Everything Real but the Drift: A Concept-Drift Benchmark Event Logs for Process Mining

Alessandro Padella[1], Francesco Vinci[1]

[1]Università degli studi di Padova, Via Trieste 63, Padova (PD), Italy

Abstract

In Business Process Management, multiple studies have addressed concept drift. A persistent challenge is the scarcity of event logs with verified drift instances and reliable ground truth. This limitation often hampers the evaluation of the proposed approaches. We address this gap by introducing real event logs enhanced with synthetic drift across different process perspectives. Logs have been manually modified following a deliberately process-tailored procedure, ensuring that no learnable pattern underlies the introduced changes and thus preventing approaches from overfitting to a specific drift-injection procedure. The logs are publicly available in XES standard format, ensuring tool compatibility.

Keywords

Process Mining, Concept Drift, Event Log, Online Learning

1. Introduction

Business Processes Management (BPM) underpin the operations of modern organizations where the process executions are routinely logged by information systems. Process mining uses such logged data to discover, monitor, and improve processes. As they evolve with regulatory, market, and organizational changes, insights from historical logs degrade. This phenomenon is known as *concept drift* [1]. A concept drift can be classified in four different types: *sudden*, *gradual*, *incremental* and *recurring* drift [2]. A *sudden drift* arises when a process is immediately replaced by a new one, with no period of coexistence, as typically happens in emergencies or following immediate regulatory changes. A *gradual drift* entails a transitional phase in which the old and new process run in parallel before the former is fully retired, a pattern often observed in company acquisitions. An *incremental drift* is the scenario where a substitution of a process with another one is done via smaller incremental changes. Finally, a *recurring drift* characterizes processes that cyclically or intermittently reappear, frequently driven by seasonal dynamics or external factors such as market conditions.

In the realm of process mining there has been an increasing attention on how to detect and deal with concept drift [2, 3]. However, despite the huge advancement that has been done in developing techniques that effectively detect and overcome concept drift, the availability of event logs to test is limited and often uncertain (cf. Section 2).

This study seeks to bridge this gap, by taking real event logs and enhancing them with a concept drift that may affect the control-flow or the time perspectives. The starting point is a real process from which an event log has been extracted. Then, it has been further enhanced through a process-specific technique designed to ensure the benchmark contains no underlying logic that a framework could learn. The resulting drifted logs are available on Zenodo at https://doi.org/10.5281/zenodo.21005862.

2. Innovation of the Resource in the Business Process Management

Concept drift has emerged as a key challenge in BPM, motivating a growing body of research to detect and handle it [2, 3, 1, 4]. These methods span several process perspectives and application areas:

Proceedings of the BPM 2026 Best Dissertation Award, Doctoral Consortium, and Demonstration & Resources Forum (BPM 2026) , Toronto, Canada, September 27th to October 2nd, 2026.
alessandro.padella@unipd.it (A. Padella); francesco.vinci@unipd.it (F. Vinci)

control-flow [1] and multi-perspective [4] drift detection, and the handling of drift in downstream tasks such as predictive process monitoring [5] and business process simulation [6].

Evaluating a drift-related approach requires event logs in which the presence, location, and nature of the drift are known with certainty, yet such logs are scarce: in real data the drift is rarely labeled and frequently subtle and/or imprecise. The lack of a benchmark containing event log with a documented concept drift is not only incidental, but has been explicitly acknowledged as an open problem [1, 7].

We address this research gap by providing 45 drifted event logs derived from 5 real-life logs, each provided in different process drift characterizations (cf. Section 1).

3. Description of the Resource

In this work, we enrich real-life event logs to introduce concept drift while keeping the rest of the log faithful to reality. Specifically, we modify the control-flow and the time perspectives after a chosen temporal point, namely a change point, while the other perspectives are preserved. Each drift is applied to a targeted activity of a given process, that makes the change meaningful in that process's context, and is provided in three different shapes: *sudden, gradual* and *recurring*. Note that *incremental drift* (cf. Section 1) has been intentionally excluded following the observations in [1], who note that it can be observed as a composition of successive sudden drifts or as a form of gradual drift. The real life processes that have been considered pertain to:

- A purchase order handling process of a Dutch company, part of the Business Process Intelligence challenge (BPIC) 2019, containing 8,000 traces and 33 distinct activities.[1]
- A travel expense reimbursement process: the event log, part of the BPIC 2020, containing 6,886 traces and 19 distinct activities.[2]
- A bank account closure process, a public IBM Process Mining use case, containing 32,429 traces events and 15 distinct activities.[3]
- The workflow-related events of a loan application process, part of the BPIC 2012, containing 8,049 traces events and 6 distinct activities.[4]
- A domestic travel expense reimbursement process: the domestic declarations event log, part of the BPIC 2020, containing 10,500 traces and 17 distinct activities.[5]

The repository of the resource available at https://doi.org/10.5281/zenodo.21005862 contains the augmented event logs along their descriptions.

Time Perturbation. A time-perturbation drift rescales the execution duration of a single targeted activity by a multiplicative factor for every case after the change point, with k capturing both directions of change. A value $k > 1$ yields a *slow-down* of the activity duration, representing phenomena such as a new process bottleneck or reduced capacity. Conversely, value $k < 1$ yields a *speed-up*, modeling, e.g. automation techniques. In our collection k ranges in the set $\{2, 3, 4, 5, 6\}$ for *slow-downs* and in the set $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ for *speed-ups*, so that each real log is paired with drift-injection in both the directions. Each perturbation is further provided in three shapes defining how the change unfolds across the cases following the change point. In the sudden shape, k is applied to every case from a change point onward, in the gradual shape, it is phased in over a transition window, with a growing fraction of cases subjected to k until the change is fully in place. Finally, in the recurring shape, the perturbed and original regimes alternate, with k switched to 1, i.e. no concept drift injected, across successive intervals of cases.

[1] https://doi.org/10.4121/uuid:d06aff4b-79f0-45e6-8ec8-e19730c248f1
[2] https://doi.org/10.4121/uuid:895b26fb-6f25-46eb-9e48-0dca26fcd030
[3] https://github.com/IBM/processmining/tree/main/Datasets_usecases
[4] https://doi.org/10.4121/uuid:3926db30-f712-4394-aebc-75976070e91f
[5] https://doi.org/10.4121/uuid:3f422315-ed9d-4882-891f-e180b5b4feb5

Table 1
The drift event logs overview. For each process and perspective, the drift technique applied, the multiplicative factor (K) used for time drifts, and the enhanced activity targeted by the change are reported. Every variant listed is provided in three shapes: sudden, gradual, and recurring.

Drifted log	Drift technique	K	Enhanced activity
Time perspective			
request_for_payment_bpi20	slow-down	5	Payment Handled
request_for_payment_bpi20	speed-up	0.2	Payment Handled
domestic_declarations_bpi20	slow-down	4	Request Payment
domestic_declarations_bpi20	speed-up	0.3	Request Payment
loan_application_process_bpi12w	slow-down	6	W_Valideren aanvraag
loan_application_process_bpi12w	speed-up	0.1	W_Completeren aanvraag
bank_account_closure	slow-down	2	Service closure Req. (BO)
bank_account_closure	speed-up	0.5	Service closure Req. (BO)
purchase_order_handling_bpi19	slow-down	3	Record Goods Receipt
purchase_order_handling_bpi19	speed-up	0.4	Record Goods Receipt
Control-flow perspective			
request_for_payment_bpi20	substitute	—	FINAL_APPROVED SUPERVISOR → DIRECTOR
domestic_declarations_bpi20	delete	—	Declaration APPROVED (ADMIN)
loan_application_process_bpi12w	rare→mand.	—	W_Beoordelen fraude (fraud check)
bank_account_closure	delete	—	Pending Req. Reservation Closure
purchase_order_handling_bpi19	rare→mand.	—	Remove Payment Block

The time perturbation drift is injected by rescaling only the execution time of the targeted activity by the factor k and shifting that case's remaining events forward (or backward) in time accordingly. In processes that allow resource to perform multiple tasks during a same temporal window (e.g. in the bank account closure dataset), we apply this rescaling directly, since resources there may legitimately work in parallel. In single-resource processes, a longer activity ($k > 1$) can leave a resource still busy when it is next needed. We resolve this by reassigning the next activity to another resource able to perform it accordingly to mined roles, or, if none is free, by postponing it until the resource becomes available. Symmetrically, a *speed-up* ($k < 1$) that keeps the waiting times between activities constant may require a resource to start an activity earlier, before it has finished a previous one. We handle this conflict in the same way, by reassigning the activity to a free resource or, if none is available, by delaying its execution until the original resource becomes free.

Control-flow Delete and Insert. Control-flow drifts change which activities occur, and the collection covers three variants of this perspective: an activity that disappears from control-flow ($delete$), an activity that from rare becomes mandatory ($rare \rightarrow mandatory$), and an activity that has been replaced with a different one ($substitute$).

As with the time perturbations, every control-flow related drift is provided in the sudden, gradual, and recurring shape. The structural change is therefore applied to all cases at once from achange point, in the first case, phased in over a transition window in the second, or alternated with the original behavior across successive intervals of cases, in the recurring drift scenario.

An activity that disappears from control-flow ($delete$), such as a control or an approval that is dismissed by the stakeholder, a task made obsolete by automation, or a request that is no longer issued, is injected by removing every occurrence of the targeted activity from the affected cases after a change point, so that the remaining events keep their timestamps, the predecessor of the deleted activity is connected directly to its successor, and the resource distribution is left untouched since no events are added, as well as the timestamps.

Conversely, an activity that from rare becomes mandatory ($rare \rightarrow mandatory$), such as a newly mandated control like a fraud check or a step prompted by a policy or regulatory change, it is injected by adding the targeted activity right after a designated predecessor to every affected case after a change point that does not already contain it. The waiting and the execution times are sampled from the activity's real ones distribution and shifting the case's remaining events forward by that amount, so that arrivals and waiting times stay coherent and a change point exact, while the resource is drawn from the activity's frequency-weighted set of resources to preserve the resource distribution.

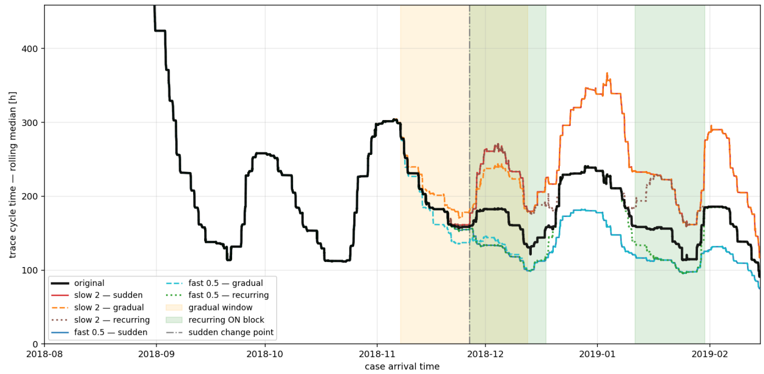

Figure 1: Time drift targeting Service closure Req. (BO), both directions and all three shapes. The curve reports the rolling median of trace cycle time over case-arrival time; slow-down (k=2) and speed-up (k=0.5) each given as sudden, gradual, and recurring.

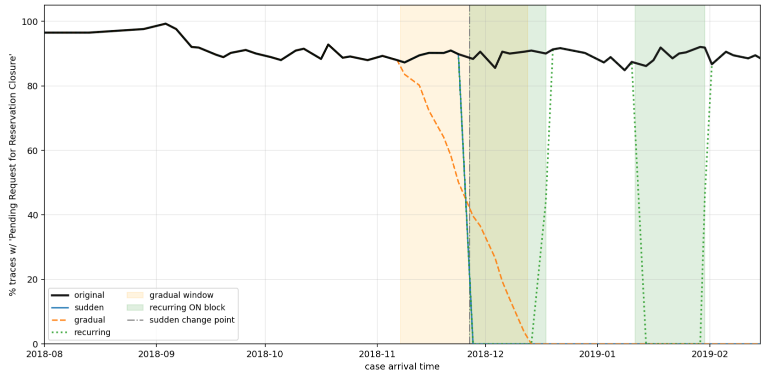

Figure 2: Control-flow delete drift targeting Pending Request for Reservation Closure, in its three shapes. The y-axis reports the percentage of traces containing the activity over case-arrival time: sudden (solid) drops to 0% at a change point, gradual (dashed) over the transition window, recurring (dotted) alternates regimes.

Finally, a replacement ($substitute$) of a given activity (e.g. *Approvement by the Supervisor* to *Approved by Director*), leaves the timestamps, the control-flow structure, and the resource distribution unchanged while altering only the activity label.

The complete list of processes, perspectives and activities targeted is reported in Table 1. The table reports, for each of the five original logs, the drift technique applied, the multiplicative factor K left blank when applied to control-flow drift, and the enhanced activity targeted by the change.

4. Preliminary Analysis

This section presents a preliminary analysis in one of the concept-drift augmented event log introduced in the previous section, i.e. the Bank Account Closure process.

For the time perspective, the targeted activity is *Service closure Req. (BO)*, which implies that the back-office of the bank has requested and approved the closure of the bank account. This activity may deal both with huge speed ups and slow-downs, since the back-office may be faster or slower depending on various external factors that can be, for instance, the automatization or the change of the procedures that resources operating in the bank has to learn from scratch. Figure 1 shows how the speed-up and the slow-down of the activity duration affects the whole trace duration (represented through the rolling median of the trace cycle time). Since this activity accounts for a large share of the trace duration in this process, rescaling its execution time by k propagates directly to the trace level: the slow-down brings the rolling median above the original event log, peaking around 360 hours against near 240 hours in the same window, while the speed-up reduces it symmetrically below them and pulls each case's remaining events earlier.

The control-flow perspective is examined through the removal of the activity *Pending Request for Reservation Closure*. Figure 2 illustrates the percentage of traces that contain the activity. This removal may be due to the full-automatization of that activity, making it a direct consequence of the previous one. The activity *Pending Request for Reservation Closure* is present in 1% of the original traces, and the delete drift removes it from all affected cases after the change point. In the sudden drift log the percentage of traces containing *Pending Request for Reservation Closure* immediately drop to 0% at the change point, while the gradual variant is phased out across the transition window, and the recurring variant alternates deletion and original behavior, so the curve falls to 0% and recovers to the original level. Throughout, the original log stays near its original level, confirming that the deviation is attributable to the injected drift rather than to background variability in the log.

5. Conclusion

This work is a first step towards equipping the Business Process Management field with reliable benchmarks for concept drift across the time and control-flow perspectives, without introducing learnable patterns. Current limitations, left to future work, are the absence of the injection of the concept drift on the resource perspective and of an explicit causal structure of the drift.

These event logs span several applications. In predictive process monitoring, they enable evaluating how well predictive models retain accuracy when processes undergo drift. In business process simulation, they allow assessing whether simulators reproduce the injected changes and generate synthetic logs that realistically capture process evolution, improving the fidelity of those models.

This paper's natural follow up is the development of a more systematic and generalizable methodology for injecting drift across all process perspectives with explicit causal structures, especially following the guidelines for drift characterization introduced in [1].

Declaration on Generative AI

The authors declare that Generative AI tools were used solely to assist with language refinement and not for the generation of scientific content or analysis.

References

[1] A. Kraus, H. van der Aa, Comprehensive characterization of concept drifts in process mining, Information Systems 135 (2026) 102584. doi:10.1016/j.is.2025.102584.

[2] R. P. J. C. Bose, W. M. P. van der Aalst, I. Zliobaite, M. Pechenizkiy, Handling concept drift in process mining, in: Advanced Information Systems Engineering - 23rd International Conference, CAiSE 2011, London, UK, June 20-24, 2011. Proceedings, volume 6741, Springer, 2011, pp. 391–405.

[3] D. M. V. Sato, S. C. De Freitas, J. P. Barddal, E. E. Scalabrin, A survey on concept drift in process mining, ACM Computing Surveys 54 (2021). doi:10.1145/3472752.

[4] E. L. Klijn, F. Mannhardt, D. Fahland, Multi-perspective concept drift detection: Including the actor perspective, in: G. Guizzardi, F. Santoro, H. Mouratidis, P. Soffer (Eds.), Advanced Information Systems Engineering, Springer Nature Switzerland, Cham, 2024, pp. 141–157.

[5] M. Maisenbacher, M. Weidlich, Handling concept drift in predictive process monitoring, in: 2017 IEEE International Conference on Services Computing (SCC), 2017, pp. 1–8.

[6] F. Vinci, G. Park, W. M. P. van der Aalst, M. de Leoni, Online discovery of simulation models for evolving business processes, in: Business Process Management, Springer Nature Switzerland, Cham, 2026, pp. 451–468.

[7] J. N. Adams, S. J. van Zelst, L. Quack, K. Hausmann, W. M. P. van der Aalst, T. Rose, A framework for explainable concept drift detection in process mining, in: Business Process Management, Springer International Publishing, Cham, 2021, pp. 400–416.