Social Science and Management
Print ISSN: 3007-6854
Online ISSN: 3007-6862
DOI: https://doi.org/10.61784/ssm4095

AN INTEGRATED PROCESS-MINING AND EXPLAINABLE MACHINE-LEARNING FRAMEWORK FOR BOTTLENECK AND INTERNAL-CONTROL RISK DETECTION IN THE MULTI-ENTITY FINANCIAL CLOSE

YaNan Jiao[1], YiWen Liang[2*]
[1]*Long Island University, Brooklyn 11201, NY, USA.*
[2]*Cornell University, Ithaca 14853, NY, USA.*
[*]*Corresponding Author: YiWen Liang*

Abstract: The financial close of a multi-entity group is a recurring, deadline-driven process in which operational congestion and internal-control weaknesses interact, yet the two are normally analysed by separate functions with separate tools. This paper proposes CLOSE-XPM, a framework integrating process mining and explainable machine learning to attribute close-cycle delay to individual record-to-report activities, to flag close instances carrying elevated internal-control risk, and to rank remediation actions against both objectives at once. Two contributions are introduced: a cause-aware Bottleneck Contribution Index (BCI-C) that decomposes every pre-activity delay into resource contention, calendar unavailability and residual, then propagates a contention-truncation counterfactual along a per-case critical path, yielding an estimate in calendar days saved; and a screening procedure that estimates the risk effect of capacity and control actions from the event log alone. Because no public financial-close log exists, we build MEFC-Sim, a resource-aware generator (30 entities, 24 periods, 14,187 events) in which weaknesses are injected as behaviour and the outcome depends on variables the log cannot reveal; paired counterfactual re-runs provide an oracle. BCI-C reaches 0.907 rank correlation with the oracle against 0.544 for the best observable baseline, and predicts the attainable reduction within 0.076 days. Risk detection reaches 0.828 AUC across unseen entities with a 2.86-fold lift in the top-20% inspected sample. Delay and control risk are strongly coupled (rho = 0.708): relieving the three dominant bottlenecks shortens the close by 1.70 days and cuts the deficiency rate by 22.7%. A ground-truth fidelity analysis shows that SHAP is faithful to the model but only moderately faithful to the data-generating process. Validation on a real public log of 1,434 cases confirms transferability and identifies a boundary condition: coupling requires waiting time to be contention-driven.
Keywords: Process mining; Explainable artificial intelligence; Internal control; Financial close; Bottleneck analysis; SHAP; Audit analytics

1 INTRODUCTION

The financial close is the recurring process through which a group of legal entities converts transactional records into consolidated financial statements. In a multi-entity group the same procedure is executed dozens of times per period, by different local teams, on heterogeneous systems, against a single group deadline. Two questions dominate management attention. The controller asks where the close loses time; the internal auditor asks which entity-period combinations carry a material risk that a control did not operate. In practice the two questions are answered by different functions, with different data extracts and different tools, and the answers are rarely reconciled.

This separation is costly, because the two phenomena are not independent. Congestion at a shared group function — an intercompany reconciliation hub, a treasury desk, a consolidation team — pushes local work into the last hours before the deadline, which is exactly when reviews are skipped, approvals are self-signed and adjustments are posted outside business hours. Conversely, control failures generate rework: rejected consolidation packages, unresolved intercompany differences and post-close adjustments all lengthen the cycle. A framework that measures only one side therefore misattributes causes and misprioritises remediation.

Process mining offers the natural substrate for such an integrated analysis, because both phenomena leave traces in the same event data: timestamps carry performance information, while the identity, role and sequence of the actors carry control information. Process mining has been applied to performance analysis and to auditing, but largely in separate literatures, and the financial close specifically has received little attention despite being one of the most control-intensive and most deadline-sensitive processes in any group.

This paper makes four contributions.

1) A framework, CLOSE-XPM, that runs a performance track and an internal-control track over one multi-entity close event log and joins them through a coupling analysis and a common remediation ranking.

2) A cause-aware, criticality-aware bottleneck metric. We first decompose each pre-activity delay into resource contention, calendar unavailability and a residual, following the causal taxonomy of waiting time, and then propagate a contention-truncation counterfactual along a critical path reconstructed per case. The resulting Bottleneck Contribution

Index (BCI-C) is expressed in calendar days saved per close, which makes it directly comparable with a managerial target.
3) A quantitative evaluation methodology. Because the ground truth for a bottleneck metric is the effect of actually removing the bottleneck, we build a resource-aware generator of multi-entity close logs with common random numbers, so that a counterfactual re-run is paired with the baseline and provides an oracle. The same construction gives a ground-truth attribution for every risk mechanism, which lets us measure explanation fidelity rather than assert it.
4) An empirical characterisation of the delay–risk coupling, together with a model-based screening procedure that estimates, from the log alone, the joint effect of capacity and control interventions; the estimates are validated against the oracle and used to rank remediation portfolios under a budget.
We also report a negative and a boundary result. SHAP attributions aggregated to the mechanism level are only moderately correlated with ground-truth attribution, and the coupling that is strong in the simulated close does not appear in a real municipal log where waiting time is dominated by calendar unavailability rather than contention.

2 RELATED WORK

2.1 Performance and Bottleneck Analysis in Process Mining

Performance mining traditionally aggregates sojourn or waiting times per activity or per directly-follows edge [1-2]. Aggregation hides variability, which motivated the performance spectrum and its use for batching analysis and for stage-based analysis [3-4]. Queue mining models the delay caused by shared resources explicitly, and inference techniques recover unobserved queueing events in systems with shared resources [5-6]. A recent line of work classifies the direct causes of waiting time — batching, prioritisation, contention, resource unavailability and extraneous factors — and quantifies their contribution to cycle-time efficiency [7-8]. Extraneous delays have also been modelled to improve simulation fidelity [9]. What is still missing is a metric that combines cause attribution with criticality: an activity may accumulate large waiting times on a parallel branch that is never on the critical path, in which case removing that waiting saves nothing. BCI-C addresses exactly this combination.

2.2 Process Mining for Auditing and Internal Control

The use of event logs as audit evidence was proposed early and demonstrated in a field study on procurement [10-12]. Subsequent work formalised rule-based compliance checking with risk management, diagnosed where deviations occur, and studied the accounting-data structures from which financial event logs must be inferred, multilevel analysis for financial audits, the detection of absent internal controls, the reduction of false positives in fraud detection, the evaluation of internal-control effectiveness and the embedding of process mining into statutory audits [13-20]. This literature is predominantly conformance-oriented and rule-based; it does not quantify how much of the residual risk is explained by operational congestion, nor does it rank remediation jointly with efficiency objectives.

2.3 Explainable Predictive Process Monitoring

Outcome-oriented predictive process monitoring is a mature field with established benchmarks, and gradient boosting remains a strong baseline against deep models on tabular case encodings [21-23]. Post-hoc explanation with LIME and SHAP has been transferred to predictive process monitoring, and explanations have been shown to help improve models [24-29]. Evaluations of these explanations are usually indirect — perturbation-based faithfulness or human studies — because the true importance of a feature is unknown on real logs. Prescriptive extensions optimise interventions for cycle-time reduction and use causal machine learning on event logs [30-31]. Our simulator makes the generative structure known, which turns explanation fidelity into a measurable quantity, and our screening procedure extends prescriptive monitoring to a mixed action space of capacity and control measures.

3 THE CLOSE-XPM FRAMEWORK

3.1 Multi-Entity Close Event Log

Let E be a set of legal entities and P a set of fiscal periods. A close instance, and therefore a case, is a pair (e, p) in E × P. An event is a tuple (c, a, r, ts, tc, x) where c is the case, a an activity of the record-to-report procedure, r the executing user, ts and tc the start and completion timestamps, and x a vector of payload attributes such as the journal-entry amount and the posting/effective date pair. Entity-level attributes (region, tier, ERP system, local-GAAP requirement) and period-level attributes (quarter-end, year-end) are case attributes. Cycle time is measured from the period-end cut-off, not from the first event, because the deadline is defined relative to the period end.
Two structural properties of this setting drive the method. First, all entities close in the same narrow window, so entity-local resources and group-shared functions are contended simultaneously; congestion is therefore an inter-case phenomenon. Second, the procedure contains parallel regions (sub-ledger closes; intercompany dispute resolution against FX revaluation; local-GAAP adjustment against variance analysis), so waiting is not automatically on the critical path.

3.2 Cause-Aware Delay Decomposition

For every event we determine its binding predecessor inside the case: the event with the latest completion time among those that complete no later than the event starts. The pre-activity gap g is the elapsed time between that completion and the event start. We split g into three parts. The contention part g_con is the amount of the gap during which the assigned resource was executing an event of another case; it is computed by intersecting the gap window with the busy intervals of that resource elsewhere in the log. The unavailability part g_una is the amount of the remaining gap that falls outside the working calendar. The residual g_res = g − g_con − g_una absorbs batching, hand-over and extraneous delay. When a log records only one timestamp per event, service intervals are first imputed per resource as the time since that resource last completed an event, truncated at a maximum service length; the decomposition then applies unchanged.

3.3 Bottleneck Contribution Index

For a case c we reconstruct the temporal DAG whose nodes are the events, whose edges connect each event to its binding predecessor, whose node weights are the observed durations and whose edge weights are the observed gaps. Replaying the DAG with the observed values reproduces the observed case end exactly, which makes the construction a valid baseline for counterfactual replay. Let q be the 25th percentile of g_con over all occurrences of activity a in the log. The counterfactual replaces, for every occurrence of a in c, the gap g by g − g_con + min(g_con, q) and recomputes the longest path. Writing CP(c) for the observed critical-path length and CP_a(c) for the counterfactual one,

$$\text{BCI-C}(a) = \frac{1}{|C|}\sum_{c \in C}\left[\text{CP}(c) - \text{CP}_a(c)\right], \tag{1}$$

expressed in days per close instance, where the sum runs over the cases that contain a and the average is taken over all cases, so that rare activities are not over-weighted. Truncating the full gap instead of its contention part yields the ablated variant BCI, which we report as a baseline in order to isolate the contribution of the cause decomposition. The index is an upper bound on what capacity or scheduling measures can achieve, because it holds all other delays fixed and ignores the extra load that an accelerated case imposes downstream; Section 5.3 quantifies the size of that approximation error against the oracle.

3.4 Control-Conformance Feature Encoding

Each case is encoded into four interpretable blocks. Block C covers control conformance: preparer–approver identity for journal entries (segregation of duties), absence of a review event before approval (four-eyes principle), post-close adjustments after sign-off, counts of resubmissions and intercompany disputes, share of off-hours events, off-hours approvals, backdated postings, the largest journal-entry amount and its ratio to the approver's authority limit, and the share of events with a missing user identifier. Block O covers the organisational perspective: number of distinct users, Herfindahl concentration of work over users, hand-over count and ratio, number of roles and the share of work performed by group-level resources. Block P covers performance: span, total gap, total contention, total unavailability, total service time, gap ratio, maximum gap, event and activity counts, a rework ratio, the contention accumulated at each of the three shared group functions, and the within-period percentile of the total gap, which encodes the inter-case position of the entity. Block X covers context: tier, region, ERP system, local-GAAP flag, quarter-end and year-end flags and the period index.

3.5 Explainable Risk Model and Coupling

A gradient-boosted tree ensemble is trained to predict whether the close instance carries a material internal-control deficiency, and TreeSHAP supplies exact additive attributions. Two coupling quantities link the tracks. The risk lift of activity a is the ratio between the deficiency rate among cases whose contention at a lies in the upper tertile and the overall deficiency rate. The Bottleneck–Risk Coupling Index normalises BCI-C and the excess risk lift to a common scale and averages them, giving a tunable joint priority.

3.6 Model-Based Counterfactual Screening of Remediation Actions

An action is either a capacity action, which truncates the contention before one activity at its 25th percentile, or a control action, which enforces a control (mandatory review, enforced segregation of duties, hard close, enforced authority limits). For a capacity action the expected delay effect is BCI-C(a) and the expected risk effect is obtained by editing the affected performance features of every case according to the same counterfactual replay and re-scoring the risk model. For a control action the delay effect is unknown to the analyst and set to zero, and the risk effect is obtained by setting the corresponding control feature to its compliant value and re-scoring. Both estimates use only the event log and the trained model, so the procedure is applicable in the field, where no simulator is available. Portfolios of k actions are then ranked by the joint index, see Figure 1.

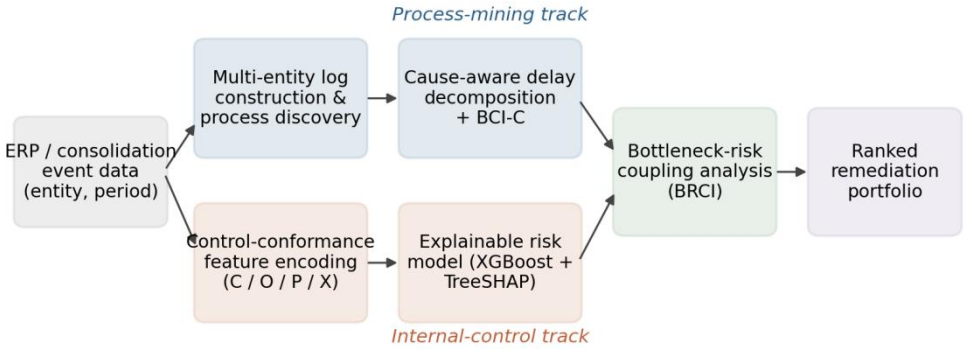

Figure 1 CLOSE-XPM

Note: A single multi-entity close event log feeds a performance track (cause-aware delay decomposition and BCI-C) and an internal-control track (interpretable case encoding and an explained risk model). The two tracks are joined by a coupling analysis and produce one ranked remediation portfolio.

4 EXPERIMENTAL SETUP

4.1 MEFC-Sim: A Generator with Ground Truth

No public event log of a financial close exists, and proprietary logs cannot be released, so evaluation of any close-specific method requires a generator. MEFC-Sim is a discrete-event simulator whose design is driven by the requirement that the ground truth be usable, not by the requirement that the data look complicated.

4.1.1 Structure

Twenty-two record-to-report activities form a DAG with three parallel regions and two rework loops (intercompany dispute resolution; group rejection and resubmission). Thirty entities in three regions and three size tiers close 24 consecutive monthly periods. Entity-local preparer, reviewer and approver pools and four group-shared pools (intercompany hub, treasury, group validation, group controlling) have finite capacity, so congestion emerges from concurrent closes instead of being parameterised. Service times are log-normal in working hours; work is normally confined to a Monday–Friday, 09:00–18:00 calendar, with a probability of deliberate out-of-hours working that increases with a weak control environment and at quarter- and year-end.

4.1.2 Control weaknesses as behaviour

Eight mechanisms are injected by altering behaviour, never by writing a label: self-approval by the preparer, a skipped review event, a post-close adjustment after sign-off, approval above the authority limit, backdated postings, excess rework, out-of-hours activity, and an intercompany difference beyond tolerance. Their probabilities depend on a latent entity-level control-environment quality, on tier and on period type.

4.1.3 Outcome

The audit outcome is drawn from a logistic model over the injected mechanisms plus standardised congestion and cycle time, an interaction between self-approval and congestion, and two latent entity variables (control-environment quality and staff experience) that are not recoverable from the log. Perfect prediction is therefore impossible by construction. Logging is imperfect: 4% of user identifiers are missing, 5% of review events are not logged, and completion timestamps carry Gaussian jitter, so the observable control indicators are noisy proxies — the segregation-of-duties indicator agrees with the injected mechanism on 99.0% of cases and the four-eyes indicator on 96.0%.

4.1.4 Oracle

Service times and out-of-hours decisions are drawn from per-event random streams keyed by case and task, and case plans are drawn before execution, so a counterfactual re-run differs from the baseline only through the intervention. Comparing a re-run with the baseline therefore measures the causal effect of the intervention on mean cycle time and on the deficiency rate, including all knock-on congestion effects that a log-derived estimate cannot see.

4.2 Real-Life Event Log

For external validity we use the public log of the receipt phase of an environmental permit application process of a Dutch municipality, collected in the CoSeLoG project [32]. It records 8,577 events in 1,434 cases over 27 activities, 48 users and 7 organisational units, and it carries a statutory deadline per case, which yields an objective, log-derivable compliance outcome: 6.42% of cases breach the deadline. The log is not a financial close, but it shares the properties the framework relies on — finite shared human resources, check/determine pairs that implement a four-eyes control, organisational units, and a deadline-driven outcome — and it records a single timestamp per event, which exercises the service-time imputation of Section 3.2.

4.3 Protocols, Baselines and Metrics

Bottleneck metrics are compared against the oracle over the eleven activities whose mean gap exceeds one hour, using Spearman and Kendall rank correlation, precision at 3, NDCG at 5, the regret of the top-1 choice, and — for the two indices expressed in days — mean absolute error and R-squared against the oracle effect. Baselines are the mean, total

and maximum pre-activity gap, mean sojourn time, frequency times mean gap, the directly-follows waiting time, the mean and total contention time, and, as a non-observable upper reference, the simulator's own queueing delay.
Risk detection is evaluated with a temporal split (periods 1–16 for training, 17–24 for testing) and with five-fold leave-entities-out cross-validation, which is the relevant generalisation question in a group with new or acquired entities. We report AUC, area under the precision-recall curve, best F1, the Brier score and precision, recall and lift in the top 20% of the inspected sample, which is the quantity an audit planner actually uses. Models are L2 logistic regression, a depth-limited decision tree, random forest, extra trees, a multilayer perceptron, XGBoost and, as a label-free baseline, isolation forest. XGBoost hyper-parameters are selected by grouped four-fold cross-validation inside the training split only.
Explanations are evaluated in four ways: rank agreement between mechanism-aggregated mean absolute SHAP and the ground-truth counterfactual attribution of each mechanism under the generative model; a deletion test that masks the k most important features and compares the AUC decay with random masking; rank stability over twenty bootstrap models; and detection of the injected interaction through SHAP interaction values.

4.4 Implementation

The pipeline is implemented in Python with pm4py 2.7 for discovery and conformance, scikit-learn 1.8, XGBoost 3.3 and shap 0.52. All code, the generated log, and every result table are released with the paper.

5 RESULTS

5.1 Logs and Discovered Models

Table 1 summarises both logs. The generated close log contains 14,187 events in 720 close instances, 19.7 events per case and 78 distinct variants. Mean cycle time is 13.81 days after period end, with a 90th percentile of 19.85 days, which is in the range reported for groups that have not yet industrialised their close. The deficiency rate is 19.6%. Inductive mining with a 0.2 noise threshold recovers a model with token-based fitness 0.995 and precision 0.809, confirming that the log is structured but not trivial. Heterogeneity across entities is substantial: the deficiency rate ranges from 11.4% to 40.6% across size tiers, and quarter-end and year-end closes are 2.5 and 3.9 days longer than month-end closes respectively.

Table 1 Characteristics of the Two Event Logs

Property	MEFC (generated)	CoSeLoG receipt (real)
Events	14,187	8,577
Cases	720	1,434
Activities	22	27
Resources	140	48
Organisational units	30 entities	7 groups
Trace variants	78	116
Median cycle time	13.73 d	0.03 d
90th pct. cycle time	19.85 d	14.10 d
Waiting share of flow time	46.9%	93.7%
Positive-outcome rate	19.6%	6.42%

Note: The real log records one timestamp per event; its median cycle time is dominated by cases completed within a single day.

5.2 Where the Close Loses Time

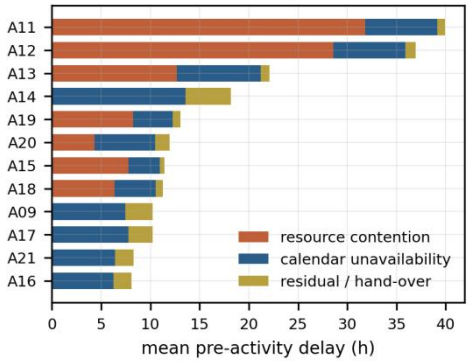

Figure 2 Causal Composition of the Mean Pre-Activity Delay for the Twelve Activities with the Largest Delay in the Generated Close Log
Note: Activities with similar total delay differ completely in composition.

Waiting accounts for 46.9% of flow time in the generated close. Its causal composition is 43.0% resource contention, 45.3% calendar unavailability and 11.7% residual. Figure 2 shows that the composition differs sharply across activities: intercompany reconciliation and dispute resolution are dominated by contention at the shared hub, whereas trial-balance preparation, consolidation submission and sign-off accumulate large gaps that are almost entirely calendar effects. This distinction is invisible to any aggregate waiting-time metric, and it is what separates a real bottleneck from an activity that merely happens to sit before a weekend.

5.3 Bottleneck Attribution Against the Oracle

Table 2 reports the agreement of each metric with the oracle over the eleven candidate activities. Metrics that use the raw gap perform poorly: the mean gap reaches rho = 0.399, the total gap 0.295, the maximum gap 0.246 and the directly-follows waiting time 0.385; the best raw-gap baseline is the mean sojourn time at 0.544. Restricting attention to the contention component raises rank agreement to about 0.90, and combining the cause decomposition with the critical-path counterfactual gives BCI-C rho = 0.907, Kendall tau = 0.769 and NDCG at 5 = 0.974, statistically indistinguishable from the 0.922 obtained by the simulator's own queueing delay, which is not observable in a real log. The ablated BCI, which truncates the whole gap rather than its contention part, collapses to 0.282, confirming that the cause decomposition rather than the critical-path replay carries the ranking accuracy.

The critical-path replay contributes something the rate-based metrics cannot provide, namely a calibrated magnitude. Figure 3(a) plots both indices against the oracle effect. BCI-C predicts the attainable cycle-time reduction with a mean absolute error of 0.076 days and R-squared 0.747 relative to the oracle, against 0.167 days and 0.120 for the ablated variant; the remaining bias is the expected over-estimation caused by ignoring the load that accelerated cases impose downstream. No baseline is on the day scale at all, so none can be compared with a managerial target such as "close two days earlier".

Table 2 Agreement of Bottleneck Metrics with the Oracle (11 Candidate Activities)

Metric	Spearman ρ	Kendall τ	P@3	NDCG@5	Top-1 regret (d)	MAE (d)	R^2
Mean pre-activity gap	0.399	0.281	0.67	0.868	0.000	—	—
Total pre-activity gap	0.295	0.187	0.67	0.898	0.000	—	—
Maximum pre-activity gap	0.246	0.175	0.00	0.373	0.603	—	—
Mean sojourn time	0.544	0.456	0.67	0.887	0.000	—	—
Frequency × mean gap	0.295	0.187	0.67	0.898	0.000	—	—
Directly-follows waiting time	0.385	0.304	0.67	0.855	0.000	—	—
Mean contention time	0.902	0.744	0.67	0.919	0.000	—	—
Total contention time	0.907	0.744	0.67	0.972	0.000	—	—
BCI (gap-based, ablation)	0.282	0.164	0.67	0.898	0.000	0.167	0.120
BCI-C (proposed)	0.907	0.769	0.67	0.974	0.000	0.076	0.747
Instrumented queueing delay*	0.922	0.792	0.67	0.875	0.143	—	—

Note: *Requires the simulator's internal queueing times and is therefore not computable from an event log; it is reported as an upper reference. MAE and R^2 are defined only for the two indices expressed in days.

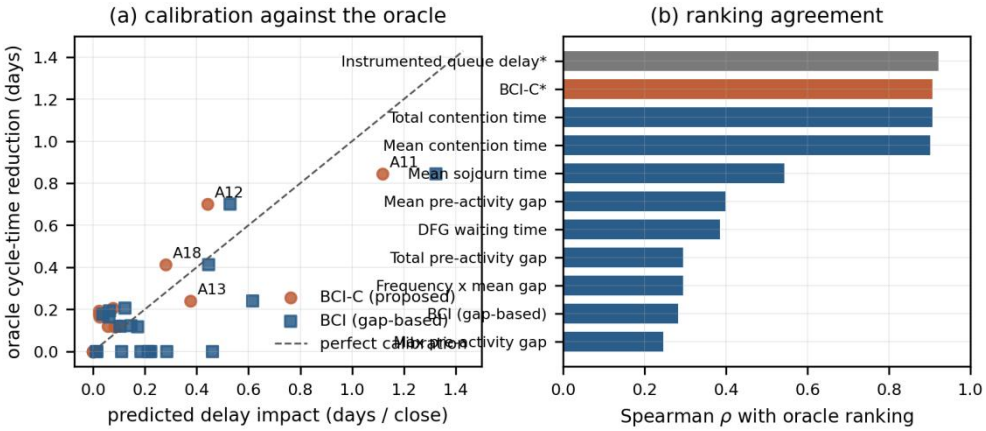

Figure 3 (a) Calibration of the Two Day-Scale Indices Against the Oracle Cycle-Time Reduction

Note: The dashed line is perfect calibration. (b) Rank agreement of all metrics with the oracle ranking; the entry marked with an asterisk requires instrumented queueing data.

5.4 Internal-Control Risk Detection

Table 3 reports detection performance. On the temporal split, logistic regression reaches 0.801 AUC, extra trees 0.800 and XGBoost 0.794; the isolation forest, which uses no labels at all, reaches 0.776, which is a useful reference for groups that have no history of audit findings. Under leave-entities-out cross-validation, which is the harder and more relevant protocol, extra trees reaches 0.828 ± 0.076 AUC and 0.611 area under the precision-recall curve against a

19.6% prior. The operational quantity is the lift: inspecting the top 20% of close instances ranked by XGBoost captures 57.1% of all deficiencies, a 2.86-fold lift over random selection. Restricting the encoding to the prefix up to consolidation submission — that is, predicting before management review and sign-off — costs 0.037 AUC and still gives a 2.45-fold lift, so the model is usable as an early warning rather than only as a post-mortem. Detection degrades gracefully with scarce labels: 48 labelled closes already give 0.704 AUC and 240 give 0.780.

Figure 4(c) reports the feature-block ablation under leave-entities-out cross-validation. Control-conformance features alone give 0.778 AUC and are the strongest single block; performance features alone give 0.760, well above the 0.693 obtained from context attributes only, which confirms that congestion carries genuine internal-control signal rather than merely reflecting entity identity. Combining the two blocks gives 0.807, and adding the organisational and context blocks does not improve AUC further (0.807) although it does improve the precision-recall area (0.579 against 0.542). The practical reading is that a compact encoding of control conformance plus congestion is sufficient, which matters because the omitted blocks are precisely the entity-identifying ones.

Table 3 Internal-Control Risk Detection on the Generated Close Log

Model	AUC	AUPRC	F1	Lift@20%	AUC, LEO (mean ± sd)	AUPRC, LEO
Logistic regression (L2)	0.801	0.549	0.598	2.55	0.816 ± 0.075	0.586
Decision tree (depth 5)	0.617	0.362	0.458	2.14	0.675 ± 0.061	0.338
Random forest	0.791	0.536	0.559	2.76	0.815 ± 0.071	0.561
Extra trees	0.800	0.576	0.588	2.55	0.828 ± 0.076	0.611
Multilayer perceptron	0.735	0.551	0.552	2.55	0.781 ± 0.082	0.556
XGBoost	0.794	0.562	0.588	2.86	0.807 ± 0.074	0.579
Isolation forest (unsupervised)	0.776	0.514	0.535	2.45	—	—

Note: The first four numeric columns use the temporal hold-out; LEO = five-fold leave-entities-out cross-validation. Prior deficiency rate 19.6%. Lift@20% is the ratio between the deficiency rate in the top 20% of the ranked sample and the overall rate.

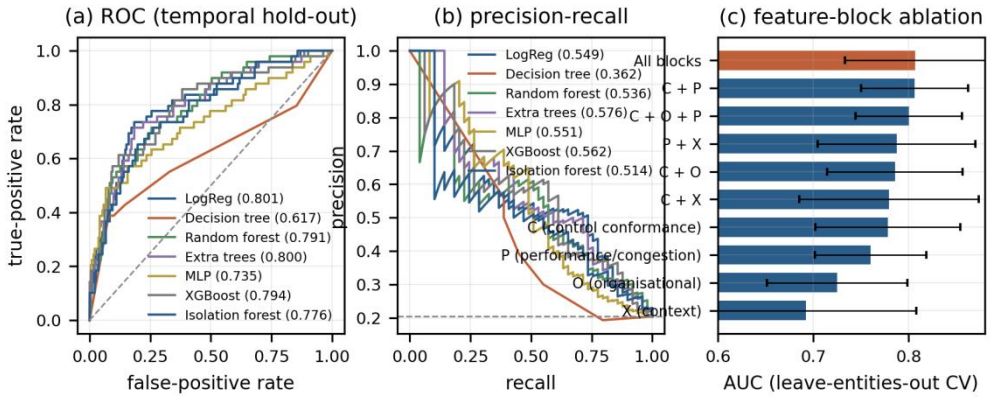

Figure 4 (a) ROC and (b) Precision-Recall Curves on the Temporal Hold-Out

Note: (c) Feature-block ablation under leave-entities-out cross-validation; error bars are the standard deviation across folds.

5.5 What the Explanations Are Worth

Figure 5 shows the SHAP summary of the deployed model. The largest contributions come from off-hours activity, elapsed span, congestion features and the intercompany dispute count, with the discrete control violations contributing smaller but non-zero amounts.

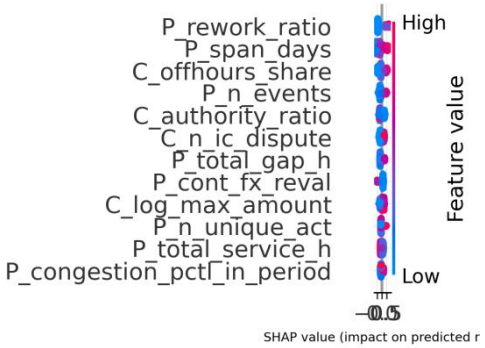

Figure 5 SHAP Summary for the Control-and-Performance Model on the Temporal Hold-Out (Twelve Most Important Features)

Because the generative model is known, we can go further and measure fidelity. Table 4 compares mechanism-level attributions with the ground truth, defined as the mean absolute change of the true risk probability when the mechanism is reset to its population baseline. Three findings emerge. First, the raw generative coefficients are an invalid reference: they correlate negatively with every attribution method (rho between −0.40 and −0.83), because a coefficient multiplies a variable whose variance differs across mechanisms. Any fidelity study that compares importance scores with coefficients will reach the wrong conclusion. Second, against the correct reference, agreement is moderate: rho = 0.400 over all nine observable mechanisms and 0.667 once the amount-based authority-limit mechanism is excluded. That mechanism is the one whose observable proxies — the largest journal-entry amount and its ratio to the authority limit — are also proxies for transaction complexity, so the mapping from mechanism to feature is not injective and the aggregation over-credits it. Third, and in contrast, the deletion test shows that the explanations are highly faithful to the model: masking the k most important features reduces AUC far faster than masking k random features, with a mean gap of 0.097 AUC, and the global ranking is stable across bootstrap models (mean pairwise rho = 0.822 ± 0.059).

The gap between the two results is the substantive finding: SHAP recovers what the model uses, but the model itself under-weights rare control violations relative to their true causal contribution. Self-approval is the clearest case — the third most important mechanism in the ground truth, but the least important feature in the explanation. Figure 6(b) shows that this is largely a sample-size effect: on a 48-period replication, fidelity rises from 0.765 at 240 labelled closes to about 0.91 at 480–720 and then oscillates around 0.78–0.79 while AUC keeps improving from 0.825 to 0.873. Auditors should therefore treat attributions of rare violations from short histories as lower bounds, and should not conclude from a small SHAP value that a control does not matter. Consistently, the injected interaction between self-approval and congestion is not recovered: its best pair ranks 137th of 300, so the shallow ensemble selected by cross-validation captures the marginal effects but not the interaction.

Table 4 Mechanism-Level Explanation Fidelity

Mechanism	True attrib.	Σ mean|SHAP|	True |β|
Segregation-of-duties conflict	0.0175	0.007	1.35
Four-eyes bypass	0.0146	0.080	1.20
Approval limit breach†	0.0052	0.352	1.10
Post-close adjustment	0.0164	0.041	0.95
Backdated posting	0.0075	0.071	0.85
Rework / resubmission	0.0761	0.677	0.75
Congestion	0.0375	0.720	0.70
Off-hours activity	0.0417	0.301	0.60
Elapsed cycle time	0.0282	0.330	0.45
Spearman ρ with true attribution	—	0.400 / 0.667†	−0.550

Note: †The authority-limit mechanism has no injective observable proxy; the second ρ excludes it. The last column shows that generative coefficients are a mis-specified reference.

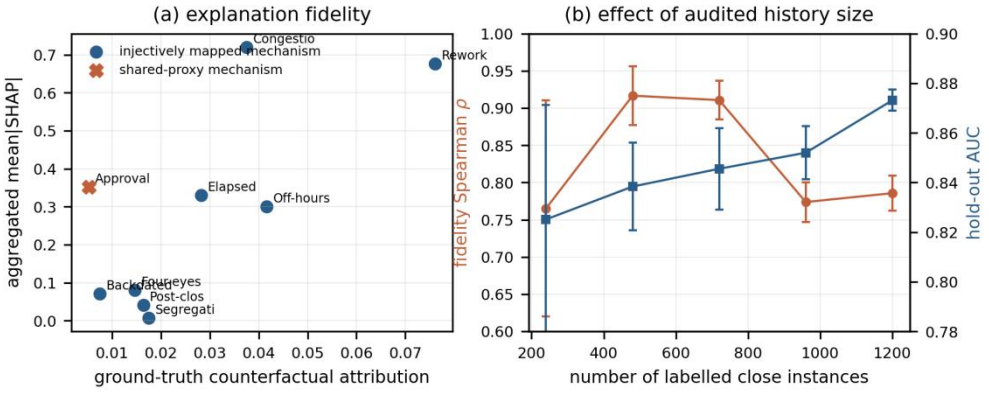

Figure 6 (a) Mechanism-Level SHAP Attribution Against Ground-Truth Counterfactual Attribution

Note: The cross marks the mechanism whose observable proxies are shared with transaction complexity. (b) Fidelity and predictive performance as a function of the number of labelled close instances (48-period replication).

5.6 Coupling and Remediation

Relieving contention is beneficial on both objectives. Across the eleven capacity actions, the oracle cycle-time reduction and the oracle risk reduction correlate at rho = 0.708, and BCI-C correlates with the observed risk lift at rho = 0.655. Efficiency and control are therefore not in tension in this process; congestion is a shared cause. Figure 7(a) makes the trade-offs concrete. The two intercompany actions and group validation lie in the upper right, improving both objectives. Control actions behave differently: a hard close is the single most effective risk measure (−6.5 percentage points) and even shortens the cycle by 0.31 days, whereas enforcing segregation of duties costs 0.17 days and enforcing authority limits 0.08 days, and mandatory review costs almost nothing while removing 3.9 percentage points of risk.

Enforcing the intercompany tolerance is the one action whose risk benefit is cancelled by the additional dispute-resolution work it creates.

The model-based screening of Section 3.6 recovers these effects from the log alone: its estimates correlate with the oracle at rho = 0.782 for the delay effect and 0.702 for the risk effect (0.669 for capacity actions, 0.600 for control actions). This is the practically important result, because it means an analyst can rank actions without a calibrated simulation model.

Table 5 evaluates portfolios of two, three and four actions. Ranking by the naive mean gap is uniformly the weakest: at budget four it still selects trial-balance preparation, whose entire delay is calendar effect, and gains nothing over budget three. BCI-C-based selection achieves the largest mean hypervolume (0.481 against 0.281 for the naive ranking, a 71% improvement); the joint index is best at budget three, where it substitutes group validation for FX revaluation and buys 1.4 additional percentage points of risk reduction at no cost in days, but it is not uniformly best. Under strong coupling this is expected: a delay-impact ranking already captures most of the available risk benefit, and the joint index pays off mainly when the budget is tight enough that the marginal capacity action is weak.

Table 5 Oracle-Evaluated Remediation Portfolios

k	Selection strategy	Days saved	Risk reduction	Hypervolume
2	Naive gap / BCI-C / risk / joint (identical)	0.96	7.8%	0.075
3	Naive mean gap	1.70	22.7%	0.385
3	Delay impact only (BCI-C)	1.70	22.7%	0.385
3	Risk reduction only	1.15	29.8%	0.344
3	Joint index (BRCI)	1.70	24.1%	0.409
4	Naive mean gap	1.70	22.7%	0.385
4	Delay impact only (BCI-C)	2.62	37.6%	0.985
4	Risk reduction only	1.68	39.0%	0.656
4	Joint index (BRCI)	1.68	39.0%	0.656

Note: Baseline: 13.81 days, 19.6% deficiency rate. Risk reduction is relative. Hypervolume is the product of days saved and relative risk reduction.

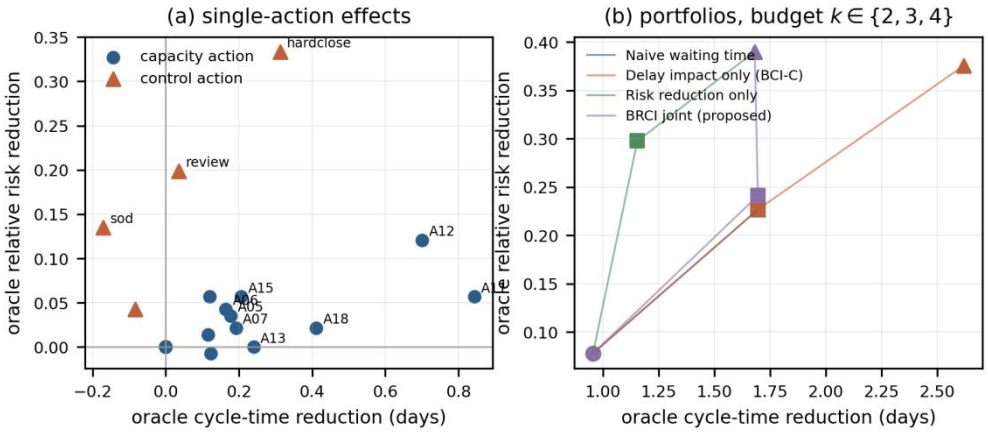

Figure 7 (a) Oracle Effect of Every Single Remediation Action on Cycle Time and on the Deficiency Rate

Note: Capacity actions cluster in the upper right, whereas control actions trade time against risk. (b) Portfolios selected by each strategy at budgets two, three and four.

5.7 External Validation and a Boundary Condition

On the real municipal log the pipeline runs end to end after service-time imputation. Waiting is 93.7% of flow time, but its composition is entirely different from the close: 73.2% calendar unavailability, 18.7% residual and only 8.1% contention. BCI-C consequently reorders the activities relative to the mean gap (rho = 0.484 between the two rankings), demoting the determination step whose large gap is almost all calendar effect. Risk detection for deadline breach reaches 0.705 AUC and 0.211 area under the precision-recall curve against a 6.4% prior, a 2.13-fold lift in the top 20%, and the prefix model using only the first three events loses almost nothing (0.691 AUC), which is the operationally useful configuration. The SHAP ranking is dominated by seasonality, elapsed span, the organisational unit and the maximum gap, with the four-eyes indicator contributing weakly.

The coupling effect, however, does not replicate: cases in the upper tertile of contention exposure have a deficiency lift of only 1.05 (Mann–Whitney p = 0.41), and the correlation between BCI-C and per-activity risk lift is −0.167. The decomposition explains why. Coupling requires that waiting time be contention-driven, because it is contention that concentrates work against the deadline; where waiting is idle calendar time, delay carries no control signal. This is a scope condition for the framework rather than a defect, and it is diagnosable in advance from the decomposition itself, before any risk model is trained, see Figure 8.

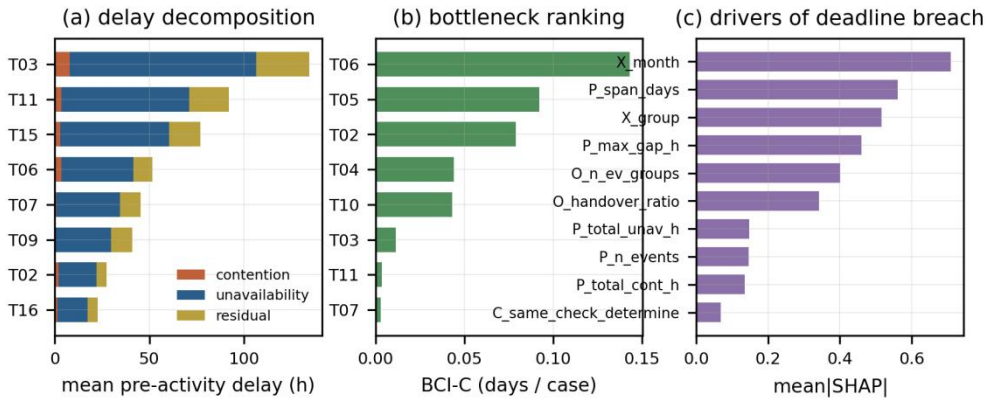

Figure 8 Real CoSeLoG Receipt Log
Note: (a) Delay decomposition, dominated by calendar unavailability. (b) BCI-C ranking. (c) SHAP drivers of statutory deadline breach.

6 DISCUSSION

6.1 Implications

Three results are directly actionable. First, aggregate waiting time is a poor guide to close acceleration: in our setting it identifies the right target only about 40% as consistently as a cause-aware index, and the activities it over-ranks are exactly those whose delay is calendar rather than contention. Any close-acceleration programme should decompose delay before allocating resources. Second, congestion should be part of the audit risk model. Performance features alone detect deficiencies well above context attributes, and adding them to control-conformance features raises AUC from 0.778 to 0.807 under the hardest protocol. Third, the two objectives are largely aligned where waiting is contention-driven, so the classical framing of a trade-off between a faster close and a well-controlled close is not supported here; the genuine trade-offs sit in specific control measures, such as enforced segregation of duties, whose time cost we quantify at 0.17 days.

6.2 Threats to Validity

The principal threat is that the quantitative claims about bottleneck attribution and coupling rest on a generator. This is unavoidable — the ground truth for a bottleneck metric is a counterfactual that cannot be observed in the field — but it bounds what can be concluded. The generator's structure, parameters and outcome model are released, and the mechanisms are drawn from standard internal-control taxonomies rather than fitted to a particular group, but the effect sizes we report are properties of this configuration. The real-log experiment is what tests transferability, and it does so only partially, since the outcome there is a statutory deadline breach rather than an audit finding. A second threat is that our control indicators are proxies; on a real log, missing users and unlogged reviews would be more prevalent than the 4% and 5% we inject. Third, the risk model is trained on case-level encodings, so it inherits the known limitations of that representation. Fourth, the screening procedure assumes that the risk model extrapolates correctly to counterfactual feature values, which holds only within the observed support; the correlations of 0.70–0.78 with the oracle should be read as evidence for ranking, not for point estimation.

6.3 Future Work

The close is naturally object-centric — journal entries, intercompany pairs, consolidation packages and entities are distinct objects with their own life cycles — and an object-centric formulation would remove the flattening we perform when we fix the case notion to an entity-period pair. Extending the screening procedure with causal, rather than predictive, estimates of intervention effects is the second obvious direction. Finally, a field study with a group audit function is required to establish whether the coupling observed here holds in practice and whether the ranked portfolios are accepted by controllers who face constraints that our action space does not model.

7 CONCLUSION

We presented CLOSE-XPM, a framework that analyses bottlenecks and internal-control risk in the multi-entity financial close from one event log, and joins the two through a coupling analysis and a common remediation ranking. Its central technical component, the cause-aware Bottleneck Contribution Index, attains 0.907 rank correlation with an oracle against 0.544 for the best log-observable baseline, and estimates the attainable cycle-time reduction to within 0.076 days. Risk detection reaches 0.828 AUC across unseen entities and a 2.86-fold lift in the inspected sample, and a log-only screening procedure recovers the oracle effect of candidate remediation actions at 0.70–0.78 rank correlation. Because the generator is released with known structure, we could also measure what explanations are worth: SHAP is

faithful to the model but only moderately faithful to the data-generating process, rare control violations are systematically under-attributed on short histories, and generative coefficients are an invalid fidelity reference. On a real public log the pipeline transfers but the coupling does not, and the delay decomposition explains why. Practitioners can use that decomposition as an inexpensive test of whether an integrated efficiency-and-control analysis is worthwhile in their process at all.

COMPETING INTERESTS

The authors have no relevant financial or non-financial interests to disclose.

REFERENCES

[1] van der Aalst W M P. Process Mining: Data Science in Action. 2nd ed. Berlin, Germany: Springer, 2016.
[2] Dumas M, La Rosa M, Mendling J, et al. Fundamentals of Business Process Management. 2nd ed. Berlin, Germany: Springer, 2018.
[3] Denisov V, Fahland D, van der Aalst W M P. Unbiased, fine-grained description of processes performance from event data. In: Business Process Management (BPM 2018), LNCS 11080. Cham, Switzerland: Springer, 2018: 139-157.
[4] Klijn E L, Fahland D. Performance mining for batch processing using the performance spectrum. In: Business Process Management Workshops (BPM 2019), LNBIP 362. Cham, Switzerland: Springer, 2019: 172-185.
[5] Senderovich A, Weidlich M, Gal A, et al. Queue mining for delay prediction in multi-class service processes. Information Systems, 2015, 53: 278-295.
[6] Fahland D, Denisov V, van der Aalst W M P. Inferring unobserved events in systems with shared resources and queues. Fundamenta Informaticae, 2021, 183(3-4): 203-242.
[7] Lashkevich K, Milani F, Chapela-Campa D, et al. Why am I waiting? Data-driven analysis of waiting times in business processes. In: Advanced Information Systems Engineering (CAiSE 2023), LNCS 13901. Cham, Switzerland: Springer, 2023: 174-190.
[8] Lashkevich K, Milani F, Chapela-Campa D, et al. Data-driven analysis of batch processing inefficiencies in business processes. In: Research Challenges in Information Science (RCIS 2022). Cham, Switzerland: Springer, 2022: 231-247.
[9] Chapela-Campa D, Dumas M. Modeling extraneous activity delays in business process simulation. In: Proceedings of the 4th International Conference on Process Mining (ICPM), 2022: 72-79.
[10] van der Aalst W M P, van Hee K M, van der Werf J M, et al. Auditing 2.0: Using process mining to support tomorrow's auditor. Computer, 2010, 43(3): 90-93.
[11] Jans M, Alles M, Vasarhelyi M. The case for process mining in auditing: Sources of value added and areas of application. International Journal of Accounting Information Systems, 2013, 14(1): 1-20.
[12] Jans M, Alles M G, Vasarhelyi M A. A field study on the use of process mining of event logs as an analytical procedure in auditing. The Accounting Review, 2014, 89(5): 1751-1773.
[13] Caron F, Vanthienen J, Baesens B. Comprehensive rule-based compliance checking and risk management with process mining. Decision Support Systems, 2013, 54(3): 1357-1369.
[14] Ramezani E, Fahland D, van der Aalst W M P. Where did I misbehave? Diagnostic information in compliance checking. In: Business Process Management (BPM 2012), LNCS 7481. Berlin, Germany: Springer, 2012: 262-278.
[15] Werner M. Financial process mining — Accounting data structure dependent control flow inference. International Journal of Accounting Information Systems, 2017, 25: 57-80.
[16] Werner M. Multilevel process mining for financial audits. IEEE Transactions on Services Computing, 2015, 8(6): 820-832.
[17] Werner M. Identifying the absence of effective internal controls: An alternative approach for internal control audits. Journal of Information Systems, 2019, 33(2): 205-222.
[18] Baader G, Krcmar H. Reducing false positives in fraud detection: Combining the red flag approach with process mining. International Journal of Accounting Information Systems, 2018, 31: 1-16.
[19] Chiu T, Jans M. Process mining of event logs: A case study evaluating internal control effectiveness. Accounting Horizons, 2019, 33(3): 141-156.
[20] Werner M, Wiese M, Maas A. Embedding process mining into financial statement audits. International Journal of Accounting Information Systems, 2021, 41: Art. 100514.
[21] Teinemaa I, Dumas M, La Rosa M, et al. Outcome-oriented predictive process monitoring: Review and benchmark. ACM Transactions on Knowledge Discovery from Data, 2019, 13(2): Art. 17.
[22] Verenich I, Dumas M, La Rosa M, et al. Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM Transactions on Intelligent Systems and Technology, 2019, 10(4): Art. 34.
[23] Kratsch M, Manderscheid J, Röglinger M, et al. Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction. Business & Information Systems Engineering, 2021, 63(3): 261-276.

[24] Ribeiro M T, Singh S, Guestrin C. 'Why should I trust you?' Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 1135-1144.
[25] Lundberg S M, Lee S-I. A unified approach to interpreting model predictions. In: Advances in Neural Information Processing Systems 30, 2017: 4765-4774.
[26] Lundberg S M. From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2020, 2(1): 56-67.
[27] Galanti R, Coma-Puig B, de Leoni M, et al. Explainable predictive process monitoring. In: Proceedings of the 2nd International Conference on Process Mining (ICPM), 2020: 1-8.
[28] Rizzi W, Di Francescomarino C, Maggi F M. Explainability in predictive process monitoring: When understanding helps improving. In: Business Process Management Forum (BPM 2020), LNBIP 392. Cham, Switzerland: Springer, 2020: 141-158.
[29] Mehdiyev N, Fettke P. Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring. In: Interpretable Artificial Intelligence: A Perspective of Granular Computing. Cham, Switzerland: Springer, 2021: 1-28.
[30] Bozorgi Z D, Teinemaa I, Dumas M, et al. Prescriptive process monitoring for cost-aware cycle time reduction. In: Proceedings of the 3rd International Conference on Process Mining (ICPM), 2021: 96-103.
[31] Bozorgi Z D, Teinemaa I, Dumas M, et al. Process mining meets causal machine learning: Discovering causal rules from event logs. In: Proceedings of the 2nd International Conference on Process Mining (ICPM), 2020: 129-136.
[32] Buijs J C A M. Receipt phase of an environmental permit application process ('WABO'), CoSeLoG project. Dataset. 4TU.ResearchData, 2014. DOI: 10.4121/uuid:a07386a5-7be3-4367-9535-70bc9e77dbe6.