it will classify a throughput decrease as stable behavior and an increase in throughput as a risk. The entire detection and routing logic is illustrated in Figure 2.

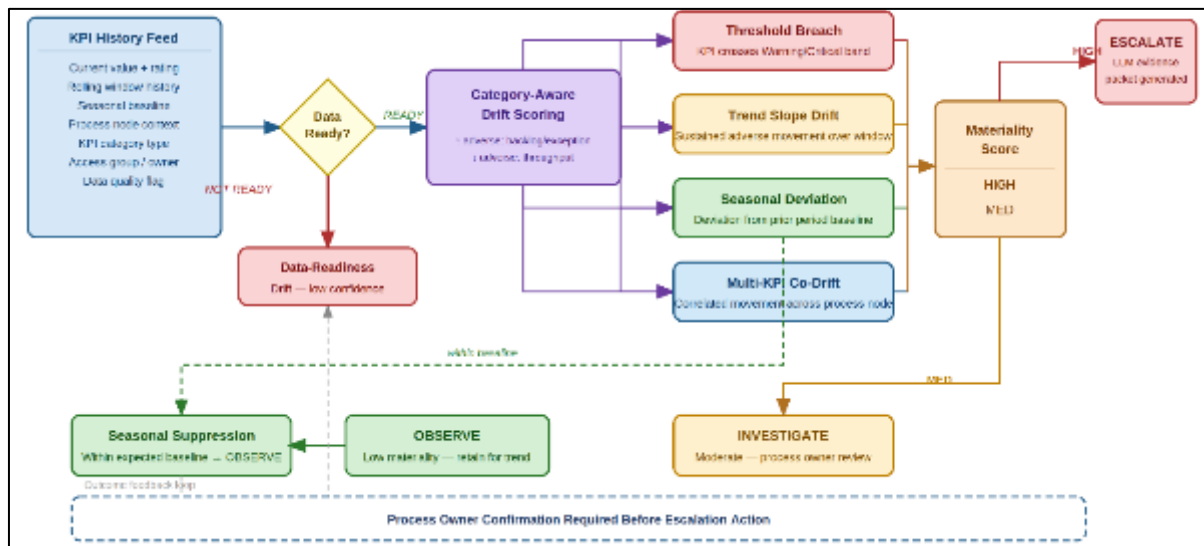


Figure 2: Process Drift Detection and Routing Logic

Table 5: Process drift taxonomy

Drift Pattern	Detection Logic	Why Threshold Alerting May Miss It
Threshold breach	Current value crosses Warning or Critical band	Not missed; this is the baseline signal the framework also includes
Trend slope drift	Sustained adverse directional movement over a rolling window	KPI may worsen steadily while remaining inside the threshold band
Seasonal deviation	Deviation from expected periodic pattern using matched prior periods	Recurring expected spikes misclassified as anomalies without seasonality baseline
Multi-KPI co-drift	Correlated adverse movement across related KPIs in the same process node	Individually mild deviations invisible to single-KPI threshold alerts
Data-readiness drift	Missing, stale, or newly started KPI history	May be confused with process stability; requires separate classification

Table 6: Drift scoring model components

Score Component	Formula Concept	Interpretation
Trend score	direction multiplier(category) x slope over rolling window	Whether the KPI is moving adversely for its category
Seasonal deviation score	abs(current period - matched prior period baseline) / baseline	Whether current behavior deviates from expected seasonal pattern

Co-drift score	Count of correlated KPIs x shared process node weight	Whether related KPIs are moving together
Threshold proximity score	Distance between current value and Warning/Critical threshold	How close the KPI is to entering Critical state
Business criticality weight	Criticality of affected process node	Whether the affected process matters to the business
Data quality penalty	Missing, stale, or insufficient history discount	Whether the explanation and forecast should be treated as tentative

7 LLM-AUGMENTED PREDICTIVE ANALYTICS

The LLM layer is deliberately constrained. It does not inspect raw KPI data or generate explanations beyond the structured evidence packet. It takes in a structured drift evidence packet and generates a bounded narrative for process owners rather than platform engineers. There is a specific item in the evidence packet to which each output field refers. The prompt explicitly instructs the model to distinguish between observed evidence and inferred cause; to make a forecast only when a trend or seasonality is noted in the evidence; to state conditions that make a forecast invalid; and to refuse to form a hypothesis when evidence is incomplete or the correlation is weak. Figure 3 shows the governance flow.

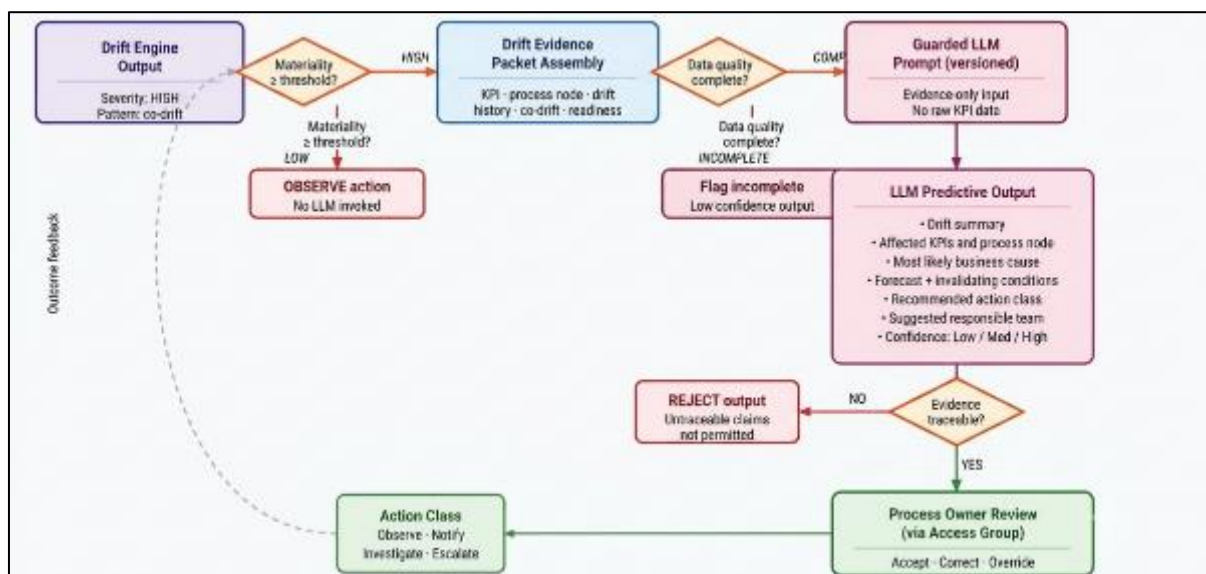


Figure 3: LLM Predictive Explanation Governance Flow

The drift evidence packet includes the following information: KPI ID, category, current value and rating, process node and business process area, pattern classification and severity of the drift, the slope of the trend, the seasonal baseline deviation, co-drift candidate KPIs with a shared node and timing relationship, and output requirements such as the confidence level and owner routing. This structured input helps to prevent the LLM from creating claims that are not linked to verifiable evidence, a major governance risk in using language models for operations in business processes.

Table 7: LLM predictive output schema

Output Field	Description	Governance Purpose
Drift summary	Plain-language description of detected change	Orients the process owner without requiring technical interpretation
Affected KPIs and process node	Which KPIs and which process hierarchy node are involved	Scopes the explanation to the evidence
Drift pattern classification	Threshold, trend, seasonal, co-drift, or data-readiness	States what changed and how it was detected

Most likely business cause	Hypothesis grounded in evidence — never invented	Separated from observed fact by prompt guardrail
Forecast and invalidating conditions	Short-horizon prediction with conditions that would falsify it	Makes forecast falsifiable; prevents overconfident action
Recommended action class	Observe / Notify / Investigate / Escalate	Converts explanation into operational next step
Suggested responsible team	Owner derived from access group — suggestion, not assignment	Preserves human accountability for routing decisions
Confidence	Low / Medium / High with stated reason	Calibrates process-owner trust per incident

8 GOVERNANCE AND DASHBOARD

BPMon KPI data can be proprietary and business-sensitive. The delivery block backlog and credit hold rates, regional lead times and supplier invoice exceptions may reveal operational performance and commercial exposure. For this purpose, the framework will route all drift narratives through existing access groups used by the BPMon process structures and not generate generic AI summaries. Each LLM output should be linked to the evidence packet, prompt version, model version and decision made by the process owner.

The action classes should be treated as sequential workflow states rather than independent actions. The LLM makes suggestions; a process owner or designated analyst validates, corrects, or overrides them prior to action. Confirmed results become ‘feedback’ labels that can be used for subsequent model calibration, completing the cycle of explanation and learning.

Table 8: Governance controls

Risk	Control	Evidence to Retain
Sensitive business data exposure	Route explanations through existing access-group permissions	Access group mapping and delivery log
Inaccurate forecast undermining trust	Require confidence level and invalidating conditions in every forecast	Forecast text, confidence label, and later outcome comparison
Instrumentation failure misread as drift	Track expected KPI freshness; classify pipeline gaps as data-readiness drift separately	Data-quality status per custom KPI and collection log
Retention too short for trend analysis	Validate historical depth before seasonal or co-drift scoring	Retention setting and available history length
Prompt drift	Version all LLM prompt templates; store version with every output	Prompt version and approval history
Over-automation	Require process-owner confirmation before any escalation action	Confirmation record and override log

Table 9: Dashboard sections and persona alignment

Dashboard Section	Main Content	Primary Persona
Process health overview	Process hierarchy with KPI rating and drift status by node	Process Manager
KPI trend and history	Standard and custom KPI trends, rating, seasonal baseline	Application Owner / Process Manager
Drift and co-drift detail	Drift pattern, severity, correlated KPIs, timing order	Process Manager
Predictive explanation panel	LLM cause narrative, forecast, evidence, confidence	Process Manager

Action and ownership	Recommended action, responsible group, status	Operational Agent
Feedback and closure	Confirmed cause, outcome, override, learning label	Process Manager

9 MATURITY MODEL AND EVALUATION

The maturity model separates the extent of BPMon configuration from the level of process intelligence that is created on top of it. An organization may configure both standard and custom KPIs but still operate at a low level of process intelligence if drift is detected only when a threshold is breached and explanations are being compiled manually. Predictive process intelligence should not be attempted until managed components are connected, ST-A/PI is current (when applicable), data collection is in place, data has been retained for an adequate amount of time, and API extraction is reliable.

Table 10: BPMon intelligence maturity model

Level	Characteristic State	Key Dependency
Level 1 — Threshold-only monitoring	Standard and custom KPIs configured; alerts fire only on threshold breach	Basic KPI and threshold setup
Level 2 — Readiness and hierarchy control	Managed components connected, ST-A/PI current, data collection active, process hierarchy and access groups defined	BPMon setup quality and retention
Level 3 — API-enabled trend analysis	BPMon Analytics API used for historical and trend reporting beyond native dashboards	API reliability and historical depth
Level 4 — Drift detection	Trend, seasonal, and co-drift patterns detected beyond threshold breach	Category-aware feature engineering
Level 5 — Predictive explainable process intelligence	Grounded cause narratives and forecasts delivered to process owners with feedback learning	Governance, trust, and forecast accuracy tracking

Implementation should proceed in phases: first, establish data and process readiness by inventorying standard and custom KPIs and validating service and system connections and ST-A/PI status; second, establish active data collection and sufficient retention depth; third, introduce LLM-assisted explanations through human-reviewed pilots; and finally, deploy process-owner dashboards with feedback capture. Each phase is valuable and does not require the whole LLM layer to be running.

Table 11: Research propositions

Prop.	Statement
P1	Trend-slope and co-drift detection will identify material process issues earlier than threshold-only alerting alone
P2	Seasonally aware drift detection will reduce false drift alerts compared with non-seasonal trend detection during known cyclical periods
P3	LLM-generated drift explanations grounded in process hierarchy context will improve process-owner understanding compared with raw KPI trend review
P4	Custom-process-aligned drift explanation will improve routing and actionability compared with explanations based only on standard process taxonomy
P5	Data-readiness gating will reduce false drift conclusions caused by stale custom KPI collection or insufficient retention
P6	Explicit forecast confidence and invalidating conditions will improve process-owner trust compared with unqualified forecasts
P7	Access-group-consistent explanation routing will be necessary for adoption in organizations with regulated or commercially sensitive process data

9.1 Limitations and Validation Scope

In this paper, a conceptual framework and a detailed design model are proposed. The paper does not report controlled field results from an active BPMon KPI dataset in a connected tenant. The framework is based on existing SAP documentation as well as on existing process-mining and predictive-monitoring literature and practitioner design judgment, but the empirical claims should be treated as propositions for future research.

Future validation would require: (1) access to historical BPMon KPI data for several monitoring periods and process areas, preferably across several tenant configurations; (2) a formal evaluation protocol comparing the lead time of drift detection with subsequent threshold violations; (3) a comparative assessment of forecast accuracy by KPI category within a rolling time window to prevent temporal data leakage; and (4) collection of process-owner usefulness and trust ratings for LLM-generated explanations versus raw KPI trend review. These are the main requirements for future empirical validation of the research propositions defined in Section 9. A similar validation exercise for BPMon KPI history can be implemented with minimal effort by extending the same exercise that was carried out in the companion telemetry paper in this portfolio for the same evidence domains.

There are some additional framework dependencies as follows: KPI history depth – for seasonal and co-drift analysis, newly instrumented custom KPIs will show low-confidence forecasts until sufficient KPI history has accumulated; instrumentation reliability for ABAP-based and BTP-based custom KPIs; and LLM forecast maturity – this is the least mature component and should only be introduced once the evidence model is stable and human validation is confirmed as a requirement before any operational action is taken.

10 CONCLUSION

With SAP Cloud ALM Business Process Monitoring businesses find a structured, business-oriented platform for monitoring the health of processes with standard KPIs, custom KPIs, process hierarchies, category-aware threshold alerting and the BPMon Analytics API. This foundation can consistently identify point-in-time threshold breaches. However, threshold monitoring alone does not explain gradual, seasonal, or multi-KPI drift, nor does it provide a forward-looking business narrative.

This paper proposes a Business Process Monitoring Intelligence framework that leverages LLM capabilities to provide drift explanations and predictive root-cause explanations while incorporating data-readiness gating, customizable KPI lifecycle awareness, process-hierarchy ownership, category-aware drift scoring across four pattern types, co-drift analysis, and access-group governance, co-drift analysis, and access group governance. The research output is a reusable framework and validation agenda to study the possibility of utilizing BPMon KPI history as a predictive layer using business language. The potential usability lies in moving from threshold-triggered alerting to explainable process intelligence that process owners can review before acting.

REFERENCES

- [1] Dumas, M., Fournier, F., Limonad, L., Marrella, A., Montali, M., Rehse, J.-R., Accorsi, R., Calvanese, D., De Giacomo, G., Fahland, D., Gal, A., La Rosa, M., Völzer, H., & Weber, I. (2023). AI-augmented business process management systems: A research manifesto. *ACM Transactions on Management Information Systems*, 14(1), Article 11, 1–19.
- [2] Márquez-Chamorro, A. E., Resinas, M., & Ruiz-Cortés, A. (2018). Predictive monitoring of business processes: A survey. *IEEE Transactions on Services Computing*, 11(6), 962–977.
- [3] Ceravolo, P., Comuzzi, M., De Weerd, J., Di Francescomarino, C., & Maggi, F. M. (2024). Predictive process monitoring: Concepts, challenges, and future research directions. *Process Science*, 1, Article 2.
- [4] Tax, N., Verenich, I., La Rosa, M., & Dumas, M. (2017). Predictive business process monitoring with LSTM neural networks. In *Advanced Information Systems Engineering: 29th International Conference, CAiSE 2017* (pp. 477–492). Springer.
- [5] Teinmaa, I., Dumas, M., La Rosa, M., & Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Transactions on Knowledge Discovery from Data*, 13(2), Article 17, 1–57.
- [6] Verenich, I., Dumas, M., La Rosa, M., Maggi, F. M., & Teinmaa, I. (2019). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology*, 10(4), Article 34, 1–34.
- [7] Rama-Maneiro, E., Vidal, J. C., & Lama, M. (2023). Deep learning for predictive business process monitoring: Review and benchmark. *IEEE Transactions on Services Computing*, 16(1), 739–756.

- [8] Sato, D. M. V., De Freitas, S. C., Barddal, J. P., & Scalabrin, E. E. (2022). A survey on concept drift in process mining. *ACM Computing Surveys*, 54(9), Article 189, 1–38.
- [9] Elkhawaga, G., Abuelkheir, M., Barakat, S. I., Riad, A. M., & Reichert, M. (2020). CONDA-PM—A systematic review and framework for concept drift analysis in process mining. *Algorithms*, 13(7), Article 161.
- [10] Adams, J. N., van Zelst, S. J., Rose, T., & van der Aalst, W. M. P. (2023). Explainable concept drift in process mining. *Information Systems*, 114, Article 102177.
- [11] Kraus, A., & van der Aa, H. (2025). Machine learning-based detection of concept drift in business processes. *Process Science*, 2, Article 5.
- [12] Darban, Z. Z., Webb, G. I., Pan, S., Aggarwal, C. C., & Salehi, M. (2024). Deep learning for time series anomaly detection: A survey. *ACM Computing Surveys*, 57(1), Article 15, 1–42.
- [13] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115.
- [14] Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): Overview, definition, and architecture. *IEEE Access*, 11, 31866–31879.
- [15] Birkstedt, T., Minkkinen, M., Tandon, A., & Mäntymäki, M. (2023). AI governance: Themes, knowledge gaps and future agendas. *Internet Research*, 33(7), 133–167.
- [16] Peffers, K., Tuunanen, T., & Niehaves, B. (2018). Design science research genres: Introduction to the special issue on exemplars and criteria for applicable design science research. *European Journal of Information Systems*, 27(2), 129–139.