Human Factors in Design, Engineering, and Computing, Vol. 238, 2026, 1931–1941
https://doi.org/10.54941/ahfe1008421

Detecting Information Leakage in Predictive Process Mining: A Three-Layer Evaluation Framework

Fathima Lodi[1] and Amir Schur[2]

[1]TTM Technologies, Farmingdale, NY, USA
[2]National University, San Diego, CA, USA

ABSTRACT

Predictive process monitoring aims to provide early warnings about process outcomes, yet published benchmarks frequently report near-perfect prediction scores on full process traces. This paper investigates whether those scores reflect genuine predictive ability or information already embedded in the event log. A simulated supply-chain dataset with 818,292 events across 25,000 linked cases was used to evaluate 276 model configurations under controlled conditions where ground truth causal labels were available. Three forms of information leakage were identified- activity names that encode outcomes, conformance features that overlap with prediction targets, and temporal variables that measure downstream effects rather than upstream causes. When these sources were retained, models achieved an AUC of 1.00. After leakage was controlled through prefix-based evaluation, scores fell to between 0.76 and 0.93. A diagnostic procedure with specific detection thresholds is proposed for each layer. The framework was examined on two external benchmark datasets from different domains, and all three leakage patterns were independently observed in both. These findings suggest that near-perfect classification scores in predictive process mining should be interpreted with caution, and that a systematic leakage audit should precede model comparison or deployment.

Keywords: Predictive process monitoring, Information leakage detection, Process mining, Prefix-based evaluation, Event log analysis, Process intelligence, Supply chain analytics

INTRODUCTION

Process mining extracts operational knowledge from event logs recorded by enterprise information systems. Over the past decade the field has expanded from process discovery and conformance checking into predictive analytics, where machine learning models trained on historical event sequences forecast future process outcomes (van der Aalst, 2016). Applications range from predicting service-level agreement violations to flagging anomalous cases for review. The appeal is that a model identifying a problematic case early enough allows the organization to intervene before the problem fully materializes.

Benchmark studies report area under the ROC curve (AUC) above 0.95 on standard event log datasets, particularly when models are trained on complete process traces (Teinemaa et al., 2019). Those numbers suggest a mature

Received September 25, 2026; Revised November 5, 2026; Accepted November 15, 2026; Available online December 1, 2026

© 2026 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

capability ready for operational deployment. Most published evaluations, however, overlook a structural problem with event log data.

Event logs record what happened. Activity names such as Reject Goods, Failed Quality Check, or Invoice Dispute describe the outcome in plain text. A model trained on a bag-of-words encoding of the full trace does not need to learn a predictive pattern. It reads the activity name. The score then reflects the model's ability to identify a label it was given rather than to anticipate an outcome that has not yet occurred. This is information leakage, where input features carry information about the target that would not be available at the time a prediction is needed.

Weytjens and De Weerdt (2022) demonstrated that activity-label encoding inflates accuracy on several benchmark datasets. That work identified the problem conceptually but did not decompose it into distinct, diagnosable layers or supply a step-by-step procedure with quantitative thresholds that practitioners could apply to their own data.

This paper proposes a three-layer framework for detecting information leakage in predictive process mining, where each layer corresponds to a different mechanism through which outcome information enters the feature set, together with a diagnostic procedure carrying specific numeric thresholds. The analysis uses a simulated supply-chain dataset with known causal dependencies, which permits controlled examination of each mechanism, and two external benchmarks then test whether the patterns generalize. The codebase, simulator, and evaluation pipeline are publicly available under MIT license.

RELATED WORK

Predictive Predictive process monitoring uses partial or complete event traces to forecast case-level outcomes such as the next activity, remaining time, or a binary deviation label (Teinemaa et al., 2019). The standard protocol trains a classifier on historical traces and reports performance on a held-out test set. A widely used variant truncates traces to a fixed prefix length and evaluates at each prefix, simulating predictions made at different points during execution.

Most published work trains and evaluates on the full trace, which produces the highest scores. Teinemaa et al. (2019) benchmarked several classifiers across multiple logs and reported AUC approaching 1.0 at longer prefix lengths. Those results were taken at face value for several years, and the possibility that the data structure itself inflated them received limited attention until Weytjens and De Weerdt (2022) showed that activity labels in standard benchmark logs encode outcome information. Their recommendation was to build unbiased benchmarks by removing leakage-inducing features, stopping short of a structured diagnostic procedure.

Process mining rests on three analytical pillars: discovery, conformance, and enhancement (van der Aalst, 2016). Discovery algorithms such as the Inductive Miner construct models from logs, conformance checking compares observed traces against those models and classifies each deviation as an insertion, skip, or repetition, and enhancement adds predictive capability on top. The leakage problem sits at the intersection of the latter two, because

conformance produces deviation counts derived from the same activities that define prediction targets at the enhancement stage. This structural dependency between two of the three pillars has not been examined in the literature.

A separate line of work on anomaly detection has developed autoencoder models that train only on conformant cases and flag deviations through reconstruction error (Nolle et al., 2018). That approach avoids some leakage risk by using no labeled target, but combining autoencoder features with conformance-derived features in a downstream classifier reintroduces the same mechanisms. Cross-organizational process mining has focused on discovery and conformance within individual processes (Jokonowo et al., 2018), while supply chain disruption studies have relied on network models with no connection to operational event log data. Neither has examined leakage in multi-process settings.

Object-centric process mining allows events to reference multiple object types simultaneously (Ghahfarokhi et al., 2021). The extension opens analytical possibilities and also introduces new leakage surfaces, since shared activities across process views can encode cross-process outcomes that remain invisible when each process is examined alone. The OCEL 2.0 standard provides the format for multi-object logs, but published work using it has not addressed the implications of feeding cross-process conformance features into predictive models.

The present work extends these contributions in two ways. It decomposes information leakage into three layers rather than treating it as a single phenomenon, each with a different mechanism and a different detection approach, and it provides a replicable diagnostic procedure with specific numeric thresholds for each layer.

METHODOLOGY

Dataset Construction

A controlled simulation environment supplied the primary dataset rather than an operational enterprise event log, because causal ground-truth labels are not available in publicly accessible event logs. The dependency rules were calibrated against documented supply chain patterns in the published literature, but simulation cannot reproduce the organizational behavior and data-quality problems of a live environment. Results should be read as evidence of framework behavior under controlled conditions rather than as expected operational performance. External validation on the OCEL 2.0 and BPI Challenge 2019 datasets partially offsets this by showing that the identified leakage patterns are observable outside the simulation.

The simulator generates 25,000 integrated cases across procure-to-pay (12 activities, 253,354 events), logistics (44 activities, 347,587 events), and accounts payable (11 activities, 217,351 events), totaling 818,292 events across 75,000 traces linked by a shared case identifier with verified vendor and amount consistency. Five rules govern propagation. A procurement rejection raises logistics rework probability from 0.62 to 0.82. A procurement issue raises AP invoice dispute probability from 0.147 to 0.414. Logistics rework adds three to ten days to AP processing. Multiple rework types compound

deviation severity. Procurement approval beyond seven days raises express escalation probability to 0.35.

Each case carries binary labels for four targets: process health (15.5% positive), procurement-caused rework (24.6%), procurement-caused invoice dispute (16.1%), and logistics-caused payment delay (30.3%). Labels are set only where the upstream deviation actually triggered the downstream effect through the elevated probability path, not where both merely co-occurred, a distinction no public dataset provides. The seed was fixed at 42, and validation confirmed case presence across all three logs, correct temporal ordering of every causal label, and agreement between observed rates and the parameter file at 100% integrity.

Feature Encoding

Four strategies converted event sequences into fixed-length vectors. Bag-of-words counts activity frequency per trace, producing 64 dimensions. Act2Vec applies positive pointwise mutual information and singular value decomposition to produce 32-dimensional activity representations, following the skip-gram equivalence shown by Levy and Goldberg (2014); averaging per process view yields a 96-dimensional case representation. Prefix-truncated encodings use only the first k events (k = 3, 5, 7, 10). Temporal features contribute 23 measurements, including inter-event durations, inter-process gaps, and cyclical hour-of-day and day-of-week transformations.

Classification and Evaluation

A multilayer perceptron with hidden layers of 64 and 32 nodes served throughout, using adaptive learning rate, L2 regularization at alpha = 0.01, early stopping with patience of 10, and class-weighted loss. Hyperparameters were frozen after preliminary experiments, so differences across the 276 configurations of encoding, prefix length, and target reflect input representation alone. Splits were 70/15/15, stratified across all four targets. AUC was the primary metric, since positive rates of 15.5% to 30.3% make accuracy misleading (Fawcett, 2006). McNemar's test with Bonferroni correction supported pairwise comparison, and five-fold cross-validation on the top three models returned standard deviations below 0.01.

Leakage Detection Approach

The central diagnostic compares full-trace performance against truncated prefixes. At k = 3 or k = 5 the deviation-encoding activities have not yet appeared, so a higher full-trace score under the same classifier and test set marks information drawn from outcome-revealing features rather than predictive patterns, and the size of that gap quantifies the leakage. Feature groups were then removed one at a time and the AUC change measured, isolating activity-name features, conformance-derived features, and downstream temporal features. This ablation assigns each portion of the gap to a specific group, which is how the three layers were separated.

RESULTS

When trained on complete traces, every encoding variant achieved AUC of 1.00 on all four prediction targets. Bag-of-words, Act2Vec, temporal, and combined encodings all produced identical perfect scores. The uniformity of this result across fundamentally different encoding strategies was the initial indicator that the performance came from the data structure rather than from any particular representation.

At prefix k = 3, where the model sees only the first three events per trace, the mean AUC across targets dropped to 0.752. At prefix k = 5, it recovered partially to 0.899. Table 1 shows the performance breakdown by target and prefix length for the best-performing encoding (Act2Vec with behavioral cluster features).

Table 1: AUC by prediction target and prefix length (Act2Vec + cluster encoding).

Prediction Target	Full Trace	k = 7	k = 5	k = 3	Gap
Overall clean (15.5%)	1.000	0.912	0.897	0.764	0.236
Rework (24.6%)	1.000	0.941	0.931	0.802	0.198
Dispute (16.1%)	1.000	0.928	0.917	0.771	0.229
Delay (30.3%)	1.000	0.763	0.751	0.672	0.328

The gap column shows the difference between full-trace AUC and prefix k = 3 AUC. Every target has a gap exceeding 0.19, confirming that a substantial portion of the full-trace performance came from information unavailable at early prefix lengths. The delay target has the largest gap at 0.328 and the lowest prefix-truncated score, a pattern examined further below.

Three Layers of Leakage

Systematic removal of feature groups traced the leakage to three distinct mechanisms, each operating through a different channel.

Layer 1: Activities such as Reject Goods, Failed Quality Check, and Restocked (Rework) appear only in deviant cases, so a bag-of-words encoding that includes them supplies a direct indicator of the target. An audit confirmed that no case labeled clean contains a deviation-encoding activity at full-trace length, meaning the vocabulary perfectly partitions clean from deviant cases. At k = 3 those activities have not yet occurred, the partition disappears, and the model must rely on early, non-descriptive patterns.

Layer 2: The alignment-based algorithm counts insertions, skips, and repetitions against a discovered model. Those counts derive from the same activities that define the targets, so an insertion count for Reject Goods proxies directly for whether a rejection occurred. Correlations between several conformance features and the targets exceeded r = 0.85.

Layer 3: AP processing duration correlated at r = 0.90 with the delay target, measuring the delay itself rather than a precursor to it. Removing AP-derived temporal features dropped delay-model AUC from 1.00 to 0.72, and the remaining activity-pattern and timing-gap features plateau near

0.75 across the MLP, an attention-based model, and a graph-based extractor, confirming the ceiling belongs to the data rather than to any model.

The delay target marks a boundary condition. Its causal mechanism runs through elapsed time, since logistics rework extends the timeline and pushes accounts payable past payment terms. That effect lives in timestamps and not in activity sequences, and no activity-based feature can predict a timing outcome the data does not encode at that level.

Other targets behave differently. At k = 3 the model sees roughly Create Requisition, Approve Requisition, and Create Purchase Order, with no rejection, rework, or dispute yet recorded. Stock rework is detectable at AUC 0.802 because behavioral clustering assigns early cases to archetypes correlated with downstream deviation. The model does not see the rejection. It recognizes a profile that historically leads to one. Cancellations surface by k = 5, when the decision point typically falls, while returns stay hidden until nearly the full trace completes.

These timelines carry operational weight. An alert at k = 3 catches stock rework before the procurement rejection occurs, and an alert at k = 5 catches cancellation-bound cases before the cancellation is logged. That the signals survive all three layers confirms genuine early-warning capability in event log data, weaker than the inflated scores suggest at AUC 0.76 to 0.93 rather than 1.00, but operationally actionable.

DIAGNOSTIC FRAMEWORK

The three layers can be checked in sequence using the procedure illustrated in Figure 1. The steps are ordered from the fastest and most common form of leakage to the most subtle.

Step 1. Compute AUC on full traces and at prefix k = 3 using the same classifier and the same test partition. If the gap exceeds 0.15, the dataset warrants further investigation. A gap below 0.15 suggests that leakage, if present, is not materially affecting performance estimates.

Step 2 (Layer 1). Scan the activity vocabulary for terms that describe the prediction outcome. Words such as reject, fail, cancel, rework, dispute, delay, or error in any activity name indicate that the bag-of-words encoding will contain direct outcome indicators. This check takes minutes and catches the most common form of leakage in event logs with descriptive activity names.

Step 3 (Layer 2). Compute Pearson correlation between every feature column and the target variable. Remove or flag any feature with r above 0.85. Conformance-derived features are the most common source at this layer because the alignment algorithm counts the same deviation activities that define the prediction labels. Features that pass this threshold may still be useful in a deployment context but should not be included in a predictive evaluation intended to demonstrate early detection capability.

Step 4 (Layer 3). Remove all temporal features derived from the down stream process (the process where the predicted outcome occurs) and recompute AUC. If the drop exceeds 0.10, those temporal features were measuring the effect of the outcome rather than providing information available before the outcome occurred.

Step 5. Report both full-trace and prefix-truncated AUC side by side. This transparency allows readers and practitioners to judge what portion of the reported performance reflects genuine predictive ability versus structural artifact.

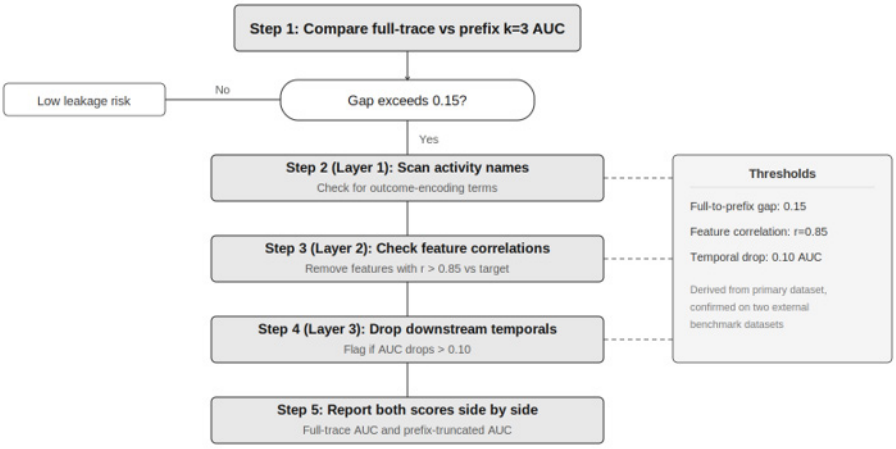


Figure 1: Diagnostic procedure for detecting information leakage in predictive process mining.

The thresholds of 0.15 for the full-trace-to-prefix gap, 0.85 for feature correlation, and 0.10 for temporal feature removal are derived from the magnitudes observed in the primary dataset and confirmed on the two external benchmarks described below. They are intended as practical guidelines rather than universal constants. Datasets with unusual properties, such as very short traces or highly skewed class distributions, may require adjustment.

EXTERNAL VALIDATION

Two publicly available benchmarks tested whether the leakage patterns are specific to the simulated data or reflect a structural property of event logs. Both differ from the primary dataset and from each other in process structure, activity vocabulary, class distribution, scale, and domain.

The OCEL 2.0 Order Management dataset (Ghahfarokhi et al., 2021) holds 21,008 events across 11 activities with many-to-many event-object relationships, each event referencing an average of seven objects, which tests the framework against structures more complex than the one-to-one case mapping of the primary dataset. Full-trace models scored near-perfect, the prefix gap exceeded the 0.15 threshold, Layer 1 leakage appeared in activity names describing order outcomes, and conformance features showed elevated correlations with the derived targets. Two of three cross-process dependency paths showed significant associations. Where the third did not, the framework reported no signal rather than a false positive, which matters because a diagnostic that finds leakage everywhere would be useless.

The BPI Challenge 2019 procurement dataset (van Dongen, 2019) holds 163,937 events across 36 activities with a class balance nearly opposite to the primary dataset, at 81.3% clean cases against 15.5%. Full-trace bag-of-words encoding reached AUC 1.00 on the rework target while prefix-truncated models dropped to realistic levels, five interpretable behavioral clusters emerged, and all three layers were independently observed.

Leakage in predictive process mining is therefore a structural property of how event logs are recorded. Any log in which activity names describe what happened, conformance features derive from deviation-encoding activities, or downstream temporal features measure the predicted outcome is susceptible, whatever its process type, industry, class distribution, or size.

DISCUSSION

Implications for Model Evaluation

The gap between full-trace and prefix-truncated AUC ranged from 0.198 to 0.328 across the four targets. For context, a researcher comparing two models on full traces might observe a difference of 0.02 or 0.03 and conclude that one is meaningfully better. The leakage contribution here is an order of magnitude larger than typical reported model differences, which suggests that in many published comparisons the variation attributed to modeling choices is smaller than the variation attributable to leakage. Model selection decisions based on full-trace evaluation may not hold under prefix-controlled conditions.

The three layers also call for different remediation, which matters for practitioners deciding where to invest effort. Layer 1 is the easiest to address by renaming activities to non-descriptive codes or switching from bag-of-words to learned embeddings that do not treat activity names as literal features. Layer 2 requires more care, since conformance features are often valuable for process analysis but create a circular dependency in prediction when the same deviation activities define both features and targets; the remedy is to reserve them for retrospective analysis. Layer 3 is the most subtle, because temporal features carry legitimate predictive information or leakage depending on whether they measure upstream causes or downstream effects, and telling the two apart requires domain knowledge of the process structure.

Implications for Emerging Technology Deployment

Process mining has transitioned over the past five years from an academic research area to a commercial technology sector. Platforms such as Celonis, Minit, and Apromore now serve enterprise customers across supply chain, healthcare, manufacturing, and financial services. Gartner included process mining in its emerging technology categories, and venture capital investment in the space has grown substantially. As the technology moves from research labs into operational settings, the accuracy claims that accompany platform deployments carry real financial consequences. A procurement department deciding whether to invest in a process mining platform for anomaly

detection will base that decision partly on the prediction accuracy the vendor demonstrates during evaluation.

The findings in this paper suggest that organizations evaluating such platforms should request prefix-truncated performance metrics alongside full-trace results. A platform that achieves AUC of 0.98 on full traces but drops to 0.72 at prefix k = 3 is not providing a 98% accurate early warning system. It is providing a 72% accurate system supplemented by information that would not be available in a real deployment. The gap between those two numbers has direct implications for return-on-investment calculations and for the level of trust an organization should place in the platform's alerts.

The diagnostic procedure proposed in this paper requires no commercial software and no specialized infrastructure. It runs on standard open-source libraries in Python. An organization can apply it to its own event logs in a single day before engaging with any vendor, establishing a baseline expectation for what genuine prediction accuracy should look like on its data. This baseline serves as a vendor-agnostic quality gate. It does not favor any particular platform. It asks a single question: is the reported accuracy real, or is it an artifact of how event logs are structured.

Relationship Between Feature Quality and Architecture

An incidental observation from the 276-configuration comparison was that the choice of input representation affected prediction performance substantially more than the choice of model architecture. Adding behavioral cluster membership features to the input raised AUC by 0.095, while replacing the standard classifier with an attention-based architecture raised it by 0.001. While a thorough investigation of this ratio is beyond the scope of this paper, the observation is consistent with findings in the broader tabular machine learning literature where feature engineering outperforms architectural sophistication on medium-sized structured datasets. For organizations deploying process mining, the practical implication is that time spent understanding process behavior through clustering and feature engineering is likely to produce larger performance gains than time spent selecting or tuning complex model architectures.

Table 2: Effect of removing each leakage layer on the delay prediction target.

Feature Configuration	AUC (Full Trace)	AUC(Prefix k = 5)	Gap
All features (including AP temporals)	1.000	1.000	0.000
Remove AP temporal features	0.754	0.722	0.032
Remove conformance features	0.751	0.718	0.033
Inter-process gaps only	0.742	0.710	0.032

Table 2 shows the delay target specifically, where the leakage effect is most pronounced. With AP temporal features included, the full-trace and prefix models both reach AUC of 1.000 because the temporal features encode the delay at every prefix length. Removing those features drops performance to

0.754 on full traces and 0.722 at prefix k = 5. The residual gap of 0.032 between full-trace and prefix scores after temporal leakage is removed reflects the contribution of Layer 1 and Layer 2 leakage in the remaining features. Once all three layers are controlled, the prefix and full-trace scores converge, confirming that the diagnostic procedure accounts for the identified leakage sources.

Application to Operational Settings

Consider an organization deploying a process mining platform to monitor repair operations across intake, diagnostics, parts procurement, and quality inspection. The platform vendor demonstrates AUC of 0.97 on historical repair data, suggesting that the system can predict repair delays with high accuracy. Applying Step 1 of the diagnostic, the organization computes AUC at prefix k = 3 and finds it drops to 0.68. The gap of 0.29 exceeds the 0.15 threshold. Step 2 reveals that several activity names in the repair log, such as Failed_Final_Test and Rework_Required, directly encode the deviation outcome. Step 3 identifies two conformance features with correlations above 0.85 against the delay target. Step 4 shows that removing downstream inspection timing features drops AUC by 0.14.

The diagnostic takes less than a day to run on standard Python libraries. The result is not that the platform is flawed, but that the reported 0.97 accuracy reflects information the system would not have during a real-time prediction. The honest early-warning accuracy is closer to 0.68. That is still operationally useful, but it sets a different expectation for how much trust the organization should place in the system's alerts, and it changes the return-on-investment calculation for the platform deployment. Without the diagnostic, the organization might commit resources based on an accuracy estimate that is 29 points higher than what the system would actually deliver.

CONCLUSION

The study identifies three mechanisms through which information leakage inflates prediction scores in process mining and proposes a diagnostic procedure with specific thresholds for detecting each one. The framework was validated on a controlled simulated dataset and confirmed on two external benchmarks drawn from different domains with different structural properties.

Two practical points follow. Near-perfect AUC on full process traces should prompt investigation rather than confidence, and the diagnostic described here runs in a single day on standard open-source tools, requires no specialized infrastructure, and returns a clear verdict on whether reported performance is genuine. Useful early-warning capability does survive the removal of leakage: stock rework was detectable by the third process step and cancellations by the fifth, at AUC between 0.76 and 0.93 depending on the target. Those figures reflect what a deployed system could realistically achieve.

Limitations remain. The primary dataset is simulated, with causal rules simpler than real operational dependencies, and the external validations confirm framework transfer but lack the causal ground truth needed to verify specific dependency claims. The attention and graph neural network components used random rather than trained weights, which limits the architecture comparison. Future work should apply the framework to enterprise logs with operational ground truth, implement end-to-end training, and test whether supplementary sources such as sensor or GPS data can break the delay prediction ceiling identified here.

ACKNOWLEDGMENT

The authors acknowledge the faculty of National University for their guidance and support throughout this research.

REFERENCES

Fawcett, T. (2006) An introduction to ROC analysis. Pattern Recognition Letters, 27(8), pp. 861–874.

Ghahfarokhi, A. F., Park, G., Berti, A., & van der Aalst, W. M. P. (2021). OCEL: A standard for object-centric event logs. Lecture Notes in Business Information Processing, 432, 169–175. https://doi.org/10.1007/978-3-030-85082-1_16.

Jokonowo, B., Claes, J., Sarno, R., & Rochimah, S. (2018). Process mining in supply chains: A systematic literature review. International Journal of Electrical and Computer Engineering, 8(6), 4626–4636. https://doi.org/10.11591/ijece.v8i6.pp4626-4636.

Levy, O. Goldberg, Y., & Dagan, I. (2014). Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics, 3, 211–225. https://doi.org/10.1162/tacl_a_00134.

Nolle, T., Seeliger, A. and Mühlhäuser, M. (2018). BINet: Multivariate business process anomaly detection using deep learning. Lecture Notes in Computer Science, 11080, pp. 271–287. Springer, Cham. https://doi.org/10.1007/978-3-319-98648-7_16.

Teinemaa, I., Dumas, 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. https://doi.org/10.1145/3301300.

van der Aalst, W.M.P. (2016) Process mining: Data science in action. 2nd ed. Berlin: Springer.

Van Dongen, B. F. (2019). BPI Challenge 2019 [Data set]. 4TU.ResearchData. https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1.

Weytjens, H. and De Weerdt, J. (2022). Creating unbiased public benchmark datasets with data leakage prevention for predictive process monitoring. In Business Process Management Workshops (BPM 2021) (LNBIP Vol. 436, pp. 18–29). Springer. https://doi.org/10.1007/978-3-030-94343-1_2.