



## Research paper

## Outcome-oriented local explanation using context-aware process patterns



Mozhgan Vazifehdoostirani <sup>a</sup>,\* , Andrei Buliga <sup>b,c,\*</sup>, Laura Genga <sup>a</sup>, Chiara Di Francescomarino <sup>d</sup>, Massimiliano Ronzani <sup>b</sup>, Chiara Ghidini <sup>c</sup>, Remco Dijkman <sup>a</sup>

<sup>a</sup> Eindhoven University of Technology, Eindhoven, The Netherlands

<sup>b</sup> Fondazione Bruno Kessler, Trento, Italy

<sup>c</sup> Free University of Bolzano, Bolzano, Italy

<sup>d</sup> University of Trento, Trento, Italy

## ARTICLE INFO

## Keywords:

Explainable Predictive Process Monitoring

Local explanation

Context-aware process pattern

## ABSTRACT

Predictive Process Monitoring leverages machine learning models to predict the future states of business processes. Most Predictive Process Monitoring approaches rely on black-box models which, while powerful, lack interpretability, limiting their applicability in critical decision-making scenarios. Explainable Predictive Process Monitoring has emerged to address this gap, focusing on delivering actionable and transparent insights into predictions. Current approaches, however, often fail to incorporate multiple process perspectives and granularity levels in their explanations, overlooking crucial factors that influence process outcomes. This paper proposes a novel Explainable Predictive Process Monitoring approach to deliver explanations, integrating multiple process perspectives at various levels of granularity. The proposed approach addresses the limitations of existing methods, providing comprehensive, process context-aware explanations. The effectiveness of the proposed method is assessed through experimental evaluations on real-life event logs, showing how the integration of diverse process perspectives improves the interpretability and predictive insights of local explanations with improvements ranging from 2%–3% to 15%–20% depending upon the event log under analysis.

## 1. Introduction

Predictive Process Monitoring (PPM) encompasses a range of methods designed to predict the future status of ongoing executions (aka, cases) within a business process, helping to mitigate risks associated with undesired behaviours (Di Francescomarino and Ghidini, 2022). These models, often built using machine (including deep) learning techniques like Extreme Gradient Boosting (XGBoost) and Long Short-Term Memory (LSTM), are trained on historical event logs of business process executions (Teinemaa et al., 2019). Several studies have shown the effectiveness of exploiting machine learning techniques to build PPM models in various domains (Teinemaa et al., 2019; Rama-Maneiro et al., 2021; Márquez-Chamorro et al., 2017). However, the usage of these black-box techniques opens up challenges related to the explanation of how predictions are made (Rizzi et al., 2020a; Galanti et al., 2023).

Following a trend in Artificial Intelligence, where the popularity of very effective black-box machine and deep learning techniques has triggered the emergence of the eXplainable Artificial Intelligence (XAI) research area (Selbst and Powles, 2018), a novel research subfield has emerged in PPM. This subfield focuses on eXplainable Predictive Process Monitoring (XPPM), with the goal of tailoring XAI methods

to address specific challenges within the PPM domain (Stierle et al., 2021).

Recently, the XPPM literature has introduced various local and global explanation methods. Local explanations refer to explanations provided for specific cases, while global explanations aim to explain the overall behaviour of the black-box model, revealing its underlying mechanisms.

Concerning local explanations, different methods can be categorised into two main families: the family of *factual* explanations, where methods offer insight into the factors determining the obtained prediction (Rizzi et al., 2020a; Wickramanayake et al., 2023, 2022; Harl et al., 2020; Aversano et al., 2023), and the family of *counterfactual* explanations, where the methods meet the need to transform undesirable outcomes into desirable ones (Hundogan et al., 2023; Hsieh et al., 2021). Recent approaches have highlighted the benefits of creating a third family of approaches which integrates factual and counterfactual explanations (Huang et al., 2021; Buliga et al., 2024).

This paper focuses on local explanation methods, offering predictive insights to process analysis for a specific case by integrating factual

\* Corresponding authors.

E-mail addresses: [m.vazifehdoostirani@tue.nl](mailto:m.vazifehdoostirani@tue.nl) (M. Vazifehdoostirani), [abuliga@fbk.eu](mailto:abuliga@fbk.eu) (A. Buliga).

<https://doi.org/10.1016/j.engappai.2025.111387>

Received 17 December 2024; Received in revised form 14 April 2025; Accepted 3 June 2025

Available online 12 July 2025

0952-1976/© 2025 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

and counterfactual explanations. Our work starts from the observation that state-of-the-art approaches mainly focus on considering process executions (cases) from the *control-flow* perspective (where they consider a case only as a sequence of executed activities). They fall short of providing explanations that also include different process perspectives such as the *resource* perspective, the *time* perspective, and the *data* perspective, just to mention popular important aspects of business process executions. The impact of these limitations becomes particularly evident when examining real-life examples. As explored with a motivating example in Section 2, a closer look at existing XPPM methods, including LORELEY (Huang et al., 2021), CREATED (Hundogan et al., 2023), and the PAttern Based LOcal explanation (PABLO) framework (Buliga et al., 2024), reveals that they predominantly focus on control-flow or case attributes, often overlooking the potential of combining multiple perspectives for more actionable and insightful explanations. This gap in the current literature is particularly important given (i) the growing importance of considering Business Processes as complex artefacts composed of different perspectives (Mannhardt, 2018; Klijn et al., 2024; Adamo et al., 2017); and (ii) the ability of PPM techniques to forecast running process executions also considering the data payload of ongoing cases which represent perspectives that go beyond the control-flow (see (Di Francescomarino and Ghidini, 2022) for an overview).

This paper proposes a systematic approach to define and integrate multiple process perspectives in a combined process context-aware factual and counterfactual explanation. In the settings of this work, *context* refers to the aggregation of all process perspectives into a general one, including all available information related to a given process execution. These perspectives include information at different levels of granularity, namely at the level of a single activity (e.g., the throughput time of an activity), at the execution level (e.g., the throughput time of a complete execution), and at the level of *portions* of process executions or subprocesses (e.g., the throughput time of a subset of process activities). Interestingly, the last level is usually overlooked by existing approaches. In this study, we show how to extract meaningful features from these perspectives at different granularity levels and how to combine them in the generated explanations to uncover hidden drivers of process outcomes. Particularly, we extend the scope of explanation beyond isolated activities or cases by incorporating features that capture the complex interplay between activities and cases, thereby offering a more holistic understanding of process behaviour.

The methods presented here build upon our previous work introducing the PABLO framework (Buliga et al., 2024). The reason for choosing PABLO is that it is the only XPPM framework allowing for process-aware local explanations that combine both factual and counterfactual explanations. The main limitation of PABLO, as stated before, is its focus on only the control-flow perspective. In particular, the previous approach is extended by:

- Introducing a context-aware predictive process pattern discovery method
- Exploring novel encoding strategies for integrating different process perspectives into the final local explanation.
- Introducing a novel evaluation framework to assess the impact of integrating different process perspective in discovering effective patterns and delivering a comprehensive local explanation.

The effectiveness of the proposed methods is compared and tested against PABLO, as the baseline, by using an extensive evaluation protocol over different real-life datasets. The results suggest that leveraging multiple process perspectives allows to provide more accurate explanations, especially in terms of the accuracy of the surrogate model, with improvements ranging from 2% to 20% depending upon the event log under analysis.

The remainder of the paper is organised as follows: Section 2 discusses the limitation of existing approaches with a motivational

example. In Section 3, we outline the concepts required to understand the proposed method. Section 4 introduces our approach for providing process context-aware local explanations. Section 5 describes the experimental setup used to evaluate the proposed methods. Section 6 discusses the evaluation results of the context-aware process pattern discovery method, while Section 7 focuses on the evaluation of the final process context-aware local explanations. A comprehensive discussion of the obtained results, including the implications and limitations of the proposed method, is presented in Section 8. Section 9 reviews related work, and finally, Section 10 concludes the paper and suggest future research directions.

## 2. Motivational example

Let us consider the *SEPSIS* log,<sup>1</sup> a widely used public event log in PPM. This log details the medical progression of patients suffering from severe sepsis in a hospital in the Netherlands, from when they first enter the emergency room to when they are either discharged or returned to the emergency room.

Fig. 1 illustrates the simplified process model that represents the patients' pathway (Munoz-Gama et al., 2022). The model shows the high flexibility of the process, resulting in a wide range of process variants leading to either a desired outcome, *Discharge* (DI), or an undesired one, *Return to ER* (RE).

Each row of Table 1 corresponds to a trace, i.e., the recorded sequence of events of one specific sepsis patient within a hospital, where each event is composed of an activity with its respective timestamp. Let us consider the given trace with the ID of 1 in Table 1, hereafter referred to as the inquiry trace. The patient is first *registered* (RG) at midnight (stated as 0:00), followed by ER Triage after 1:30 h. Subsequent measurements include *CRP* (CR) at 3:00, while *ER Sepsis Triage* (ST) occurs at 4:30, and *Leucocytes* (LE) measurements are performed 30 min after that at 5:00. Treatment activities are then performed: *Intravenous Liquids* (IL) at 6:35 and *Intravenous Antibiotics* (IA) after 25 min of IL. A final measurement of *Lactic Acid* (LA) is performed at 8:00. Let us assume there is a black-box prediction model that can predict the outcome of the specific executed trace accurately. As noted in Table 1, this trace leads to an undesirable outcome from the black-box model: RE.

Given the undesirable predicted outcome for this patient, the process analyst seeks to understand the factors leading to this prediction and potential modifications that could shift the outcome to the desired one: DI. State-of-the-art methods in XPPM, such as LORELEY (Huang et al., 2021), CREATED (Hundogan et al., 2023), and the PABLO framework (Buliga et al., 2024), exhibit significant limitations in providing actionable and insightful explanations.

LORELEY focuses on case attributes, such as patient age, to generate explanations. While these attributes may influence outcomes, they are inherently static and unchangeable, offering limited value to practitioners seeking actionable insights. LORELEY entirely ignores the dynamics of the control flow, failing to capture how changes in the sequence or timing of activities might impact outcomes. CREATED generates synthetic traces that modify attributes and control flows to explore alternatives, resulting in traces T1 to T3 in Table 1. However, it does not provide automated or practical solutions. Analysts are burdened with manually comparing synthetic traces with the original to identify significant changes, making the approach time-consuming and less effective for deriving actionable alternatives to improve outcomes.

On the other hand, the PABLO framework seeks to distinguish between patterns in generated factual and counterfactual traces to identify patterns influencing positive or negative outcomes. Therefore, following the PABLO framework, we also generate traces T4–T6 in Table 1 as factual traces for our inquiry trace. However, PABLO primarily

<sup>1</sup> [https://data.4tu.nl/articles/dataset/Sepsis\\_Cases\\_-\\_Event\\_Log/12707639](https://data.4tu.nl/articles/dataset/Sepsis_Cases_-_Event_Log/12707639)

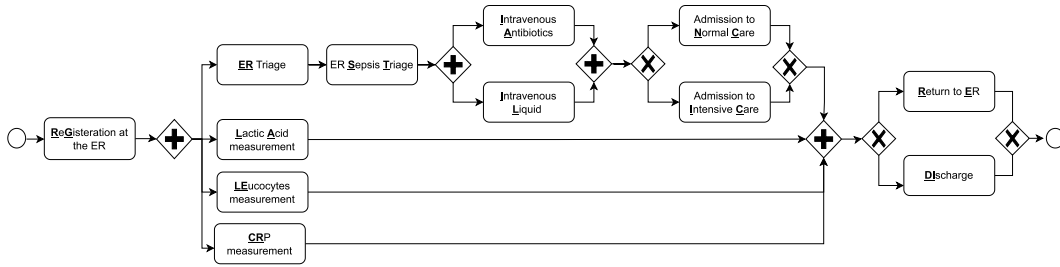


Fig. 1. Simplified process model for the sepsis event log (Munoz-Gama et al., 2022).

Table 1

A hypothetical example of an inquiry trace and the generated neighbourhood.

ID	Age	Inquiry trace execution																Predicted outcome
		Act <sub>1</sub>	Time <sub>1</sub>	Act <sub>2</sub>	Time <sub>2</sub>	Act <sub>3</sub>	Time <sub>3</sub>	Act <sub>4</sub>	Time <sub>4</sub>	Act <sub>5</sub>	Time <sub>5</sub>	Act <sub>6</sub>	Time <sub>6</sub>	Act <sub>7</sub>	Time <sub>7</sub>	Act <sub>8</sub>	Time <sub>8</sub>	
1	75	RG	0:00	ER	1:30	CR	3:00	ST	4:30	LE	5:00	IL	6:35	IA	7:00	LA	8:00	RE
ID	Age	Synthetic trace execution																Predicted outcome
		Act <sub>1</sub>	Time <sub>1</sub>	Act <sub>2</sub>	Time <sub>2</sub>	Act <sub>3</sub>	Time <sub>3</sub>	Act <sub>4</sub>	Time <sub>4</sub>	Act <sub>5</sub>	Time <sub>5</sub>	Act <sub>6</sub>	Time <sub>6</sub>	Act <sub>7</sub>	Time <sub>7</sub>	Act <sub>8</sub>	Time <sub>8</sub>	
T1	73	RG	1:00	CR	1:15	ER	1:30	LE	3:15	LA	4:45	ST	5:30	IL	6:30	IA	7:40	DI
T2	74	RG	8:10	CR	9:00	LA	10:15	LE	11:30	ER	12:00	ST	13:00	IL	14:00	IA	15:30	DI
T3	75	RG	2:50	LE	3:15	LA	4:00	CR	4:15	ER	4:30	ST	5:00	IA	6:30	IL	6:45	DI
T4	75	RG	1:00	CR	1:15	ST	1:30	LE	2:00	ER	5:00	IL	6:30	IA	7:00	LA	8:00	RE
T5	75	RG	7:15	LE	7:30	CR	9:45	ER	10:00	ST	11:00	IA	12:30	IL	12:30	LA	14:15	RE
T6	75	RG	6:10	CR	6:30	ER	6:30	ST	6:45	LE	7:00	IL	7:45	IA	8:00	LA	9:15	RE

focuses on control flow patterns and neglects temporal dependencies. This omission prevents it from identifying nuanced, data-dependent patterns (e.g., specific time gaps between events) that could influence outcomes. For example, the control flow pattern where the activity CR is eventually followed by LE is present in both desired and undesired predicted outcomes, as well as in the inquiry trace. Consequently, this pattern would not be recognised as a discriminative pattern. By examining the timestamps, however, we can uncover a hidden rule: if LE is executed more than 2 h after CR, it leads to the desired outcome. Conversely, if the time between these two events is short (less than 45 min), it results in an undesired outcome. This pattern suggests that for a better outcome, the CR should be conducted first, followed by the LE at least 2 h later. If the tests are conducted too close together, the reliability of the second test's results could be compromised, leading to an undesired outcome. Without incorporating a time perspective such as the time between events, patterns like this remain hidden.

In summary, while these methods address different aspects of process analysis, they fail to account for the multi-perspective nature of process data, such as the interplay between control flow dynamics, temporal dependencies, and case or event attributes. In this paper, an approach is proposed that addresses this challenge by integrating multiple process perspectives to deliver insightful and actionable explanations.

### 3. Background

In this section, we first recall the basic concepts of PPM. Then, we introduce the PABLO framework as the baseline of this work.

#### 3.1. Preliminary

**Definition 1 (Event).** Let  $\mathcal{AC}$  be the universe of activities,  $\mathcal{C}$  be the universe of case identifiers,  $\mathcal{T}$  be the time domain, and  $D_1, D_2, \dots, D_m$  be the sets of additional attributes with  $i \in [1, m]$ ,  $m \in \mathbb{N}$ . An event is a tuple of  $e = (a, c, t, d_1, \dots, d_m)$ , where  $a \in \mathcal{AC}$ ,  $c \in \mathcal{C}$ ,  $t \in \mathcal{T}$  and  $d_i \in D_i$ .

**Definition 2 (Trace, event log).** A trace  $\sigma = \langle e_1, \dots, e_n \rangle$  is a finite non-empty sequence of events  $e_1, \dots, e_n$  such that their timestamp does not decrease. Let  $S$  denote the universe of all possible traces, an event log can be defined as  $L = \{\sigma_1, \sigma_2, \dots, \sigma_n\} \subseteq S$  which is a set of traces.

We use  $E_\sigma$  for the set of events in trace  $\sigma$ . We define  $\pi_{act}(e)$ ,  $\pi_{time}(e)$ ,  $\pi_{case}(e)$ , and  $\pi_{D_i}(e)$  to return the activity, timestamp, case identifier and the attribute  $d_i$  associated with  $e$ , respectively.  $D_i$  represents either case attributes, constant across a trace  $\sigma$ , or event attributes, which can vary for each event  $e$ . Timestamps ( $t \in \mathcal{T}$ ) are used only to establish the order of events. Any data derived from these timestamps, such as time since the last event or event duration, is treated as a  $D_i$  and considered as event attributes.

**Definition 3 (Process Outcome).** We define  $\mathcal{Y}$  to be the set of possible process outcomes. A labelling function is defined as  $y : S \rightarrow \mathcal{Y}$  that maps each  $\sigma \in L$  to its corresponding outcome  $y(\sigma) \in \mathcal{Y}$ .

A well-known issue of log traces is that they flatten the real ordering relations among process events, hiding possible concurrency (Lee-mans et al., 2023). To preserve concurrency information, we convert log traces in so-called *partially ordered traces*, assuming a conversion oracle function is obtained from expert knowledge or data analysis (Diamantini et al., 2016; Lu et al., 2015).

**Definition 4 (Partially Ordered Trace).** Given a conversion oracle function  $\varphi$  and a log trace  $\sigma$ , a *partially ordered trace*  $\varphi(\sigma) = (E_\sigma, <_\sigma)$  is a Directed Acyclic Graph (DAG), where  $E_\sigma$  and  $<_\sigma \in E_\sigma \times E_\sigma$  corresponds to the set of nodes and edges, respectively. We define matrix  $A_{\varphi(\sigma)}$  as an upper triangular adjacency matrix that specifies directed edges from  $e$  to  $e'$ , with  $e, e' \in E_\sigma$ . Also,  $R_{\varphi(\sigma)}$  is the reachability matrix derived from  $A_{\varphi(\sigma)}$  to represent all possible paths from  $e$  to  $e'$  of length  $l$  such that  $2 \leq l \leq |\sigma| - 1$ . For each pair of events  $e, e' \in E_\sigma$ , such that  $e \neq e'$ , we define the following ordering relations:

- $R_{\varphi(\sigma)}(e, e') \neq 0 \leftrightarrow e'$  eventually follows  $e$ ,
- $A_{\varphi(\sigma)}(e, e') \neq 0 \leftrightarrow e'$  directly follows  $e$ ,
- $R_{\varphi(\sigma)}(e, e') = 0$  and  $R_{\varphi(\sigma)}(e', e) = 0 \leftrightarrow e$  is concurrent with  $e'$ .

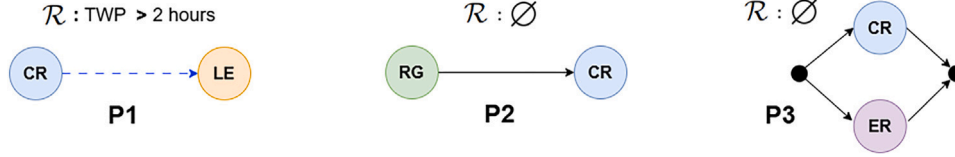


Fig. 2. Process pattern examples from the Table 1.

**Definition 5 (Discriminative Rule).** We define a discriminative rule  $\mathcal{R}$  as a conjunction of conditions over sets of attributes  $D_i$  characterising a certain outcome  $c_j \in \mathcal{Y}$  as below:

$$\mathcal{R} : \bigwedge_{i \in I} (D_i \diamond_i \omega_i) \Rightarrow c_j$$

- $I$  is the set of indices for features involved in rule  $\mathcal{R}$
- $\diamond_i$  represent the relation operator (e.g.,  $\leq, \geq, =, \neq$ )
- $\omega_i$  is the value or threshold that feature  $D_i$  is compared against
- $c_j \in \mathcal{Y}$  is the associated outcome for rule  $\mathcal{R}$

**Definition 6 (Process Pattern).** A process pattern  $P = (N, \mapsto, \alpha, \mathcal{R}, \cdot)$  is a DAG, where:

- $N$  is a set of nodes,
- $\mapsto$  is a set of edges over  $N$ ,
- $\alpha$  is a function that assigns an activity ( $a \in \mathcal{AC}$ ) to any node  $n \in N$ ,
- $\mathcal{R}$  is a discriminative rule assigned to pattern  $P$ ,

Fig. 2 shows a few examples of the process patterns introduced in Section 2.  $P1$  represents an eventual relation,  $P2$  shows a direct relation and  $P3$  presents a concurrent relation<sup>2</sup> between nodes, which, in this context, means that the two events are executed at the same time (i.e., within a given time window).  $P1$  includes a discriminative rule indicating that the Time Within Pattern (TWP) should be above 2 h.

Given a process pattern, an *instance* of the pattern is an occurrence of the pattern in a trace.

**Definition 7 (Pattern Instance).** Let  $P = (N, \mapsto, \alpha, \mathcal{R})$  be a pattern,  $\varphi(\sigma) = (E_\sigma, <_\sigma)$  a partially ordered trace,  $A_{\mapsto}$  be an upper triangular adjacency matrix over  $N$ , and  $R_{\mapsto}$  be the reachability matrix of size  $|N| - 1$  derived from  $A_{\mapsto}$ . Given a subset  $E' \subseteq E_\sigma$  of nodes in  $\varphi(\sigma)$ , such that there is a bijective function  $I$ . Then we define  $E'$  to be a *pattern instance* of  $P$  in  $\varphi(\sigma)$ , that is  $E' \in PI(P, \varphi(\sigma))$ , if  $\forall e, e' \in E'$  all the following properties hold:

- $A_{\varphi(\sigma)}(e, e') = A_{\mapsto}(I(e), I(e'))$
- $R_{\varphi(\sigma)}(e, e') = R_{\mapsto}(I(e), I(e'))$
- $\pi_{act}(e) = \alpha(I(e))$
- $(\pi_{D_i}(e) \diamond_i \omega_i) \in \mathcal{R}, i = 1, \dots, m$

**Definition 8 (Sequence/trace Encoder).** A sequence (or trace) encoder  $e : S \rightarrow F_1 \times \dots \times F_m$  is a function that takes a trace  $\sigma$  and transforms it into a feature vector  $x = e(\sigma)$  in the  $p$ -dimensional vector space  $F_1 \times \dots \times F_m$ , with  $F_j, 1 \leq m \leq p$  being the domain of the  $j$ th feature.

**Definition 9 (Outcome Predictive Model).** An outcome predictive model is a function  $M : F_1 \times \dots \times F_m \rightarrow \mathcal{Y}$ , where  $F_1 \times \dots \times F_m$  represents a  $m$ -dimensional feature space as transformed by the sequence encoder from a trace  $\sigma$ . Here, we define  $M_b$  as a *black-box* prediction model and  $M_w$  as a *white-box* or interpretable prediction model.

<sup>2</sup> Note that we added the black dots only for the sake of clarity in the visualisation of a concurrent pattern, and they do not belong to  $P3$ .

### 3.2. Baseline local XPPM framework

Fig. 3 provides an overview of the PABLO framework (Buliga et al., 2024), which serves as the baseline for this work. The framework allows an analyst to examine the behaviour of a black box model for a specific inquiry trace. A common approach from the literature is to generate alternative perturbations, forming a synthetic neighbourhood around the inquiry trace (Buliga et al., 2024; Huang et al., 2021; Guidotti et al., 2019). Then, given this synthetic neighbourhood, a standard practice to mimic the behaviour of the black-box model is to train an interpretable surrogate model. The intuition of this approach is that, given a non-linear black-box model, the local behaviour of its decision boundary can be approximated using a linear surrogate model (Guidotti et al., 2019). The PABLO framework is inherently model-agnostic, thus allowing users to leverage any sort of black-box model as it only requires the output predictions from the model. PABLO framework allows the generation of the process-aware explanation by discovering and encoding process patterns for the final surrogate model. The PABLO framework consists of four main components, as detailed below.

- Step 1: The *predictive model training* module receives an event log  $L$  as input to train a predictive black-box model  $M_b$ . Through this step, a trace encoder  $e$  is used to convert  $\sigma \in L$  in a compatible format for the chosen predictive model, and the model will be trained.
- Step 2: Given a predictive model  $M_b$  and an encoded inquiry trace  $x = e(\sigma)$  from the previous step, the *Synthetic Neighbourhood Generation* component generates synthetic factual and counterfactual traces called  $x^{syn}$ . If we have  $M_b(x) = y$ , then a factual trace  $x^{syn}$  is defined in way that  $M_b(x^{syn}) = y$  and a counterfactual trace is defined as  $M_b(x^{syn}) = \bar{y}$ , where  $\bar{y} \neq y$ . A set of *Quality Measures* and *Feasibility Constraints* must be defined based on user needs, such as prioritising traces close to the inquiry trace while minimising changes. A set of complete measures is introduced in (Buliga et al., 2024). The output of this step is defined as  $\mathcal{L}_{syn}$  encompassing generated synthetic factual and counterfactual traces.
- Step 3: The *pattern discovery* module receives  $\mathcal{L}_{syn}$  and returns a set of discovered process patterns  $\mathcal{P}$ , where  $\forall P \in \mathcal{P} \exists \sigma^{syn} \in \mathcal{L}_{syn}, PI(P, \sigma^{syn}) \neq \emptyset$ . In other words, for each discovered process pattern, there is at least one trace that contains that process pattern.
- Step 4: The final step aims to train an interpretable prediction model (a *surrogate* model  $M_w$ ) on the event log  $\mathcal{L}_{syn}$  encoded in terms of the frequency of the discovered patterns from the previous step.

## 4. Process context-aware local explanation

This section introduces the proposed approach aimed at incorporating multiple process perspectives in a unified factual and counterfactual local explanation framework. We utilise the PABLO framework as the foundation of our research, extend and adapt it to *discover* process patterns accounting for multiple perspectives of the process context and *encode* the process context meaningfully for the surrogate model. Accordingly, Section 4.1 introduces a novel method for discovering

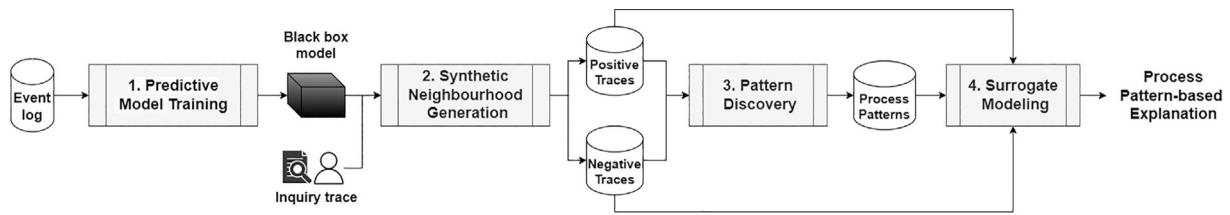


Fig. 3. PABLO method (Buliga et al., 2024).

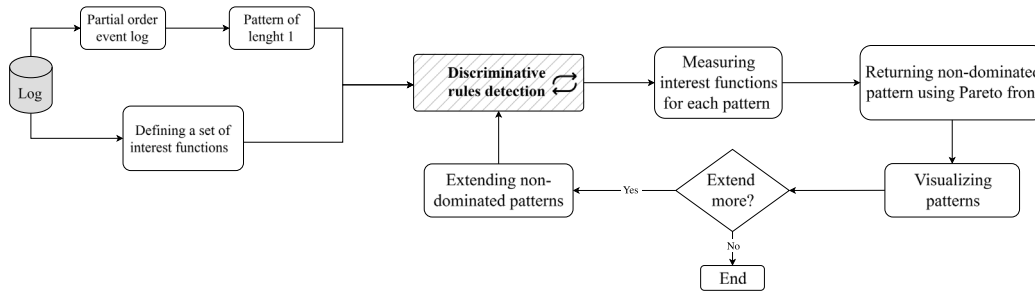


Fig. 4. Context-Aware Interactive Multi-Interest Process Pattern Discovery framework.

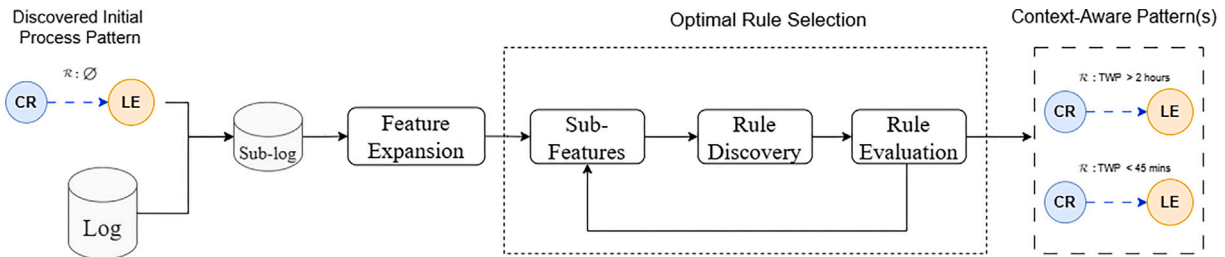


Fig. 5. Discriminative rule detection procedure.

context-aware predictive process patterns. In Section 4.2, we present different encoding scenarios to encode the discovered context-aware patterns in a compatible format for the surrogate model.

#### 4.1. Context-aware predictive process pattern discovery

Given a neighbourhood of factual and counterfactual traces, the goal of context-aware process pattern discovery is to identify a set of interesting control-flow patterns that incorporate multiple process perspectives, thus generating a set of context-aware process patterns. The interest of a pattern is often associated with various dimensions, such as its predictive power or its correlation with a specific outcome (Fang et al., 2020). To this end, we follow the Interactive Multi-Interested Process Pattern Discovery (IMPressed) (Vazifehdoostirani et al., 2023b) framework as the baseline discovery framework. IMPressed applies a *multi-optimisation* approach to determine the patterns that score best according to multiple user-defined interest functions.

In this section, we introduce an extension of this framework called Context-Aware IMPressed (CA-IMPressed), shown in Fig. 4.

As with the original IMPressed framework (Vazifehdoostirani et al., 2023b), the CA-IMPressed begins with an event log as input. The event log is then converted into a partial-order event log using a conversion oracle based either on expert input or data analysis. First, initial patterns of length 1 (representing single-activity patterns) are identified. Only non-dominated patterns based on a Pareto front, formed from interest functions defined by users, are retained and extended in each iteration. This extension incorporates additional events using relations like directly follows, eventually follows, directly precedes, eventually preceding, and concurrent (Vazifehdoostirani et al., 2023b).

The key additional step in this new variant method lies in the *discriminative rule detection* step (indicated as hatched in Fig. 4), which enables us to uncover patterns accounting for multiple process perspectives. Fig. 5 details the steps inside the discriminative rule detection procedure. The input of this procedure is all extended patterns. Given a pattern  $P$ , the pattern is initially annotated with an empty set of discriminative rules ( $\mathcal{R} = \emptyset$ ).

The goal of this step is to update  $\mathcal{R}$  with the most predictive rules associated with  $P$ . To achieve this, we adopt a strategy inspired by the minimax principle, commonly used in machine learning to optimise performance under worst-case scenarios (Cheng et al., 2014). In our context, we first focus on a sub-log  $L_p \subset L$  comprising  $\sigma \in L$  that contains at least one instance of the pattern  $P$ , i.e.,  $\forall \sigma \in L_p, PI(P, \sigma) \neq \emptyset$  (see Definition 7). Then, we create a dataframe, where each row represents one pattern instance. Through a comprehensive *feature expansion*, we maximise the data's representation dimension by including relevant features and then apply a minimisation strategy through *optimal rule selection* step to select a subset of features leading to the most predictive rules, if any, for each pattern.

As shown in Fig. 5, the initial pattern resulted in two distinct patterns, each corresponding to a specific process outcome. While both context-aware patterns share an identical control-flow structure, they are differentiated by unique rules based on the value of the TWP. The TWP feature is defined during the *feature expansion* step and identified as the most predictive feature during the *optimal rule selection* step. In the following subsections, we discuss these two main steps.

##### 4.1.1. Feature expansion

The goal of this step is to enrich the data representation by including multiple process perspectives at different granularity levels as

interpretable features. The underlying idea is transforming raw data into interpretable representations. This approach follows the same principles of techniques applied in other domains, such as super-pixel segmentation in image analysis, which transforms raw pixel data into meaningful clusters representing objects or their parts (Sokol and Flach, 2024). In the context of process data, we reviewed the literature on feature engineering in PPM to collect commonly used features. In particular, we draw upon two established levels of feature granularity from the literature, *intra-case* and *inter-case* features, and propose a novel feature granularity level that captures process perspectives by studying the interplay among multiple activities within a process pattern.

Several studies in process mining investigated different methods to enrich event logs with new information dimensions (Liu et al., 2023). Most works have focused on defining features that can be extracted from a single trace, known as *intra-case features*. Different predictive intra-case features, each focusing on one process perspective, such as time (Di Francescomarino et al., 2017; Navarin et al., 2017; Dogan, 2023; Camargo et al., 2019a; Tax et al., 2017), resource (Neu et al., 2022; Camargo et al., 2019b), ordering (control-flow) (Ceravolo et al., 2024; Vazifehdoostirani et al., 2023a), and other data (Teinmaa et al., 2019) (including additional case and event attributes) have been introduced.

Recent research has highlighted the importance of considering *inter-case features*, which reflect how ongoing cases within a process may influence each other's outcomes due to, e.g., their shared pool of resources (Senderovich et al., 2019, 2017; Grinvald et al., 2021; Kim et al., 2022). In the context of our work, we argue that inter-case features can be defined for time, resource, ordering and additional data perspectives by aggregating information related to each perspective across multiple cases. Furthermore, since inter-case features are often introduced in the literature to assess the *occupation* of a system by, for example, counting the number of open cases running (Teinmaa et al., 2019), we add the occupation perspective to this feature level.

Finally, while previous work focused mainly on inter and intra-case features, we intend to consider pattern-level features as well. The necessity of this type of feature is shown in the motivational example (Section 2), where the discriminative rule could be detected by looking at the relative time passed between two events within the pattern, not only the timestamps of the single event. We call this level of granularity *inner pattern relation* features. This level leverages the same process perspectives of the intra- and inter-case levels, allowing us to analyse these perspectives at a finer level of granularity. For example, the time perspective captures the time elapsed within a pattern. The resource perspective enables analyses such as work handovers within a process pattern, identified as a critical factor for process performance (Panpanich et al., 2015). The ordering perspective captures the positional distance between a pattern's activities. This feature is particularly valuable for patterns with eventual relations, as it indicates how far apart the activities in the eventual relation are. The occupation perspectives from inter-case features can also be meaningful here by capturing, for example, the average system occupation through the excitation of a process pattern. Additionally, data perspectives reveal dynamic changes in data within a process pattern. However, this perspective applies only to event data, as case data remain constant throughout the entire execution, making the definition of inner relations for them trivial.

In summary, three levels of feature granularity are defined and presented in Table 2. Each granularity level includes multiple process perspectives. For clarity, the table provides one example for each feature type. Depending on the process domain and given expert knowledge, additional data (such as credit score) fall under *data* perspectives. The relation between these data perspectives and control-flow patterns will be established through the optimal rule selection step (Section 4.1.2), depending on their effect on process outcome.

To provide a concrete example, we present an expanded set of features for instances of the pattern introduced in Section 2, focusing on time-perspective features. In this example, the initial control flow

pattern  $P$  is CR *eventually followed by* LE. For each instance of the pattern within, we create one data row with the ID defined by  $PI(\sigma, P)^n$ , where  $n$  refer to  $n$ th pattern instance of  $P$  within trace  $\sigma$ . For the intra-case level, we collected the time corresponding to each event ( $Time_{CR}$  and  $Time_{LE}$ ); the inter-case level is represented by  $Time_{CR}^{\rightarrow}$  and  $Time_{LE}^{\rightarrow}$ , indicating the time since the completion of the latest case in the system. Lastly, the inner pattern relation level is presented by the time passed between two events within the pattern ( $|Time_{CR} - Time_{LE}|$ ) (see Table 3).

#### 4.1.2. Optimal rule selection

In this step, we focus on selecting the most influential features across the aforementioned granularity levels and formulating discriminative rules that express the relationship between these selected features and the process outcome for each control-flow pattern. Specifically, for each control-flow pattern, first, we select the most relevant features among those introduced in the previous step. Then, we train a Decision Tree (DT) on the sub-log involving the pattern using the outcome as the target variable to derive the actual rules. The DT is chosen due to its ability to generate interpretable rules directly from the relevant features.

The feature selection step serves two purposes. First, it allows us to focus only on the feature with the highest impact on the outcome, thus reducing the data dimension and, hence, enhancing the rule interpretability. Second, it helps us prevent overfitting. To fulfil these requirements, we employ the Recursive Feature Elimination with Cross-validation (RFE-CV) technique (Freytes et al., 2023) as the instantiation of this module. RFE-CV employs an initial DT classifier trained on the sub-log  $L_p$  and aims to find the optimal set of features while maximising the classifier performance. This method iteratively eliminates the least significant features by evaluating their impact on the classifier's performance through cross-validation within  $L_p$ . This rigorous approach ensures that only the most impactful features are retained, preserving interpretability while enhancing efficiency and reducing the risk of overfitting. From the features selected by RFE-CV, discriminative rules are generated using the refined DT model and are encapsulated in  $\mathcal{R}$ . Note that if multiple rules hold for the same control-flow pattern  $P$ , duplicates of  $P$  are created, each corresponding to a unique rule.

#### 4.2. Process-context aware surrogate model

In this section, we introduce several strategies to train a surrogate model with full process context to achieve a process context-aware explanation at the end. We divide our strategies into two main groups where: the first group relies on the process patterns obtained from CA-IMPRESSED, and the second group acts as our baseline and relies on IMPRESSED.

For the first group, we propose two strategies based on the type of process data (event or case attributes) that were involved through the pattern discovery stage, particularly for the discriminative rule detection step. The first strategy, Full Context Patterns (FCP), integrates all available process contexts, both event and case, directly into the discovered patterns using CA-IMPRESSED. This ensures a comprehensive representation of the process context in patterns. Having FCP patterns, the surrogate model will be trained only on the frequency-based encoding of these patterns, as done in (Vazifehdoostirani et al., 2023b). The second strategy, the Dynamic Context Patterns (DCP) strategy, focuses on event attributes when deriving the patterns. In this strategy, the case attributes are excluded during discriminative rules detection but added later to the encoded feature vector for the surrogate model training. This separation is based on the distinction between process contexts in the final explanation. Case attributes may influence the outcome independently of any events or patterns (e.g., a patient's age affecting the outcome). In such cases, the DCP strategy ensures that these explanations are presented independently of any

**Table 2**  
Feature granularity levels associated with process patterns with the predictive potential.

Granularity levels	Perspectives	Example
Intra case features	Time	The event starting time
	Resource	Resource ID
	Ordering	Activity position
	Data	Additional case or event data
Inter case features	Time	Time since the last case in the system was completed
	Resource	Resource experience
	Ordering	Shift in activity position compared to the previous execution
	Occupation	Number of open cases at the execution time
	Data	Additional case or event data
Inner pattern relation features	Time	Time spent within a pattern instance
	Resource	Number of handovers within the pattern instance
	Ordering	Distance in the position of events of the pattern instance
	Occupation	Average number of open cases through the pattern instance
	Data	Additional event data

**Table 3**  
Expanded features associated with the instances of a pattern from Table 1.

ID	Intra case features		Inter case features		Inner pattern relation	Outcome
	$Time_{CR}$	$Time_{LE}$	$Time_{CR}^-$	$Time_{LE}^-$	$ Time_{CR} - Time_{LE} $	
$PI(T1, P)^1$	1:15	3:15	NaN	NaN	2:00	DI
$PI(T2, P)^1$	9:00	11:30	1:00	2:15	2:30	DI
$PI(T4, P)^1$	1:15	2:00	NaN	NaN	0:45	RE
$PI(T6, P)^1$	6:30	7:00	NaN	0:15	0:30	RE

**Table 4**  
Different process context inclusion scheme for the surrogate model's feature vector.

Encoding	Patterns	Discriminative rules		Event attributes	Case attributes
		Case attributes	Event attributes		
FCP	CA-IMPresseD	✓	✓	✗	✗
DCP	CA-IMPresseD	✗	✓	✗	✓
CALL	IMPresseD	✗	✗	✓	✓
CFP	IMPresseD	✗	✗	✗	✗

pattern. Conversely, event attributes are specific to individual events; therefore, they are included in discriminative rule detection to provide a unified explanation of dynamic process context. To this end, the same frequency-based encoding will be applied for the discovered pattern, while the case attributes will be concatenated to the resulting feature vector for each trace.

We introduce two baseline strategies as well, where the process patterns are obtained from IMPresseD, focusing only on the control-flow perspective. Concatenate All (CALL) relies on control-flow patterns discovered using the IMPresseD method. However, in this strategy, we concatenate all available case and event attributes to the frequency-based encoding vector for the surrogate model training. CALL builds on existing approaches from the literature and does not incorporate the contributions introduced in this paper. While this strategy includes intra-case and inter-case features during the encoding phase, it overlooks inner-pattern relations. The second baseline, called Control-Flow Patterns (CFP), leverages patterns discovered solely based on control flow using the IMPresseD method. No additional context is included during discovery, and no additional context will be added later through encoding. This strategy serves as a comparison point to evaluate the benefits of incorporating multiple process contexts into the explanation. In CFP, we only use the frequency-based encoding of the discovered patterns.

The different process context inclusion schemes are summarised in Table 4. In the table, the *Event Attributes* and *Case Attributes* refer to the fact of whether these attributes are concatenated in the feature vector used to train the surrogate model independently from the frequency-based encoding of the process patterns. The column *Discriminative rule* specifies the type of process contexts used to discover these rules within CA-IMPresseD. The *Patterns* column identifies the method employed to discover the process patterns. It is important to note that all types of patterns use frequency-based encoding for the feature representation.

**Table 5**  
Encoding output for each of the proposed method using T1 from Table 1.

Encoding	Feature vector for trace T1
FCP	$\{P_1 = 1, P_2 = 0, P_3 = 1\}$
DCP	$\{Age = 75, P_1 = 1, P_2 = 0, P_3 = 1\}$
CALL	$\{Age = 75, P_1 = 1, P_2 = 0, Time_1 = 1:00, Time_2 = 1:15, \dots, Time_8 = 7:40\}$
CFP	$\{P_1 = 1, P_2 = 0\}$

To showcase how the different encodings can be instantiated, we refer to the motivating example in Section 2. Looking at Table 1, let us assume the following patterns are discovered:

- $P_1$ : The control-flow pattern *Lactic Acid measurement* (LA) eventually followed by *Intravenous Antibiotics* (IA).
- $P_2$ : Another control-flow pattern, IA eventually followed by LA.
- $P_3$ : For the context-aware pattern, the *Leucocytes measurement* (LE) is executed more than 2 h after the *CRP measurement* (CR) is done.

Note that while all the mentioned patterns potentially can be discovered using CA-IMPresseD, IMPresseD is limited to discovering only  $P_1$  and  $P_2$ . Assume we aim to encode the trace T1 from Table 1 by leveraging each of these patterns, with the output of each strategy shown in Table 5.

## 5. Experimental setup

To evaluate the proposed approach, we designed two main evaluation setups, each dealing with one of the following research questions:

- RQ1 What are the differences in predictive capabilities between patterns discovered by the baseline method (IMPresseD) and

**Table 6**  
Event logs statistics.

Dataset	log	trace#	variant#	event class#	trace att.#	event att.#	avg. trace length	positive class %	prefix lengths	domain
BPIC2012_1 BPIC2012_2 BPIC2012_3	BPIC2012	4685	3790	36	1	10	35	47% 17% 35%	[15, 20, 25, 30]	finance
BPIC2017_1 BPIC2017_2 BPIC2017_3	BPIC2017	31,413	2087	36	3	20	35	41% 12% 47%	[20, 25, 30, 35]	finance
Sepsis_1 Sepsis_2 Sepsis_3	SEPSIS	782	709	15	24	13	14	14% 14% 14%	[7, 9, 13, 16]	healthcare
BPIC2011_f1 BPIC2011_f2 BPIC2011_f3 BPIC2011_f4	BPIC2011	1140 1140 1121 1140	815 977 793 977	193 251 190 231	6 6 6 6	14 14 14 14	25 54 21 44	40% 78% 23% 28%	[10, 15, 20, 25] [10, 15, 20, 25] [10, 15, 20, 25] [10, 15, 20, 25]	healthcare
Production	PRODUCTION	220	203	26	3	15	9	53%	[7, 11, 15, 19]	industrial

those identified by the proposed context-aware method (CA-IMPRESSED)?

RQ2 What is the effect of different process context encoding methods for the surrogate model on the fidelity and interpretability of the explanation?

To address the research question RQ1, we designed two labelling scenarios for real-life logs. The first scenario serves as a *controlled evaluation* to showcase the approach’s capabilities to discover patterns that go undetected when the context is not considered. This scenario employs a labelling function defining the outcome of a trace based on the existence of certain context-aware patterns, acting as the ground truth. The second labelling scenario leverages outcome labelling functions previously defined in the literature (Teinemaa et al., 2019). The focus of this second scenario is a *real-life evaluation* of the predictive capabilities of the proposed approach.

For RQ2, we have designed a *quantitative evaluation* followed by a *qualitative comparison*. The former measures the faithfulness of the explanations generated w.r.t the original black-box model. For the latter, we inspect the final explanation resulting from each encoding method introduced for the surrogate model in Section 4.2. Inspired by ablation studies, the aim of this analysis is to determine how the inclusion of the different types of process context, either within the context-aware patterns (FCP, DCP) or the feature vector (CALL), impacts the fidelity of the surrogate model.

### 5.1. Datasets

We analysed five event logs commonly used in literature, namely BPIC2017, BPIC2012, BPIC2011, SEPSIS and PRODUCTION. Table 6 reports some characteristics of the chosen datasets, as well as the range of prefix lengths used for the evaluation. These event logs were chosen due to the wide range of different characteristics, such as the number of trace attributes, varying trace lengths, log size, and different label distributions. The percentage of the positive class in the Table 6 is reported based on the process outcome labelling strategy used in (Teinemaa et al., 2019).

### 5.2. Experimental settings

We designed two experimental settings, each tailored to its respective question. The implementation of the methods introduced in this paper and the code required to reproduce experimental results can be found here.<sup>3</sup>

#### 5.2.1. Pattern discovery setting (RQ1)

We divide each log into 80%–20% for training and testing, respectively, using a temporal order split. During pattern discovery on the logs in Table 6, we encountered a runtime issue with both IMPRESSED and CA-IMPRESSED for BPIC2017. To address this, we applied a sampling method from the literature that preserves the predictive performance of models trained on sampled logs compared to the original log (Fani Sani et al., 2023). Using this approach, we reduced BPIC2017 to 1,153 traces and 400 variants, enabling pattern discovery within a feasible runtime. We use the same interest functions introduced in (Vazifehdoostirani et al., 2023b), including frequency, outcome and case distance interest functions for both IMPRESSED and CA-IMPRESSED. Here, we focus on comparing how the proposed method performs against the baseline predictive pattern discovery method that only takes the control flow into account.

For the feature expansion step (Section 4.1), we use the most common intra and inter-case features employed in previous PPM work (Teinemaa et al., 2019). For the inner pattern relation features, we leveraged common intra and inter-case features from the literature, and we defined straightforward aggregators such as average and sum, generating meaningful features in the context of business process management, such as average throughput time and handover. Table 7 presents the chosen features in these experiments. These features can be extracted from all real-life logs, while some other possible features require either more information to be recorded in the log, such as case priority, or expert validation, such as resource experience.

To discover the patterns, we followed the automatic discovery setup based on (Vazifehdoostirani et al., 2023b), where each non-dominated pattern in the Pareto form will be extended in the next iteration. We performed two iterations, since this setting allowed to achieve satisfying classification performance in previous work.

The discovered patterns are encoded using frequency-based encoding. We then compare the performance of DT trained on encoded logs in these experiments. The two DT models are evaluated on a test set, that was not used either for pattern discovery or for training the DT, using the Area Under the Curve (AUC) metric, a commonly used metric that considers the probability distribution of each of the possible labels (Bradley, 1997).

#### 5.2.2. Local explanation setting (RQ2)

To obtain the context-aware local explanation, we instantiate the PABLO framework with the methods proposed in Section 4. We divide each dataset into 70%–10%–20% for training, validation, and testing, respectively, using a sequential order split. For the black-box model  $M_b$ , we leverage the XGBoost model. This type of model was selected as it was shown in previous studies to provide performance on par with Deep Learning architectures, however requiring less training data (Peep-erkorn et al., 2023). We nonetheless point out the fact that the PABLO is

<sup>3</sup> [https://github.com/abuliga/PABLO/tree/data\\_pablo](https://github.com/abuliga/PABLO/tree/data_pablo)

**Table 7**  
Defined features for the experiments based on the PPM literature.

Granularity levels	Perspectives	Examples
Intra case features	Time	Hour (time of day), weekday, month, time since the last event, time since midnight, duration, time from the start of the trace
	Resource	Resource ID, Resource group
	Ordering	Activity position
	Data	Existing case and event attributes in each real-life log
Inter case features	Time	–
	Resource	–
	Ordering	–
	Occupation	Number of open cases
Inner pattern relation features	Data	–
	Time	Average of all time-related intra-case features, the sum of activity durations, binary feature indicating if all events of a pattern performed in same weekday and month, sum of time spend within events inside the pattern
	Resource	Binary feature indicating the occurrence of handover within the pattern, frequent resource within the pattern, set of involved resources in pattern
	Ordering	Distance in the position of events of a pattern instance
	Occupation	Average number of open cases through the pattern
Data	For other available event data, we used the average of numerical attributes, and a set of categorical attributes	

model agnostic and works with all types of predictive models. For each dataset, we used 4 different prefix lengths (see Table 6) and we used complex-index encoding to encode all static and dynamic information from the trace (Leontjeva et al., 2015) as a compatible choice for XGBoost. The optimal XGBoost model setup is selected via hyperparameter optimisation for each dataset and prefix length. For optimisation, the Tree Parzen Estimator (TPE)<sup>4</sup> is used with the objective of maximising the AUC.

Each instance from the test set is used as inquiry trace  $x$  to undergo the explanation pipeline. We employed the Genetic Algorithm (GA) based on (Mothilal et al., 2020) for generating the synthetic neighbourhood  $\mathcal{L}_{\text{syn}}$ . GA was chosen for its ability to generate diverse traces while optimising a fitness function to ensure that the generated traces meet the required specifications. To achieve this, we employed the same quality metrics and feasibility measures to train the GA as in (Buliga et al., 2024). The only differences are that (1) the inquiry trace in this work contains all event and case attributes used in training the black box, allowing us to generate different process perspectives in the synthetic neighbourhood as well, (2) unlike (Buliga et al., 2024), we included the feasibility constraint inside the fitness function of the GA with all other quality measures to account for the feasibility of the generated traces through the GA iterations.

The log  $\mathcal{L}_{\text{syn}}$  is composed of 200 traces generated for each inquiry trace, with 100 traces being generated for each outcome, resulting in a balanced event log. A maximum of 100 iterations was set for the evolutionary algorithm, with a population size of  $20 * 100 = 2000$ .

In addition to the interest functions defined in (Buliga et al., 2024), here we define an additional interest function to mitigate the impact of confounding variables introduced in (Vazifehdoostirani et al., 2023b). This interest function called *Case Distance*, aims at prioritising patterns with the smallest average case distance between the case attributes of traces that conform to the pattern and those that do not.

Regarding the training of the surrogate model  $M_w$ , we instantiate a DT, for which we also perform the hyperparameter optimisation. We chose a DT as it is considered a white-box interpretable model that is able to provide all the rules leading to different predictions from which we are able to extract the local rule-based explanations. The different encoding strategies introduced in Section 4.2 are implemented and compared. In terms of baselines for comparison, we consider CFP and

CALL that only leverage the control-flow patterns discovered using the standard components of the PABLO framework (i.e., IMPresseD). We omit other state-of-the-art methods for local explanations as PABLO was already shown to outperform other methods in previous work (Buliga et al., 2024).

To measure the fidelity of the surrogate model  $M_w$ , we employ the Local Fidelity (LF) metric (Buliga et al., 2024), defined as:

$$LF = \frac{1}{|\mathcal{L}_{\text{syn}}^{\text{test}}|} \sum_{\sigma^{\text{syn}} \in \mathcal{L}_{\text{syn}}^{\text{test}}} \mathbb{I}(M_w(e(\sigma^{\text{syn}})) = M_b(e(\sigma^{\text{syn}}))) \quad (1)$$

where  $\mathbb{I}(\cdot)$  is the indicator function, returning 1 if the predictions of the derived surrogate model  $M_w$  and the black box model  $M_b$  on  $e(\sigma^{\text{syn}})$  match, and 0 otherwise.  $\mathcal{L}_{\text{syn}}^{\text{test}}$  represents a hold-out set from the generated synthetic neighbourhood used to evaluate the surrogate model, and  $e$  represents the feature encoder mapping traces to a feature vector compatible with predictive model. A higher LF indicates a strong alignment between the prediction obtained from the surrogate model and the black-box model, while an LF of zero reflects completely unfaithful predictions by the surrogate. A higher LF also highlights the accuracy of the generated rules as explanations as they are able to cover the predictions of the black-box model.

## 6. CA-IMPresseD evaluation

This evaluation aims to compare the predictive capabilities of the patterns discovered by our approach to the baseline to address RQ1. We first discuss the results of the *controlled evaluation* and then extend our experiments to the *real-life logs evaluation*.

### 6.1. Controlled evaluation

From the literature, three scenarios are usually recognised for process outcomes (Teinemaa et al., 2019). The first scenario involves outcomes determined by specific events or case attributes, such as the number of rejections in a production log. The second scenario involves outcomes defined solely based on control-flow patterns, for instance, compliance with given ordering constraints. The third scenario involves outcomes influenced by multiple process perspectives, such as the timing of a patient's return to the emergency room after discharge.

The controlled evaluation focuses on the third scenario, where specific context-aware process patterns are known to affect the outcome. The aim is to assess whether CA-IMPresseD can rediscover these patterns. We will evaluate the effectiveness of CA-IMPresseD by

<sup>4</sup> Hyperparameter optimisation was done by using the hyperopt python library

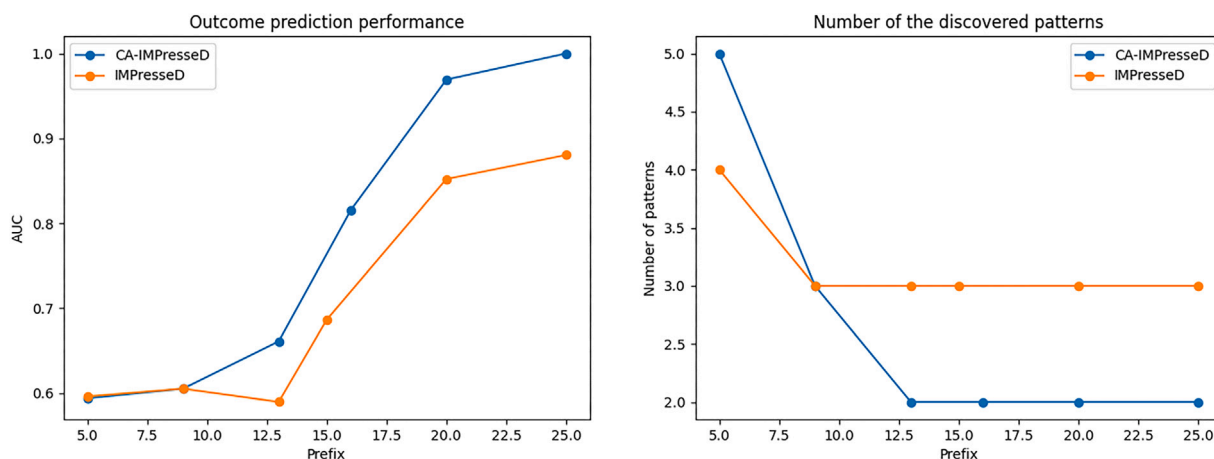


Fig. 6. Controlled evaluation results for non-processed sepsis log.

comparing it with IMPressed, as the baseline. We focus on the `SEPSIS` log, which contains such a context-aware pattern (Teinmaa et al., 2019), in Section 6.1.1. Then, to thoroughly assess the proposed model, we introduce three additional outcome labels for this log, each defined by a distinct pattern with unique characteristics, including direct, eventual, or concurrent control-flow structure in Section 6.1.2.

#### 6.1.1. non-processed sepsis log

For this experiment, we leverage the labelling defined in (Teinmaa et al., 2019), where the outcome of a trace is defined as positive if a patient returns to the emergency room within 28 days of discharge.

Note that we utilise a non-processed log containing complete traces where the complete pattern occurs, since our goal is to assess whether this pattern could be effectively discovered. Fig. 6 shows the AUC of the DT trained on encoded patterns and the number of discovered patterns for each method.

AUC does not differ for short prefixes because the target pattern appears toward the end of the process. As the trace length increases, CA-IMPRESSED outperforms the baseline, where the target pattern is discovered only by CA-IMPRESSED. All discovered patterns from prefix 25 (with highest achieved AUC) are shown in Fig. 7, arranged in order of importance based on the feature importance derived from the DT, from left to right. Each pattern's ID indicates the single activity from which the presented pattern was extended. Upon analysing these patterns we identified two patterns (each from one extension step) that led to a high AUC using CA-IMPRESSED.

The most important pattern based on the feature importance derived from DT is the single-activity pattern: Return ER with the discriminative rule of *'timesincelastevents' ≤ 40678.18 mins (28.2 days)*. The extension of Return ER in the second iteration of CA-IMPRESSED resulted in the pattern shown in Fig. 7. This pattern involves Return ER subsequent to the CRP rather than patient discharge. The uncovering of these patterns can be explained with the presence of multiple activity codes in the log, each referring to different types of discharges (Release A, Release B, Release C, and Release D). While in all cases (except for one outlier), the event preceding Return ER is associated with one type of patient discharge, the frequency of combining each discharge type with Return ER is lower than the frequency of combining CRP with Return ER, which occurs in 99% of cases. Since our pattern extension strategies, introduced in (Vazifehdoostirani et al., 2023b), focus on discovering direct, eventual, and concurrent behaviours without the capability to identify patterns with *⟨OR⟩* conditions, we cannot encompass all types of discharges in a single pattern. Nevertheless, the CA-IMPRESSED was able to identify two good proxy patterns, modelling the core of the expected behaviour. This insight underscores the need for more complex extension scenarios in future work to capture more complicated behaviours effectively.

On the other hand, IMPressed discovered three patterns, including a single-activity pattern of Return ER (most important pattern based on DT's feature importance) and two shown in Fig. 7. Since Leucocytes is also present in 99% of cases, "Pattern ID: Return ER\_6" is also discovered by IMPressed with the same reason mentioned above for CRP. Although the control flow aspect of the patterns discovered by CA-IMPRESSED can be found by IMPressed, neglecting the data perspective resulted in missing the actual discriminative behaviour, as showed by the lower prediction performance.

#### 6.1.2. Relabelled logs

Table 8 summarises the three labelling schemas for the `SEPSIS` log, where traces are labelled as positive if they contain at least one instance of the target pattern. Each target pattern features a direct, eventual, or concurrent control-flow structure paired with a specific discriminative rule. Fig. 8 presents the prediction performance and the number of discovered patterns using IMPressed and CA-IMPRESSED. The top four most important patterns for each log, determined by feature importance from the trained DT on the encoded patterns, are depicted in Appendix A.

The key observation from Fig. 8 is that CA-IMPRESSED significantly outperforms IMPressed in terms of AUC. This improvement stems from the high predictive power of context-aware process patterns compared to only the control-flow patterns. Across all datasets, CA-IMPRESSED successfully discovered the target pattern, which was consistently identified as the most important by DT for the outcome prediction. Interestingly, in the *time-based* relabelled log, where control-flow alone lacks any predictive power, IMPressed could not identify even the control-flow aspect of the target pattern, which justifies the substantial AUC difference.

It is worth mentioning that for being able to discover the target pattern defined in the *resource-based* and *time-based* relabelled logs, defining inner pattern relation features is crucial. Otherwise, relying solely on the resources or time of each event independently would not result in the target pattern. The *occupation-based* relabelled log reflects real-world scenarios where executing tasks simultaneously in a busy system may influence the outcome and is not necessarily dependent on inner pattern relation features. In this case, inter-case features play a pivotal role in detecting the discriminative rule of the target pattern.

Following the incremental pattern discovery and selection approach in (Vazifehdoostirani et al., 2023b), we selected the best patterns based on the Pareto front after each extension step. Therefore, while the target patterns all include two activities, approximations as single-activity patterns were discovered in the first iteration, and extensions of the target pattern emerged in the following pattern extension steps. This explains why, using both IMPressed and CA-IMPRESSED, we have discovered more than just one target pattern. CA-IMPRESSED often

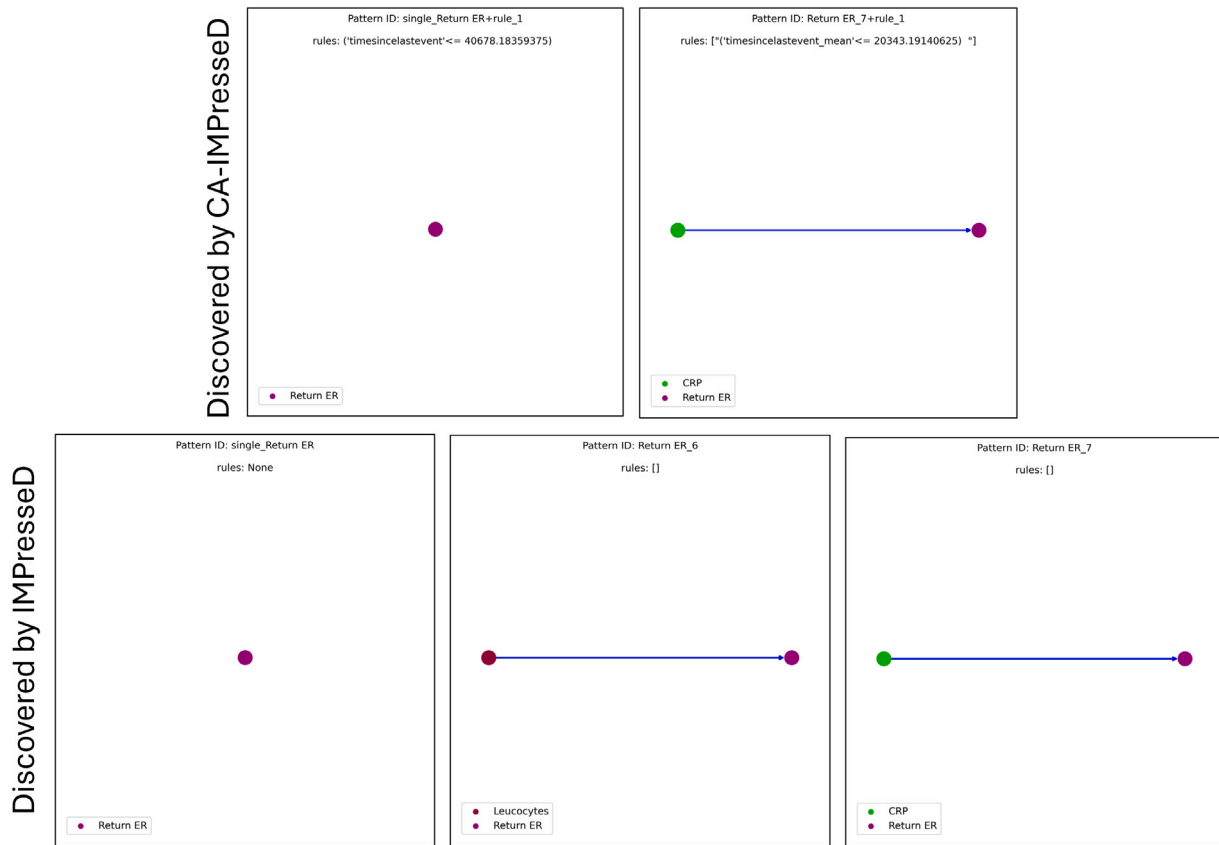


Fig. 7. All discovered patterns using both proposed and baseline methods from non-processed sepsis log (blue arrows: eventual relations, black arrows: direct relations).

**Table 8**  
Relabelled logs for controlled evaluation.

Relabelled logs	Target pattern	Structure
Resource-based	Leucocytes is directly followed by CRP, with a handover between these two events	Direct
Time-based	ER Triage is eventually followed by LacticAcid, and time between the two events $\leq 15$ min	Eventual
Occupation-based	IV Antibiotic is concurrent to IV Liquid, and the number of open cases $\geq 76$	Concurrent

discovers more patterns than IMPressed, as its discriminative rule detection step generates multiple context-aware patterns from a single control-flow pattern.

However, an unexpected observation is the significantly higher number of patterns than the target pattern discovered by CA-IMPressed in relabelled logs. The analysis of the discovered patterns revealed the following reasons:

- One of the interest functions used in (Vazifehdoostirani et al., 2023b) and current experiments may allow a pattern to dominate others despite not being highly predictive, but because of its frequency. For example, in *resource-based* relabelled log, while the single-activity pattern of CRP dominates due to its impact on the outcome, Admission NC dominates due to its frequency.
- Some hidden relations between activities and data payloads in the SEPSIS log were not completely under control. For example, in the *time-based* relabelled log, Leucocytes and LacticAcid occur consecutively in 57% of cases, always sharing the same timestamp. Consequently, the target pattern can also be interpreted as ER Triage eventually followed by Leucocytes within 15 min, a pattern identified by CA-IMPressed besides the main target pattern. Similarly, for the *occupation-based* relabelled log, the number of open cases remains constant across most events within the trace, which causes the target pattern to be interpreted by other frequent activities, such as ER Registration is eventually

followed by IV Antibiotic, where the average number of open cases  $\leq 76.7$  and *'timesinclaevent'*  $\leq 0.262$ .

Although CA-IMPressed discovered more patterns than the target pattern, it consistently begins by identifying the target pattern, which is always recognised by DT as the most important for prediction purposes.

### 6.2. Real-life logs evaluations

This section compares the predictive capabilities of the patterns discovered using both the IMPressed and CA-IMPressed frameworks on pre-processed real-life logs with the same outcome defined in the literature (Teinmaa et al., 2019). To this end, discovered patterns from different prefixes are encoded using frequency-based encoding and train a DT model on the encoded patterns to predict the process outcome. This approach is inspired by the evaluation method commonly used in *deviance mining* (Nguyen et al., 2014), which assesses the quality of the discovered patterns in predicting process outcomes without leveraging user knowledge. Here, the discovered patterns are treated as independent features, while the process outcome is the dependent variable. The results are presented in Fig. 9.

The AUC generally increases with prefix length (except for *Production* and *Sepsis\_1*), suggesting that patterns discovered later in traces have greater predictive power. This likely reflects the importance of detailed process contexts that emerge over time.

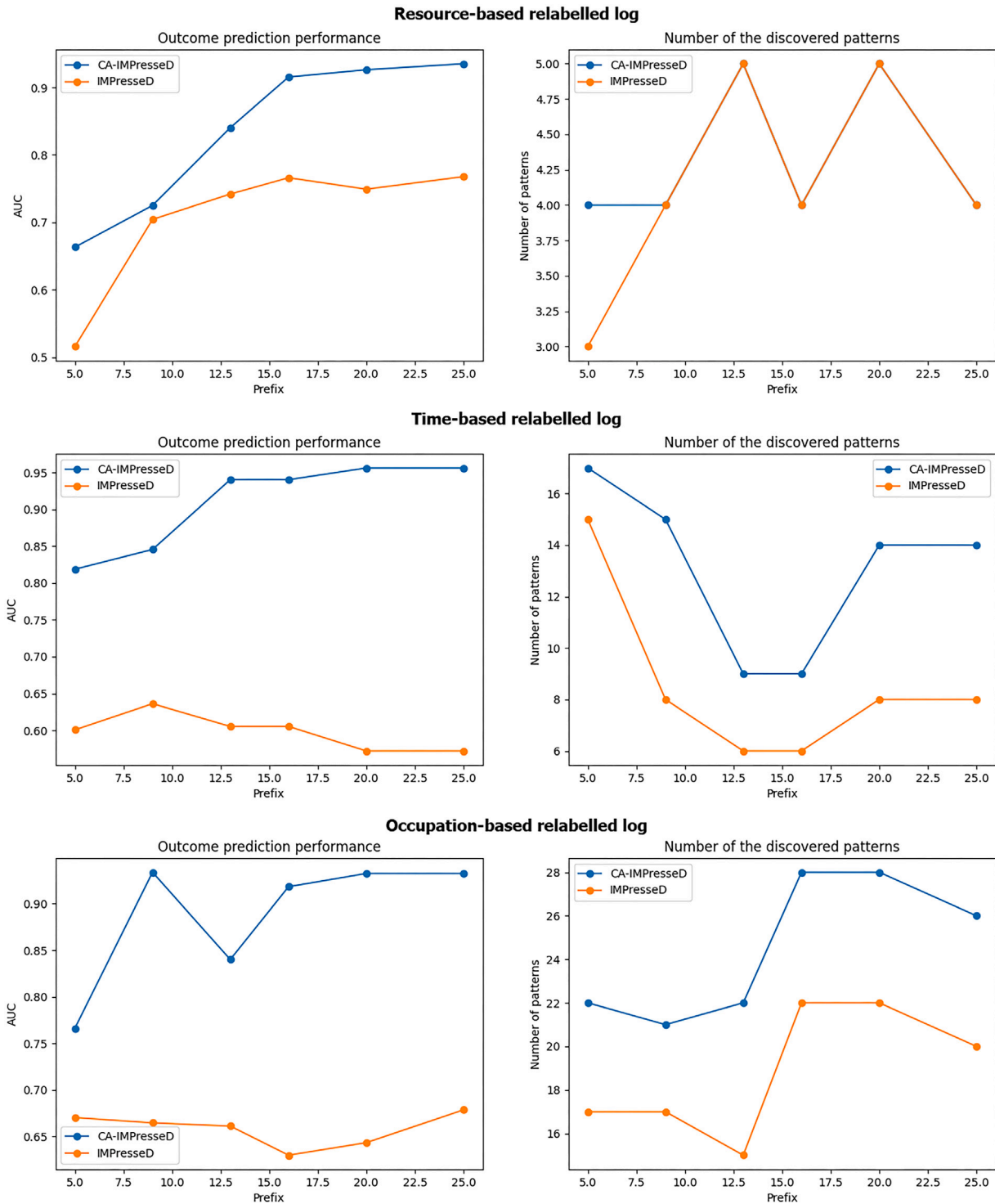


Fig. 8. AUC results and the number of discovered patterns from relabelled logs.

Prediction models trained on patterns from CA-IMPRESSED consistently outperformed those using IMPRESSED across various logs. Notably, the CA-IMPRESSED method performs significantly better in logs characterised by a higher number of event attributes. This is expected, as logs with richer attribute sets, such as BPIC2017 and BPIC2011, provide more opportunities for identifying context-aware patterns. In contrast, in logs like BPIC2012, where only one case attribute and only time-related event attributes are present, the opportunity to discover context-aware patterns is relatively limited.

Typically, the number of patterns identified by the CA-IMPRESSED method exceeds those discovered by IMPRESSED, especially in logs with more extensive data payloads. This trend is due to CA-IMPRESSED’s discriminative rule detection, which generates multiple context-aware patterns from each control-flow pattern. Interestingly, in the BPIC2011\_2 log, fewer context-aware patterns were discovered than control-flow patterns, yet these exhibited higher predictive capabilities.

Further investigation into the results for each event log provided additional insights. For instance, as indicated in (Teinmaa et al., 2019), a concept drift in the Sepsis\_1 log hinders high prediction accuracy when

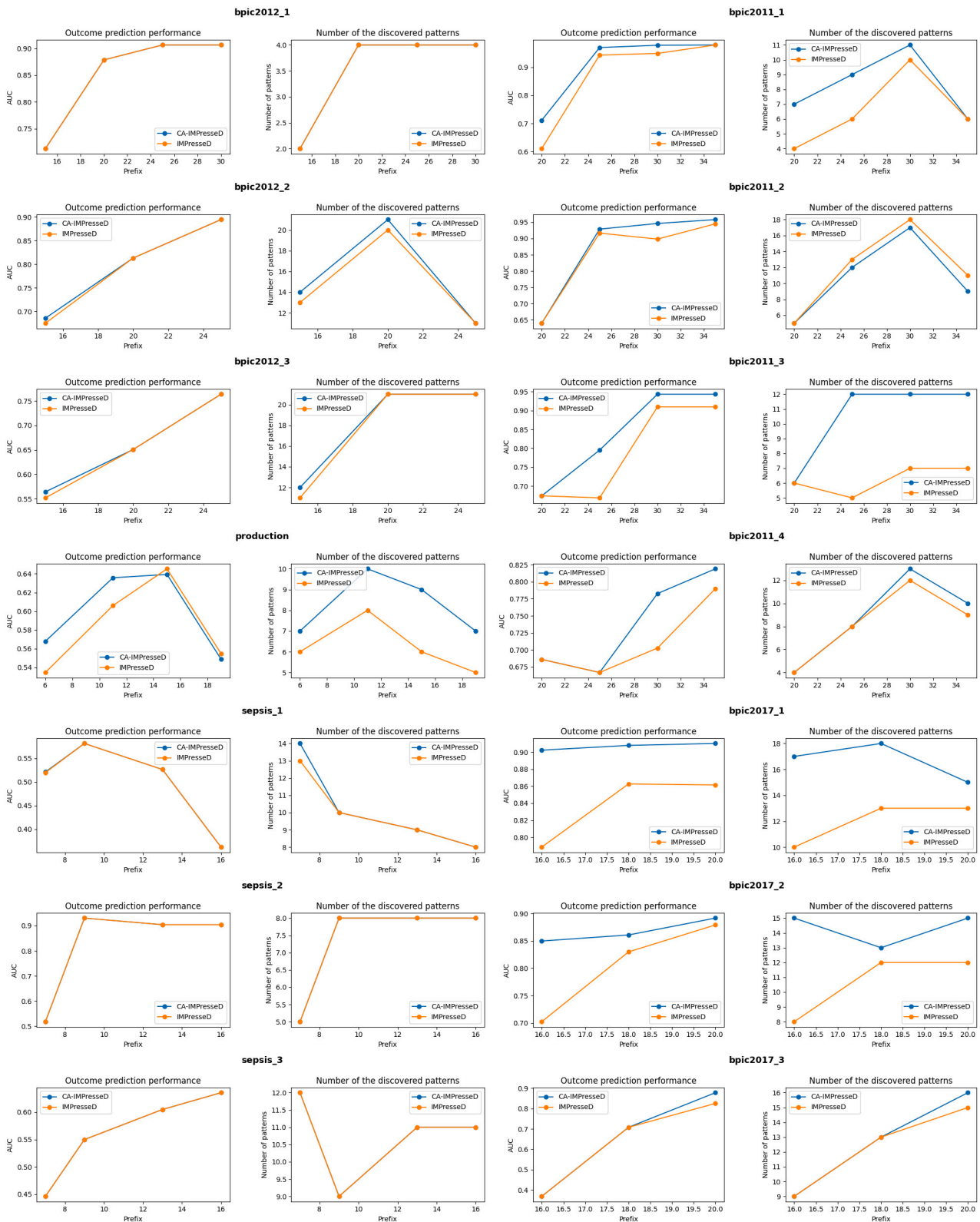


Fig. 9. Prediction performance comparison on original labelling of real-life logs.

using a temporal train–test split. This leads to very low AUC scores (close to 50%), which are sometimes worse than random guessing. Consequently, unlike other logs where increased prefix lengths correlate with improved prediction performance, the *Sepsis\_1* log shows a decline. Similarly, the issue with the *Production* log arises from differences in the distribution of prefix lengths between the training and testing

sets. Specifically, after prefix 13, only one case exists for each of the subsequent prefixes. This introduces significant variability to the test set. Furthermore, of these unique cases, only one displayed the key pattern identified for prefixes 15 and 19, which is the *presence of activity Packing when the 'PartDesc' ≠ 'Clamp'*. The limited number of

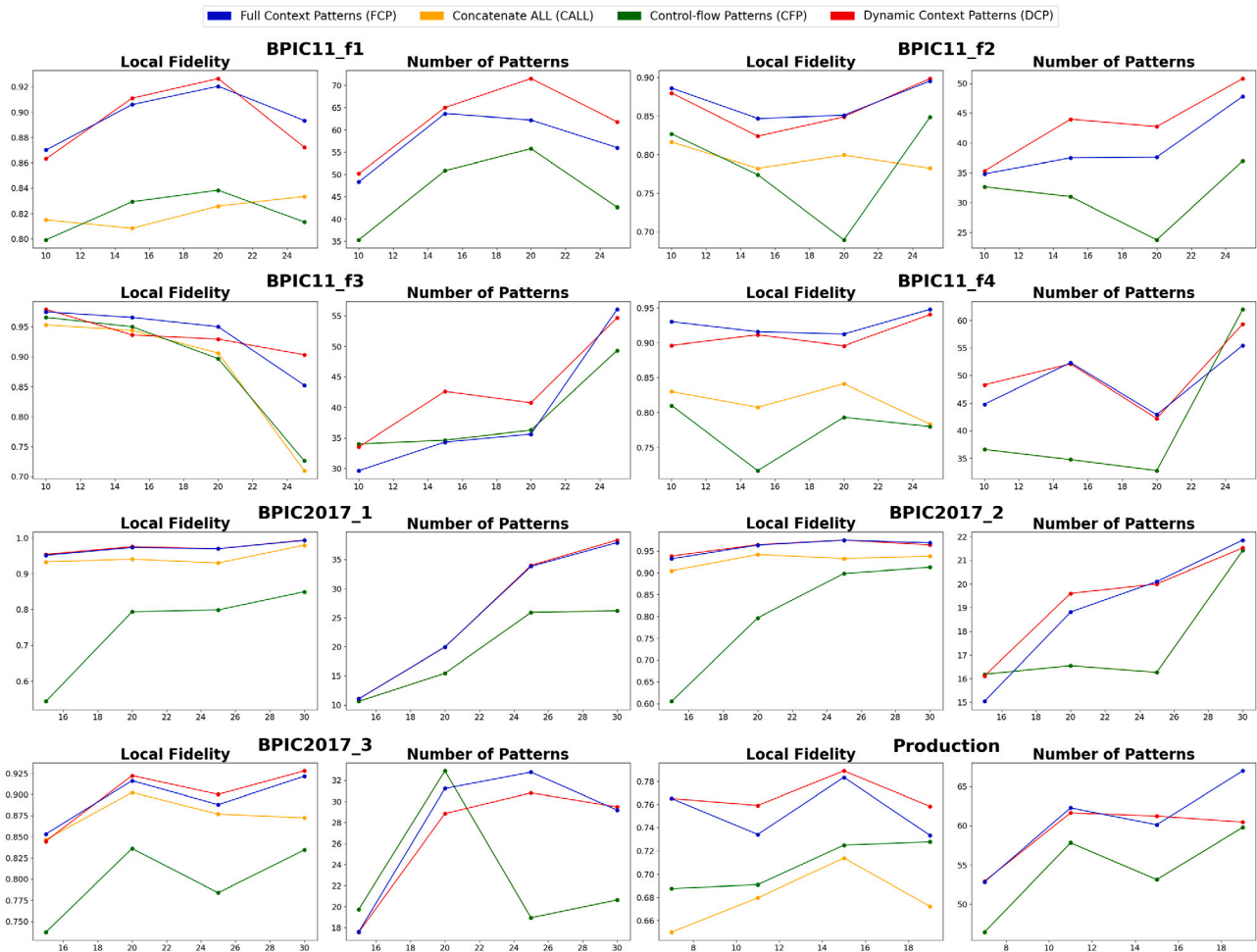


Fig. 10. Local fidelity comparison for different encoding scenarios.

cases at these critical prefixes makes the evaluation overly sensitive to variations in individual cases, leading to a decrease in AUC.

The most significant difference in prediction performance between the two sets of discovered patterns was observed in *BPIC2017\_1*. The patterns contributing to this difference were derived from the extension of three activities: *O\_Create Offer*, *O\_Sent*, and *O\_Created*. The key discriminative rule identified is: ‘Selected’ = ‘True’. According to the report by the *BPIC2017* challenge winners,<sup>5</sup> this feature indicates whether the customer signed the offer, but on its own, it leads to final loan approval in only 55% of cases. However, we found that if any of these three activities are executed and the ‘Selected’ value is ‘True’, the probability of loan approval increases to 68%. The winners’ report also highlighted *O\_Sent* and *W\_Call* after offer as activities linked to positive outcomes, suggesting that an offer is more likely to be selected when the customer is frequently notified. Another impactful pattern we identified involved *O\_Sent* eventually followed by *W\_Call* after offer while ‘Selected’ = ‘True’, which led to loan acceptance in 75.5% of cases. The overlap between the patterns discovered by CA-IMPRESSED and the findings in the literature highlights the method’s capability to uncover patterns that can be understood and utilised by domain experts.

The most notable improvement in AUC within the *BPIC2011* dataset occurred at prefix 25 of *BPIC2011\_3*. Analysis of the decision tree built on the encoded patterns revealed that the most critical pattern involves one repetition of *AC370000* followed eventually by *370488G*, with the

rule: ‘Diagnosis code’ = ‘M13’. Notably, while the IMPRESSED method identified the control flow aspect of this pattern, it was associated with positive outcomes in only 64% of cases. Incorporating the data condition (‘Diagnosis code’ = ‘M13’) into the control flow pattern substantially increased the prediction accuracy, boosting the positive outcome rate to 87%. This indicates that neither control flow nor data attributes alone are sufficiently predictive; rather, their aggregation is necessary to achieve the highest accuracy, as relying solely on the *Diagnosis code* resulted in positive outcomes in only 66% of cases.

### 6.3. Pattern discovery runtime overview

The benefit of using CA-IMPRESSED clearly depends on whether the outcome is affected by context-aware patterns. Nevertheless, the evaluation shows that even when this is not the case, CA-IMPRESSED maintains the same predictive power of the baseline while providing a powerful means to explore potential relations with multiple process perspectives. In general, we observed an improvement ranging from 2% to 15% in different event logs.

This higher predictive power, however, comes with a trade-off in computational cost. The runtime for each method is detailed in Appendix C, and we discuss key trends here. The increase in the runtime of CA-IMPRESSED compared to IMPRESSED is often proportional to the performance improvement. A longer runtime allows the algorithm to explore more complex patterns and relationships that may remain hidden in the baseline model, leading to better results. This is evident in datasets like *BPIC2011\_3*, *BPIC2017\_1*, and *BPIC2017\_2*, where the highest observed increases in runtime (87.2%, 161.5%, and 165.6%,

<sup>5</sup> [https://ais.win.tue.nl/bpi/2017/bpi2017\\_winner\\_professional.pdf](https://ais.win.tue.nl/bpi/2017/bpi2017_winner_professional.pdf)

respectively) correspond to substantial improvements in prediction performance (15%, 13%, and 15%, respectively). Conversely, in cases like BPIC2012\_1 and BPIC2017\_3, where runtime remained unchanged or increased marginally (9.6%), performance gains were minimal (0%–5% respectively).

However, there are exceptions to this trend. In the BPIC2011\_1, we observed only a 15.4% increase in the discovery run time, which leads to around 10% prediction improvement. Comparing BPIC2011\_1 with BPIC2011\_3 (which share similar characteristics) revealed that the difference stems from the number of dominant patterns identified by the Pareto Front after each discovery iteration. For example, in BPIC2011\_3, during the first iteration, both IMPressed and CA-IMPressed identified 10 single activities as dominant. However, due to the additional discriminative rule discovery step in CA-IMPressed, the runtime increased significantly. In contrast, in BPIC2011\_1, only two activities were initially identified as dominant in IMPressed, and one of them was later dominated when process context was considered in CA-IMPressed. As a result, the overall search space was reduced, leading to a smaller increase in runtime throughout the discovery process.

Another exception is seen in the Sepsis datasets. Despite noticeable increases in runtime (126% on average), the prediction performance did not improve. It can be attributed to specific characteristics of the dataset. The SEPSIS log has a high variant-to-trace ratio (0.91), indicating a highly unstructured process with many unique traces, making it difficult for the algorithm to identify patterns with high support across the log. Additionally, the high number of case and event attributes may introduce noise or redundancy, diverting computational effort toward irrelevant features.

In general, the runtime of both CA-IMPressed and IMPressed is influenced by the number of trace and variants. The shortest runtime was observed in the smallest log (PRODUCTION), taking approximately 4 and 5 min, respectively, for the entire discovery process across all prefixes. In contrast, the longest runtime occurred in the log with the highest number of traces (BPIC2012), requiring an average of around 282 and 360 min, respectively. In future work, implementation optimisation and multiprocessing can reduce the runtime and make it feasible for larger event logs. Looking specifically at CA-IMPressed, the number of case and event attributes caused a longer runtime increase compared to the baseline method.

## 7. Process context-aware explanation evaluation

This section presents the evaluation of the obtained process context-aware explanation. The goal of this evaluation is to determine how effectively the surrogate model replicates the behaviour of the black-box model using context-aware patterns (with multiple encoding scenarios) compared to solely relying on control-flow patterns. We report the predictive performance of the black-box model for each dataset in Appendix B. Overall, the predictive models demonstrate strong performance in most cases, ensuring reliable label assignment in the synthetic neighbourhood.

In the following, we first focus on the results of the *quantitative analysis*, where we compared the LF of the explanation obtained from two proposed encoding approaches (FCP and DCP) with the two baseline approaches (CFP and CALL) using Eq. (1). The objective of the evaluation is to determine how well the surrogate model trained with each encoding can approximate the local predictive accuracy compared to the original black-box model, i.e., around the generated synthetic neighbourhood. We later present two examples in *qualitative comparison* to demonstrate the application of the proposed method in real-life scenarios and to compare the explanations obtained using different encodings and pattern discovery strategies.

### 7.1. Quantitative analysis

Fig. 10 shows the results for the BPIC2011, BPIC2017, and PRODUCTION datasets, while Fig. 11 shows the results for the BPIC2012 and SEPSIS datasets, measuring the LF, along with the numbers of discovered patterns. The results are discussed for each event log below, while the general overview is presented in 7.1.6:

#### 7.1.1. BPIC2011

The BPIC2011 event logs include healthcare-related processes with high variants, where the process outcome of each variant refers to some temporal control-flow pattern. Interestingly, the BPIC2011 datasets contain a large number of event attributes within each dataset, leading to an overall larger number of discovered patterns.

For the LF, we observe that across all the BPIC2011 variants, the FCP and DCP consistently outperform the two baselines. Between the context-aware methods, we observe variations in terms of their performance, with FCP outperforming DCP on BPIC11\_f2, BPIC11\_f4, and BPIC11\_f3 (for this case, the longest prefix is the exception, where DCP outperforms FCP). Meanwhile, for the BPIC11\_f1, the two proposed methods vary in performance depending on the prefix length under analysis, performing comparably on average. When comparing the two baselines, we observe some variability as well; for the BPIC11\_f1 dataset, which has a more balanced class ratio (40%), the CFP method outperforms CALL for medium-length traces. In general, for longer traces, CFP outperforms CALL, suggesting the high dimensionality of the feature vector that is leveraged by CALL can worsen performance as opposed to only leveraging control-flow patterns.

Overall, for the BPIC2011 datasets, the two proposed encoding methods produce a more faithful surrogate model when compared to the baseline encodings. This is also reflected by the larger number of discovered patterns, which help the surrogate models using FCP and DCP to obtain a more faithful representation to the black-box. Between baselines, the CALL method performs either better or on par with CFP, depending upon the labelling.

#### 7.1.2. BPIC2017

For all variants of this log, LF shows minimal variations across prefix lengths. However, FCP and DCP consistently outperform both baselines, suggesting that CA-IMPressed captures interesting patterns more effectively than IMPressed. Even when different process perspectives are integrated with control-flow patterns in CALL, it does not achieve the best results, as some patterns may still be overlooked during the discovery. In BPIC2017\_1 and BPIC2017\_2, FCP and DCP maintain high LF across all prefix lengths, with a slight upward trend, indicating stable and high-quality LF for these methods as more information is incorporated (i.e., with longer prefixes). However, these two datasets differ significantly in the number of discovered patterns. In BPIC2017\_1, for example, as prefix length increases, the difference of discovered patterns between the proposed methods and the baseline methods also increases. On the other hand, for BPIC2017\_2, while this increase in difference is present for short prefixes, for very long ones (30) the number of discovered patterns by CFP and CALL suddenly increases to match the one of FCP and DCP. In contrast, we observed some fluctuation in LF over prefix lengths for all encoding scenarios. Notably, CALL has a lower lf, reinforcing the strength of FCP and DCP over other methods. Overall for BPIC2017, regarding discovered patterns, all methods perform similarly with shorter prefixes, but baseline methods show a sharp decline as prefix length increases. This likely reflects the focus of GA on event and case attributes over control-flow modifications for longer prefixes.

### 7.1.3. Production

In the PRODUCTION dataset, methods utilising context-aware patterns (FCP and DCP) consistently surpass the baselines; however, at maximum prefix length FCP gets close to the baseline performance. Notably, the CALL consistently underperforms compared to other methods, inferior to even the CFP. The number of patterns discovered by FCP and DCP converges with shorter prefixes, while the proposed methods consistently uncover more patterns than the baseline, leading to better mimicry of the black box's local decision boundary. Overall, DCP emerges as the most effective approach across different prefix lengths. This superior performance likely stems from the dataset's labelling, which relies on the event attribute 'Quantity rejected' being greater than 1 in any event within a trace (Teinemaa et al., 2019). Although this attribute is removed during the preprocessing of the dataset, by exclusively leveraging event attributes in the discriminative rule detection step, DCP can more directly capture correlations between event attributes and the outcome.

### 7.1.4. BPIC2012

For shorter prefixes across BPIC2012 variants, FCP and DCP outperform the baselines, with performance differences narrowing as prefix length increases. The largest differences are observed for the BPIC2012\_2 and BPIC2012\_3 event logs, which interestingly also have a more imbalanced label distribution compared to the BPIC2012\_1 dataset, which shows smaller differences between the three proposed methods. The labels for all BPIC2012 variants are based on the occurrence of final offer activities (Offer Accepted, Offer Rejected, or Offer Cancelled). As prefix length increases, patterns are more likely to include these activities, making the surrounding context (event and case attributes) less impactful. As a result, the performance gap between context-aware patterns and control-flow patterns is minimal.

However, in general, for the BPIC2012 datasets, context-aware methods (FCP and DCP) excel for LF, especially for complex or nuanced cases, as they capture specific context around each instance. Since BPIC2012 has only one static attribute, the differences between FCP and DCP are minimal.

### 7.1.5. Sepsis

For SEPSIS, the FCP and DCP always outperform the two baselines. No significant differences are observed in terms of LF between the two context-aware pattern methods. Surprisingly, CFP, which only uses the control-flow patterns, outperforms the other baseline, which adds all other attributes to the control-flow patterns (CALL). This could be due to the fact that the surrogate model receives as input a vector that has a large dimensionality (varying depending upon the prefix length, around a thousand features), meaning it can negatively influence the performance of the surrogate model, especially around the synthetic neighbourhood. Comparing surrogate model performance to the black-box model performance (Fig. B.16) reveals some interesting patterns. In all SEPSIS datasets, LF for surrogate models  $M_w$  drops at prefix length 9, aligning with a performance dip in the black-box model  $M_b$ . Although the Sepsis\_1 registers the least predictive performance, with an average AUC of around 0.6, we observe that on average, the proposed encodings DCP and FCP can nonetheless accurately mimic the predictions made by the black-box model, with an average LF of around 95%. The same behaviour is seen for Sepsis\_3 (average AUC of 0.7), where although the LF scores are around 80%, the proposed encoding methods can outperform the baselines.

For the SEPSIS datasets, the proposed methods outperform the baselines in LF, though the improvement varies across datasets. In datasets like Sepsis\_1 and Sepsis\_3, the proposed methods perform significantly better in LF, with Sepsis\_1 also showing a notable increase in the number of discovered patterns. For Sepsis\_2, differences in LF diminish as trace prefix lengths increase, as the activities that influence the outcomes (Admission NC) tend to occur later in the traces, allowing all methods to detect similar patterns. While differences in the number of discovered patterns are pronounced in some cases (e.g., Sepsis\_1), they are less significant in others, due to factors such as trace characteristics and black-box model performance.

### 7.1.6. Quantitative analysis overview

In general, when looking at the different event log characteristics in Table 6, some interesting observations emerge. For smaller event logs with a larger number of event attributes (BPIC2011, PRODUCTION), we observe, on average, both an increase in the differences in the fidelity between the proposed methods and the baseline encodings and in the differences in the number of discovered patterns; on average, the DCP and FCP encompass more patterns when compared to CFP and CALL. For larger logs with more event attributes, like BPIC2017, while there is a difference in the number of discovered patterns, this does not also translate into larger differences in the fidelity between the proposed methods and the baselines. A confounding factor, in this case, might be the length of the considered prefixes, as for larger datasets with longer prefixes (BPIC2017 and BPIC2012), we see an overall smaller difference in LF between the methods, despite the difference in the number of patterns discovered.

The time required to generate the explanations with each proposed encoding method is presented Appendix C, we discuss the main trends here. In terms of runtime, encodings leveraging CA-IMPressD (DCP and FCP) generally exhibit longer runtimes compared to those using IMPressD (CFP and CALL), with the increase often correlating with performance improvements. This is particularly evident in the BPIC11\_f1, BPIC11\_f2, and BPIC2017\_1 datasets, where the runtime increases by 180%, 102%, and 165%, respectively, alongside an average improvement in LF of 15%. In contrast, for the SEPSIS dataset variants, although an improvement of approximately 10% in LF is observed, the difference in the number of discovered patterns and runtime between methods is smaller, with an average runtime increase of 30% from DCP and FCP to CFP and CALL. This can be attributed to the high variance in the event logs and the black-box model's reliance on trace and event attributes, which DCP and FCP leverage to better replicate its predictions. Finally, for the BPIC2012 datasets, we observe a similar smaller increase in runtime (30%), coupled with small differences in terms of LF between the different encodings.

In general, the differences between the encodings using CA-IMPressD and those using IMPressD can depend upon a series of factors such as the number of variants in the synthetic neighbourhood, the number of events and trace attributes, as well as the number of individual activities. Overall, the shortest runtime across the four encodings was registered for the BPIC2012 event logs, with FCP and DCP requiring a time of 80 and 70 s, respectively, while CALL and CFP requiring 50 s across all prefixes. The longest runtimes across the board were registered for the BPIC2017 event logs, with FCP and DCP requiring on average 300 and 250 s respectively, and with CALL and CFP requiring a much lower runtime of 100 s on average across all prefix lengths. The difference between the two event logs can be attributed to the larger number of events and case attributes in the BPIC2017 logs.

Overall, the main objective of this analysis was to determine whether the proposed approaches improve the LF compared to the baseline methods. Indeed, the two proposed encoding methods can offer a more robust surrogate model within the synthetic neighbourhood, as shown in the improvement in LF. Especially the methods leveraging the context-aware patterns (i.e., FCP and DCP) were shown to be the most promising, consistently obtaining the best results across all datasets. Across the board, the CALL baseline was able to obtain moderately good results, still surpassing the CFP baseline, for most datasets, except for the SEPSIS and PRODUCTION datasets.

## 7.2. Qualitative comparison

In this section, an example of the explanations returned by the different proposed encoding and pattern discovery methods are shown, by leveraging an example from BPIC2017\_1 dataset. Moreover, we attach a second example from Sepsis\_1 in Appendix B. The aim of this analysis is to illustrate how end users can apply the proposed method

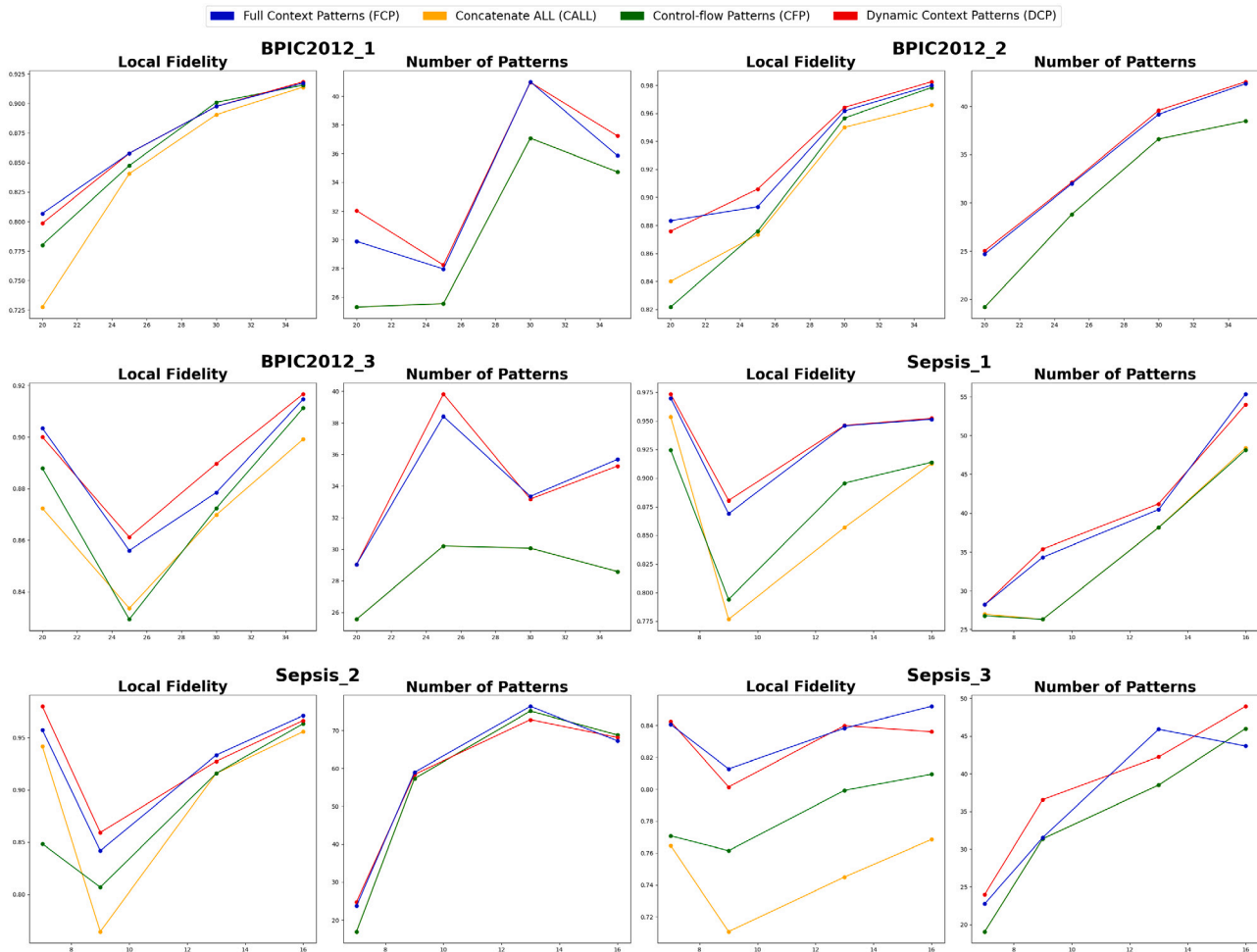


Fig. 11. Local fidelity comparison for different encoding scenarios.

Table 9

Details of the trace with ID Application\_711722413 (Amount = 5000, Loan Goal = Car, Application Type = New credit, Selected = False, CreditScore = 0.0, Accepted = True).

	concept:name	time:timestamp	timesincecasestart	org:resource	Action	lifecycle:transition
0	A_Create Application	2016-01-21 14:51:02	0.0	User_1	Created	complete
1	A_Submitted	2016-01-21 14:51:03	0.8	User_1	statechange	complete
2	W_Handle leads	2016-01-21 14:51:03	3.9	User_1	Created	schedule
3	W_Handle leads	2016-01-21 14:52:31	1.5	User_1	Deleted	withdraw
4	W_Complete application	2016-01-21 14:52:31	1.5	User_1	Created	schedule
5	A_Concept	2016-01-21 14:52:32	1.5	User_1	statechange	complete
6	W_Complete application	2016-01-22 18:40:01	1669.0	User_12	Obtained	start
7	A_Accepted	2016-01-22 18:44:34	1674.5	User_12	statechange	complete
8	O_Create Offer	2016-01-22 18:49:23	1678.3	User_12	Created	complete
9	O_Created	2016-01-22 18:49:25	1678.4	User_12	statechange	complete
10	O_Sent (mail and online)	2016-01-22 18:50:44	1679.7	User_12	statechange	complete
11	W_Complete application	2016-01-22 18:50:44	1679.7	User_12	Deleted	complete
12	W_Call after offers	2016-01-22 18:50:44	1679.7	User_12	Created	schedule
13	W_Call after offers	2016-01-22 18:50:44	1679.7	User_12	Obtained	start
14	A_Complete	2016-01-22 18:50:44	1679.7	User_12	statechange	complete
15	W_Call after offers	2016-01-22 18:57:52	1686.8	User_12	Released	suspend
16	W_Call after offers	2016-01-26 08:02:34	6791.5	User_3	Obtained	resume
17	W_Call after offers	2016-01-26 08:02:55	6791.9	User_3	Released	suspend
18	O_Create Offer	2016-01-29 14:05:43	11474.7	User_85	Created	complete
19	O_Created	2016-01-29 14:05:44	11474.7	User_85	statechange	complete
20	O_Sent (mail and online)	2016-01-29 14:05:57	11474.9	User_85	statechange	complete
21	A_Cancelled	2016-02-29 07:01:19	55690.3	User_1	statechange	complete
22	O_Cancelled	2016-02-29 07:01:19	55690.3	User_1	statechange	complete
23	O_Cancelled	2016-02-29 07:01:19	55690.3	User_1	statechange	complete
24	W_Call after offers	2016-02-29 07:01:19	55690.3	User_1	Deleted	ate_abort

and explore the practical relevance of the final explanations beyond their mere LF values.

A trace is chosen from the test set of the BPIC2017\_1 dataset with a negative outcome prediction (offer not accepted) as an inquiry trace

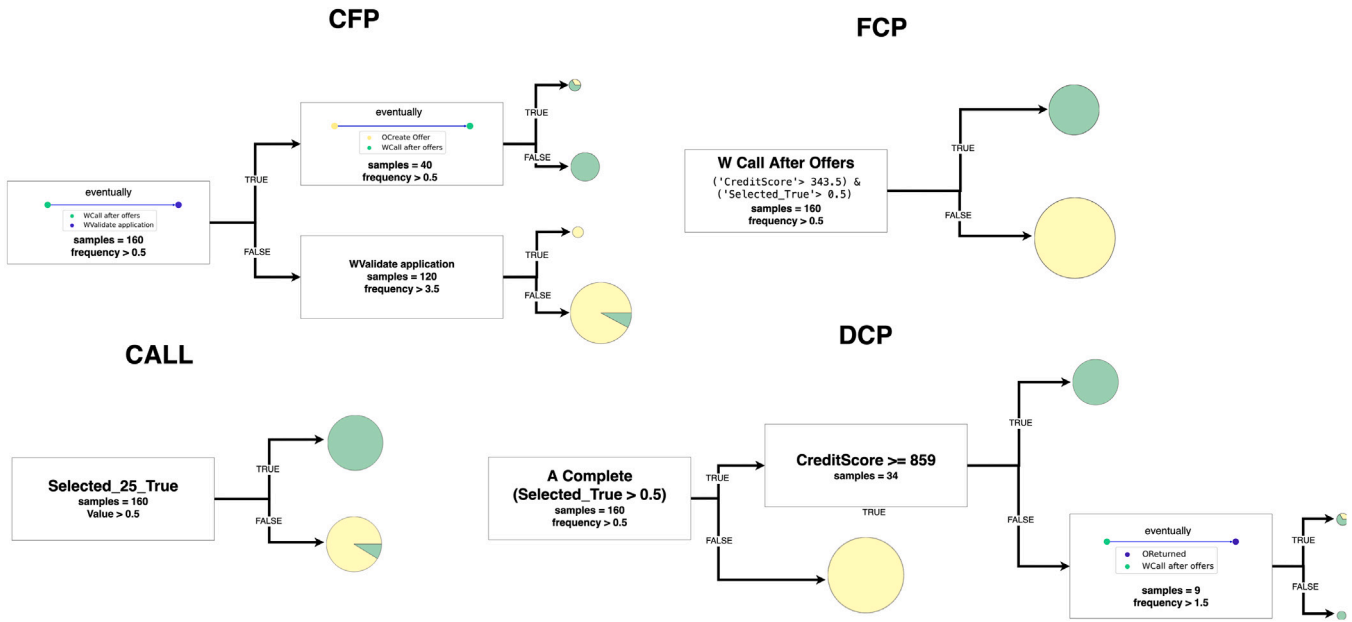


Fig. 12. Four types of explanation for inquiry trace of Application\_711722413.

**Table 10**  
Fidelity results and number of patterns discovered for Application\_711722413.

Encoding	Local fidelity	# Patterns
FCP	99%	31
DCP	99%	31
CALL	92%	21
CFP	82%	21

shown in Table 9, as a complete trace, with a length of 25. This trace follows an application for a 5000\$ loan categorised under the “Car” loan goal and tagged as a “New credit” application, with a credit score of 0. The sequence begins with the creation (A\_Create Application) and submission (A\_Submitted) of the application, followed by handling leads (W\_Handle leads). Shortly afterwards, a task to complete the application (W\_Complete application) is initiated and eventually validated (A\_Concept). After a substantial delay, a review marks the application as accepted (A\_Accepted) around 1673 min later, prompting an offer creation (O\_Create Offer) which is subsequently sent to the applicant both by mail and online. Following the offer’s dispatch, a scheduled follow-up call (W\_Call after offers) is frequently deferred and resumed over several days. An extended delay eventually occurs, and the application undergoes multiple cancellations (O\_Cancelled) and is marked as closed by the end of February.

Table 10 presents the resulting LF for the final explanation using different encoding strategies. In this case, FCP and DCP achieved the highest LF of 99%, followed by CALL with an LF of 92%. In contrast, the only control-flow dependent method obtained the lowest LF of 82%. While the LF values are relatively close to each other, we observe four completely different DTs and, consequently, different explanations obtained from each method in Fig. 12. DT nodes represent discovered patterns with their conditioned frequency or encoded attributes with required values, depending on the encoding scenario. Edges indicate whether the condition defined in each node is satisfied. Final nodes (leaves) denote the predicted outcome for traces following that path. The pie chart size reflects the sample size at the leaf, with green indicating a flipped (desired) outcome and yellow signifying the predicted outcome matches the inquiry trace.

The obtained explanation from the CALL relies on a single event attribute: ‘Selected’ being ‘True’ at the 25th prefix. As discussed earlier (in Section 6.2), although the attribute is correlated to the offer

being accepted, it does not hold for all cases. Additionally, such an explanation lacks a control flow perspective, not offering an actionable recommendation within the process context. On the other hand, the explanation obtained from CFP focuses on only control flow patterns and does not support other perspectives. The most important pattern in the first node of the presented DT (WCall after offers eventually followed by WValidate application) is aligned with the pattern that has been reported in (Buliga et al., 2024) for another log related to loan applications. In general, this pattern implies that validating the offer after calling the customer is linked to the desired outcome; however, this pattern exists in the inquiry trace, but still, the outcome is not desired. Therefore, we still miss a comprehensive explanation for this case.

Looking at the DTs obtained with FCP and DCP, we achieve a more in-depth explanation. When we incorporated both case and event attributes inside the control flow pattern using FCP, the explanation says that not only do we need to call the customer after the offer (WCall after offers) at least once, but also that the ‘Selected’ attribute should be ‘True’ and the ‘Credit score’ should be higher than 343.5 to the loan being granted. This explanation reflects both on important activities to take and required attributes in combination. By focusing on the event attributes in the pattern discovery stage (DCP), we still have ‘Credit Score’ as an important node in the DT, but a new pattern now popped up in the last DT’s node, suggesting returning the offer after calling the consumer (WCall after offers eventually followed by OReturned) which does not occur in the inquiry trace. This process pattern flipped the outcome in a few cases followed the corresponding path, offering an opportunity to achieve a desired outcome even with a low credit score.

This qualitative analysis highlights the importance of integrating multiple process perspectives to deliver actionable and context-aware explanations, enabling better decision-making and identifying opportunities for improving process outcomes.

## 8. Discussion

In this section a summary of the results from Sections 6 and 7 are first presented. Afterwards, a discussion is introduced on the implications of using the context-aware process pattern discovery within either the predictive context of Section 6 and the explanation context in Section 7, and the interplay between the two. Finally, the section

ends with a discussion of the limitations of the approach and potential solutions on how to address them in future work.

When looking at the results obtained from Section 6 and Section 7, some interesting trends emerge. When comparing IMPressed and CA-IMPressed in Section 6, we generally observe an improvement in the prediction performance of the patterns discovered with CA-IMPressed of 2%–15%, whilst also observing an increase in the runtime of 9.6%–170% depending upon the dataset. The only exception is the SEPSIS event log, where we observe a large difference in the runtime of the methods whilst the predictive performance of the discovered patterns remains the same.

In Section 7, we observe similar trends. On average, we observe an improvement of the context-aware encoding (FCP and DCP) compared to only the control-flow patterns encoding (CALL and CFP) in terms of local fidelity (LF) within the range of 2% to 20% and an increase in runtime from 30% to 180%, similar to what was observed above. The lowest improvements were observed from the BPIC2012 datasets, whilst the largest improvements were registered for the BPIC2011 and Production datasets.

The only contradictory trend between Sections 6 and 7 relates to the SEPSIS log. While in Section 6, the differences between IMPressed and CA-IMPressed are negligible, also reflected in the similar number of discovered patterns, in Section 7, a different result is observed. Specifically, DCP and FCP achieve on average 10% better LF results. Moreover, we also observe an increase in the number of discovered patterns, especially for Sepsis\_1 and Sepsis\_3. This suggests that, while the overall log does not reveal a significant difference between contextual and control-flow patterns, examining each inquiry trace individually may uncover predictive context-aware patterns specific to the case at hand. The inherent high variability within the SEPSIS dataset poses challenges for identifying overarching general context-aware patterns, yet our method excels when applied to smaller, contextually relevant neighbourhood of inquiry traces. A similar observation has also been discussed in literature when shifting from local to global explanations (Setzu et al., 2021); local explanations do not necessarily hold for the whole dataset. For example, in qualitative analysis on Sepsis\_1 (in D), both DCP and FCP emphasise the resource aspect of the control-flow pattern, whereas these patterns remain hidden when using CA-IMPressed on the entire event log. This observation aligns with the low predictive performance of the black-box model on these logs, suggesting that building a general prediction model for such highly variable data remains a challenge.

On the topic of predictive performance, a lower-performing black-box model leads to explanations from XPPM methods that, while reflecting the model's internal reasoning, do not necessarily represent the actual patterns present in the log. As highlighted in previous work (Rizzi et al., 2020b), explanations can be valuable in such cases for improving prediction performance. This highlights the usability of these explanations with different goals: when one obtains a faithful black-box model (i.e., high AUC), one could use explanations also to obtain process insights, while, in case of a lacklustre black-box model, such explanations could be leveraged to debug the black-box model and understand its spurred learned correlations. Future work could include user-based evaluations to investigate this phenomenon further and validate these hypotheses. Regardless, the proposed methods enhance users' understanding of model behaviour, offering deeper interpretability.

**Limitations.** As per the limitations of the proposed approach, a couple of challenges arise, especially for the local explanation aspect: to achieve a high-performing black-box model, data quality also plays an important role. Indeed, data quality poses a common challenge to the proposed method, as it does for existing PPM techniques. Therefore, applying standard data-cleaning procedures prior to adopting the proposed method is essential. In this regard, CA-IMPressed could also support the discovery of noisy or biased patterns with tailored interest

functions. To this end, user validation is needed to verify the discovered pattern and identify the potential issues from the original log. While this matter was out of the scope of this work, future work could explore the potential of the CA-IMPressed discovering noisy patterns in collaboration with expert users.

Another challenge is to maintain the performance of the black-box model over time in case of concept drift. It is worth mentioning that PABLO relies on offline training, which necessitates updating the black-box model in the event of concept drift. Using the provided explanation for the purpose of black-box debugging could potentially assist in detecting changes in the black-box model behaviour caused by the concept drift. While exploring concept drift falls outside the scope of this research, possible solutions to alleviate this could integrate online concept drift detection or online learning mechanisms in future work to ensure the black-box model remains up-to-date if we aim to use explanation to obtain process insights. In this regard, leveraging methods such as those proposed in (Pasquadibisceglie et al., 2023) for training the black-box model is highly recommended.

The last consideration concerns the generalisability of the proposed method. Our experiments on a wide range of real-life event logs from healthcare, finance, and industrial domains demonstrate that the method is not domain-specific and outperforms baseline methods across different sectors. However, its effectiveness can be further validated through domain-specific case studies. While the method is particularly suited for processes with a certain degree of flexibility, such as variations in activity sequences, it also provides valuable insights into highly rigid processes by analysing case and event attributes. Future work could explore the utility and understandability of the proposed explanations across industries by involving domain experts.

## 9. Related work

Since the focus of the paper is on the eXplainable Predictive Process Monitoring (XPPM) and process pattern discovery, we have organised the related work into the following two subsections:

### 9.1. Explainable predictive process monitoring

Most works in the XPPM domain leverage post-hoc XAI techniques to offer insights into model predictions made by black-box models. These explanations can be broadly categorised into factual and counterfactual explanations (Huang et al., 2021). *Factual* explanations clarify why a particular prediction was made by highlighting key attributes or patterns that contributed to the outcome. *Counterfactual* explanations, on the other hand, explore alternative scenarios, showing how altering certain aspects of a case or event could have led to a different outcome.

In terms of scope, local explanations focus on a specific instance or trace, providing insight into why a particular prediction was made for that specific case. Global explanations, conversely, aim to offer a broader understanding of the overall behaviour of the model, shedding light on patterns across all cases and events. While both are important, global explanations are often less applicable in highly complex event logs with numerous perspectives, whereas local explanations tend to offer more actionable insights. For this reason, in this work we focus on providing local explanations for process analysts.

Regarding factual local explanation, earlier studies utilised methods like Local Interpretable Model-Agnostic Explanations (LIME) (Rizzi et al., 2020a), gated graph neural networks (Harl et al., 2020), and attention-based LSTM (Wickramanayake et al., 2023) to generate local factual explanations by analysing case and event attributes. Another approach proposed building surrogate models using features derived from intermediate latent representations in deep neural networks (Mehdiyev and Fettke, 2021). However, these methods primarily focus on factual explanations.

Counterfactual explanations in the XPPM context have also gained attraction, but applying traditional methods often results in unrealistic



Fig. A.13. The top four important patterns discovered from the resource-based log.

explanations. This is due to the failure to account for process constraints and the complexity of event logs, which often span multiple perspectives (Huang et al., 2021; Hsieh et al., 2021). Some methods have begun adapting counterfactual generation to the process domain. Huang et al. introduced LORELEY (Huang et al., 2021), which integrates control-flow constraints to produce more realistic counterfactuals, though this approach limits changes to case attributes alone. Additionally, their evaluation focused on only one event log (BPIC2017) and relied on prior knowledge of case attributes influencing the process outcome, which may limit the generalisability of their findings. Similarly, DiCE4EL (Hsieh et al., 2021) builds on DiCE (Mothilal et al., 2020) to address process-specific challenges but requires significant domain knowledge and limits the exploration of alternative process paths. Hundogan et al. proposed CREATED (Hundogan et al., 2023), leveraging an Evolutionary Algorithm to generate new counterfactual sequences based on existing traces. While this method produces feasible counterfactual sequences, it does not provide clear explanations or derive patterns to ensure confidence in the predicted outcomes.

Nonetheless, little effort has been paid to developing methods that are able to produce both types of explanations while simultaneously providing control-flow-aware explanations. A single solution has been proposed thus far by Buliga et al. within the PAttern Based Local explanation (PABLO) (Buliga et al., 2024) framework, where the authors propose a method that is able to generate both factual and counterfactual control-flow aware explanations, specifically in terms of discriminative process patterns. This method has the added benefit of also allowing the instantiation of examples satisfying the returned control-flow explanations due to the use of synthetic neighbourhood that is generated around a single trace of interest that is used to train a Surrogate Model, from which explanations are derived. However, PABLO focuses on the only the control flow perspective and offers limited support for the only case attributes, while other event attributes are not considered. This work extends the PABLO framework to consider multiple process perspectives while delivering a combined factual and counterfactual local explanation.

## 9.2. Predictive process pattern discovery

The discovery of start-to-end process models has long been the main focus of process discovery research (Augusto et al., 2018). However,

high process variability makes extracting a single start-to-end process model challenging (Slaats, 2020). Recent studies have explored alternative approaches to manage process variability, aiming to reveal data-driven links between events and recurring patterns (Vazifehdoostirani et al., 2022). As a result, in recent years, there has been growing interest in detecting behavioural patterns from process event logs (Tax et al., 2016; Peeva et al., 2022; Vazifehdoostirani et al., 2023b; Maggi et al., 2013). This shift reflects the increasing recognition that not all insights can be captured through a complete process model (Tax et al., 2016) and that understanding localised or recurring patterns of behaviour may offer more actionable insights, particularly when it comes to improving or predicting process performance (Vazifehdoostirani et al., 2022).

Two main research streams have emerged within this pattern detection approach. The first stream focuses on extracting patterns based on *intrinsic* properties of the event logs, such as the frequency or support of a given pattern. These methods aim to identify patterns that occur frequently and are thus representative of typical process behaviour (Tax et al., 2016). In contrast, the second stream focuses on *extrinsic* properties, specifically the impact of these patterns on process outcomes or performance (Vazifehdoostirani et al., 2023b). Given that the objective of pattern discovery in this work is to identify patterns based on their effect on process outcomes, our approach aligns more closely with the second research stream. We are particularly interested in patterns that are not only frequent but significantly differentiate between successful and unsuccessful outcomes.

Most existing approaches to process pattern discovery, like Local Process Models (LPMs) (Tax et al., 2016), IMPressed (Vazifehdoostirani et al., 2023b), and Declare (Pesic et al., 2007), focus on identifying control-flow relations. While some of these methods have been extended to incorporate other process perspectives, such as data-aware Declare (Maggi et al., 2013) and interest-driven LPMs (Tax et al., 2018), they remain focused on uncovering frequent patterns that, while informative, do not necessarily reveal factors driving variations in process outcomes. IMPressed, instead, aims to discover predictive patterns, but it largely overlooks the role of other perspectives in shaping these patterns. This work extended IMPressed to address these needs.

## 10. Conclusion

In this paper, we addressed the challenge of providing combined factual and counterfactual explanations for different process perspectives. Central to our approach was the adaptation of the PABLO framework, augmented with multi-perspective insights that incorporate control flow, time, resource, and other data attributes. A primary contribution of our work is a novel method for discovering context-aware predictive process patterns, enabling the extraction of meaningful process patterns from a synthetic neighbourhood surrounding the inquiry trace. This approach empowers the surrogate model, enriching it to deliver explanations that account for multiple process perspectives, unlike prior models that often focused on one aspect at a time.

Additionally, this paper introduced innovative encoding methods for the surrogate model, embedding it with context from the process to support richer process context-aware explanations. Through rigorous evaluation, we demonstrated the advantage of context-aware patterns compared to only control flow patterns in detecting influential process behaviours on the outcome. These patterns deliver not only stronger predictive power but also enable context-aware explanations with greater fidelity, effectively bridging the gap between data-driven and process-oriented insights. Our qualitative comparison highlighted the advantage of context-aware explanation beyond improvements in fidelity values in delivering actionable insights illustrating the interplay of multiple process perspectives by showcasing two examples from different real-life processes.

We adapted the neighbourhood generation approach from (Buliga et al., 2024) with adjustments to include all process perspectives, but ensuring the feasibility of generated traces remains a challenge. While conformance-checking measures in the genetic algorithm address control flow, they do not fully verify data payload feasibility, leaving a limitation for this work.

While this work demonstrates promising advancements, there are exciting opportunities for future exploration. Refining the feature expansion step with domain-specific metrics and expert input in future work could enhance accuracy and prevent redundancy. Future research could also investigate adaptive mechanisms to dynamically tailor the granularity of explanations according to different user requirements. Exploring less computationally intensive methods for optimal rule selection would improve scalability, and extending the approach to provide process context-aware global explanations offers valuable directions for future research. Moreover, a user-based study to measure the utility and understandability of these patterns would help determine how users interact with such explanations. In addition, future work could focus on optimising the complexity of the explanations to improve interpretability, for instance, by incorporating rule simplification techniques into the final surrogate model. Finally, developing practical guidelines to support business users in applying these explanations effectively enhances the applicability of our approach in future.

### CRediT authorship contribution statement

**Mozhgan Vazifehdoostirani:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Andrei Buliga:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Conceptualization. **Laura Genga:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Chiara Di Francescomarino:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Massimiliano Ronzani:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Chiara Ghidini:** Writing – review & editing, Validation, Supervision, Methodology, Formal analysis, Conceptualization. **Remco Dijkman:** Writing – review & editing, Validation, Supervision, Methodology, Formal analysis, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgements

This work was supported by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

## Appendix A. Discovered patterns for controlled evaluation

Here, we present the top four most important patterns discovered from the relabelled logs during the controlled evaluation in Section 6.1.2. The patterns in Figs. A.13, A.14, and A.15 are arranged from left to right in order of importance, from highest to lowest. The black arrows in the figures represent a direct relation, while the blue arrows present an eventual relation:

## Appendix B. Performance of black-box models

The results of the black-box model performance (measured by AUC over different prefix lengths) are presented in Fig. B.16.

The datasets exhibit different behaviours as prefix length increases. In BPIC2012 and BPIC2017, models consistently improve, with AUC rising from 0.7 to 0.95 in BPIC2012 and from 0.9 to 0.99 in BPIC2017. A similar trend is seen in most BPIC2011 variants, except BPIC2011\_f4, where performance slightly declines before improving at the longest prefix. In contrast, SEPSIS datasets shows more variability in performance. For Sepsis\_1, the model goes from 0.52 to 0.65 AUC, which is similar to the behaviour observed in (Teinemaa et al., 2019), due to the presence of a concept drift. For Sepsis\_3, although the predictive model achieves a performance of around 0.7 of AUC, indicating an average good performance. Finally, for the PRODUCTION a decent performance is observed, with AUC reaching around 0.75, which is due to the shift in the distribution of the length of traces from the training set to the test set, as discussed in Section 6.2.

## Appendix C. Runtime report

Table C.11 presents the runtime per event logs for the baseline and proposed methods in both evaluation sections. The Pattern discovery columns represent the runtime of the IMPressed and CA-IMPressed applied on the entire event log (except for the BPIC2017, which is sampled as mentioned in Section 5.2.1). The local explanation columns present the runtime of the PABLO pipeline with either IMPressed or CA-IMPressed depending on the encoding introduced in Section 4.2.

## Appendix D. Sepsis\_1 qualitative example

For the Sepsis\_1 dataset, we chose a trace belonging to the test set, where we obtained a negative prediction (patient has returned to the ER after more than 28 days of discharge) from  $M_b$ . Such a trace is shown in Table D.12. For readability purposes, we limit the attributes shown in the table to focus more on the rules provided by the DTs.

The trace represents the sequence of medical events for a 75-year-old patient diagnosed with the condition ‘KA’, capturing the progression of their treatment journey. It begins with the patient being registered in the emergency room ER Registration at 17:45 on December 25, 2013. Shortly after, the patient undergoes ER Triage and ER Sepsis Triage, indicating initial steps to assess the severity of their condition. The patient then receives Lactic Acid and Leucocytes measurements, followed by IV Liquid and IV Antibiotics, suggesting an urgent treatment approach likely aimed at addressing infection or

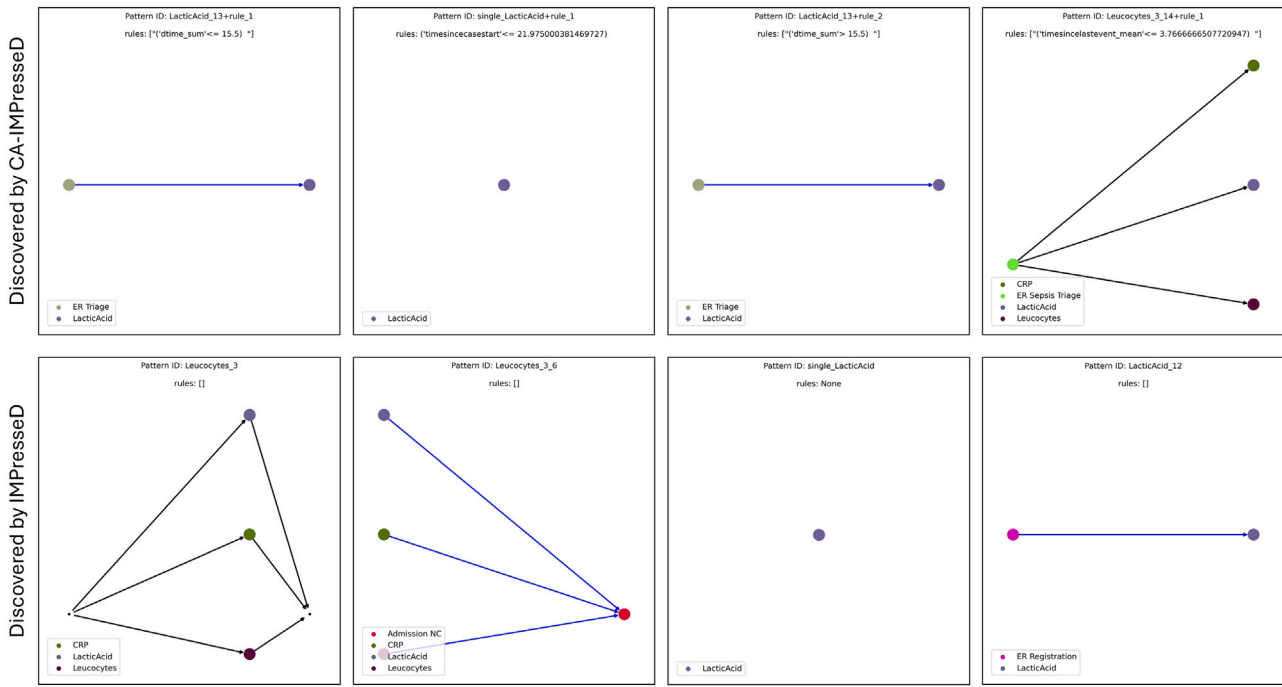


Fig. A.14. The top four important patterns discovered from the time-based log.

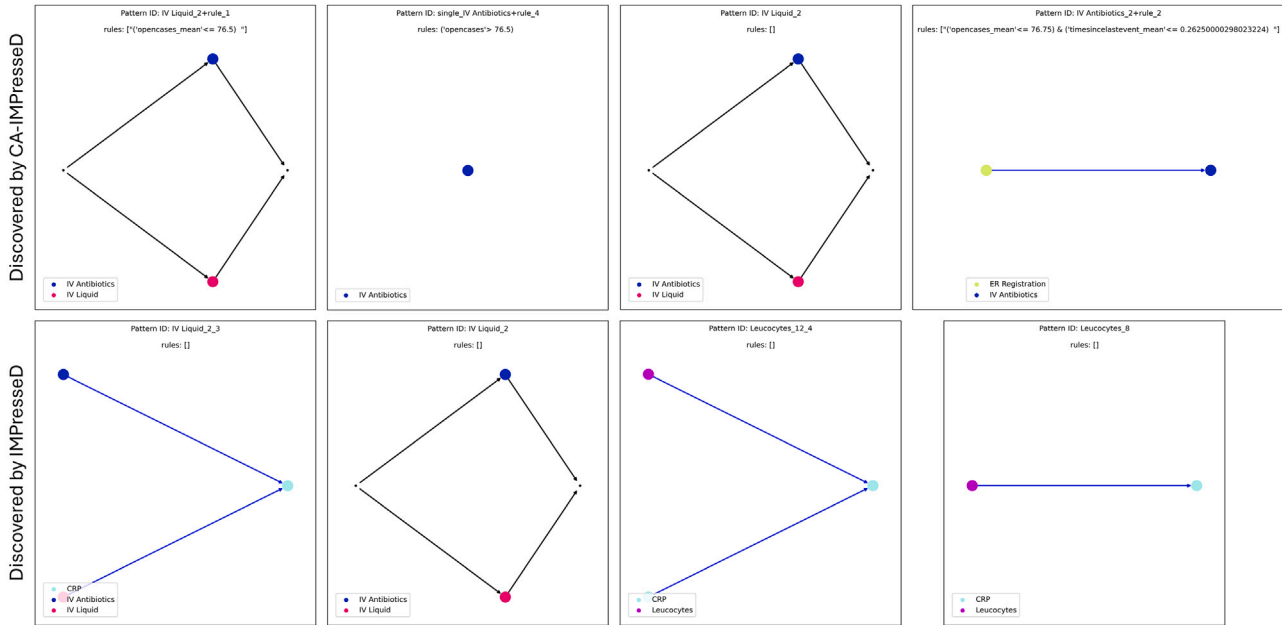


Fig. A.15. The top four important patterns discovered from the occupation-based log.

sepsis. Multiple clinical measurements, including CRP, Lactic Acid, and Leucocytes, are recorded periodically, showing some fluctuations. On December 27, further tests indicate a moderate response, with adjustments in CRP and Leucocyte levels. Over the next several days, these levels continue to be monitored, showing gradual improvement.

To gather explanations into both the reasons leading to the negative outcome and how we could flip the outcome to a positive one, we use the proposed methods to generate these explanations using the strategies presented in Section 4.2. The results for each of the encodings for the inquiry trace are reported in Table D.13. Here, we can observe that the FCP and DCP methods obtain similar values for the LF of 94.29%, 93.15%, respectively, while CFP obtains 87.48% and CALL a

lower score of 83.50%, showing the advantage of utilising the context-aware patterns. We also note that FCP and DCP find more patterns, 36 and 38, respectively, although even just using the control-flow patterns returns a set of 25 patterns.

In Fig. D.17, we compare the possible explanations obtained from DTs using different encoding strategies. Interestingly, the control flow aspect of the context-aware patterns presented in FCP and DCP explanations is different from the one in CFP, where the focus is only on the control flow. The higher fidelity values for FCP and DCP, as compared to CFP, indicate that these patterns were significant for outcome prediction only under specific data conditions associated with them. The shortest explanations are obtained from DCP and FCP. For DCP, the most important condition is the absence of a context-aware

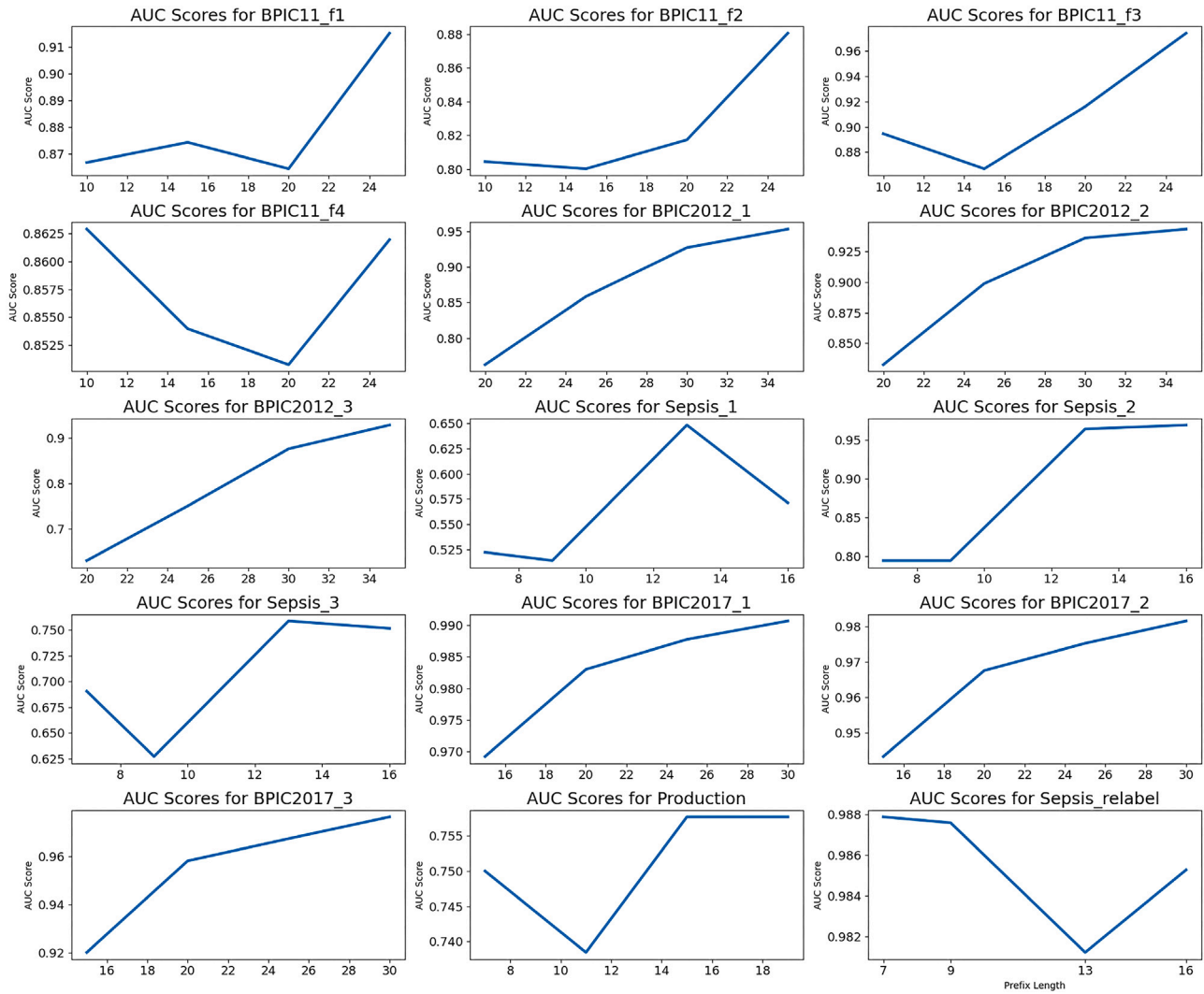


Fig. B.16. Predictive performance of black-box models across datasets and prefix lengths.

Table C.11  
Recorded run time over all prefixes (in seconds)

Log	Local explanation (Section 7)				Pattern discovery (Section 6)	
	CALL	CFP	FCP	DCP	IMPresseD	CA-IMPresseD
BPIC2012_1	65	65	91	86	14,160	14,160
BPIC2012_2	61	61	89	78	9240	11,640
BPIC2012_3	58	53	82	76	27,360	39,000
BPIC2012 average	61.3	59.6	87.3	80	16,920	21,600
BPIC2017_1	79	79	738	732	3900	10,200
BPIC2017_2	71	71	276	212	3840	10,200
BPIC2017_3	74	74	474	219	4380	4800
BPIC2017 average	74.6	74.6	496	387.6	4040	8400
Sepsis_1	87	87	127	117	1500	2700
Sepsis_2	85	85	131	114	900	2220
Sepsis_3	57	57	129	95	900	2280
Sepsis average	76.3	76.3	129	108.6	1100	2400
BPIC2011_f1	185	185	381	268	8580	9900
BPIC2011_f2	77	77	278	282	15,420	20,340
BPIC2011_f3	168	168	225	250	7500	14,040
BPIC2011_f4	261	261	264	349	15,060	19,200
BPIC2011 average	172.7	172.7	287	287.2	11640	15,870
Productions	124	124	172	165	240	300

**Table D.12**  
Details of the trace with ID XJA from Sepsis\_1 dataset.

Age	Diagnose	concept:name	time:timestamp	timesincecasestart	org:group	CRP	LacticAcid	Leucocytes	Label
75	KA	ER Registration	2013-12-25 17:45:10	0.0	A	0.0	0.0	0.0	false
75	KA	ER Triage	2013-12-25 18:01:15	16.1	C	0.0	0.0	0.0	false
75	KA	ER Sepsis Triage	2013-12-25 18:01:56	16.8	A	0.0	0.0	0.0	false
75	KA	LacticAcid	2013-12-25 18:11:00	25.8	B	0.0	1.9	0.0	false
75	KA	Leucocytes	2013-12-25 18:11:00	25.8	B	0.0	1.9	9.0	false
75	KA	CRP	2013-12-25 18:11:00	25.8	B	131.0	1.9	9.0	false
75	KA	IV Liquid	2013-12-25 18:37:36	52.4	A	131.0	1.9	9.0	false
75	KA	IV Antibiotics	2013-12-25 18:37:42	52.5	A	131.0	1.9	9.0	false
75	KA	Admission NC	2013-12-25 20:27:06	161.9	F	131.0	1.9	9.0	false
75	KA	CRP	2013-12-27 10:00:00	2414.8	B	113.0	1.9	9.0	false
75	KA	Leucocytes	2013-12-27 10:00:00	2414.8	B	113.0	1.9	6.2	false
75	KA	Leucocytes	2013-12-29 07:00:00	5114.8	B	113.0	1.9	6.7	false
75	KA	CRP	2013-12-29 07:00:00	5114.8	B	88.0	1.9	6.7	false

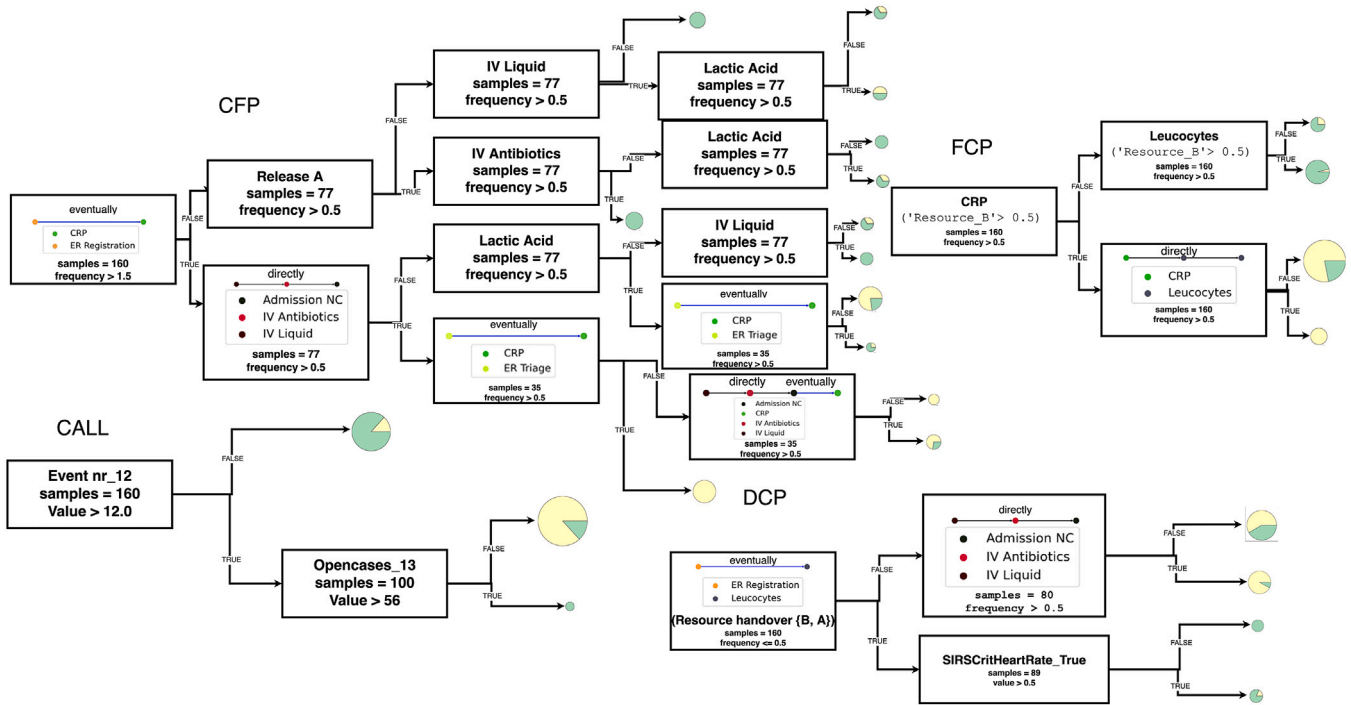


Fig. D.17. Four types of explanation for Trace of XJA.

**Table D.13**  
Fidelity results and number of patterns discovered for trace XJA.

Encoding	Local fidelity	# Patterns
FCP	94.29%	36
DCP	93.15%	38
CALL	83.50%	25
CFP	87.48%	25

pattern where ER Registration is eventually followed by Leucocytes with a handover between the two events. This condition is not satisfied in the inquiry trace, leading to the next decision node. The next node is a control-flow pattern, stating that IV Liquid is directly followed by IV Antibiotics and then Admission NC, which is present in the inquiry trace, explaining the reason for a negative outcome prediction. To flip the outcome, satisfying the condition in the first node is crucial. If the aforementioned context-aware pattern was absent from the trace, the outcome could be flipped regardless of the presence or absence of a critically high heart rate alert ('SIRSCritHeartRate\_True').

Meanwhile, for FCP, the first (most predictive) node of the DT is the frequency of the CRP activity, under the condition that activity is performed by the 'Resource group B', which the inquiry trace satisfies. The next patterns suggests that a CRP measurement should be directly

followed by two further Leucocytes measurements, which is also satisfied towards the end of the prefix trace, explaining the reasons for the negative outcome of the trace. To increase the likelihood of flipping the outcome, there should be one or no occurrence of CRP (there are 3 in the inquiry prefix trace), with the node on top of the first one leading to a high likelihood of flipping the outcome even if the condition of having at least an occurrence of Leucocytes executed by 'Resource B' is not present.

When looking at the baseline methods, the obtained explanation from the CALL relies on two event attributes: 'Event\_nr' being higher than 6 at position 12. This is caused by the DT splitting criterion, as it possibly suggests that longer traces, along with having 'opencases' lower than 93 at position 13, will lead to a negative outcome, as is the case for the inquiry trace XJA. Such an explanation lacks a control flow perspective and leaves the user uncertain about the right action. Conversely, the explanation obtained from CFP is quite complex as it generates a larger DT: the most important node is CRP being eventually preceded by ER Registration, which should occur at least 2 times in the trace to lead to a negative outcome. This pattern suggests that multiple CRP measurements lead to a negative outcome in the trace, as the other predominantly negative paths all focus on having multiple occurrences of measurement activities (CRP, LacticAcid, Leucocytes).

For the positive nodes, we can see a focus on the most common release activity Release A, along with administrating either IV Liquids or IV Antibiotics to the patient may lead to a higher likelihood of flipping the outcome.

This analysis demonstrates that context-aware explanations not only provide higher fidelity but also enable actionable insights by integrating data and control flow perspectives.

## Appendix E. Abbreviations list

AUC	Area Under the Curve
CA-IMPRESSED	Context-Aware IMPRESSED
CALL	Concatenate All
CFP	Control-Flow Patterns
DAG	Directed Acyclic Graph
DCP	Dynamic Context Patterns
DT	Decision Tree
FCP	Full Context Patterns
GA	Genetic Algorithm
IMPRESSED	Interactive Multi-Interested Process Pattern Discovery
LF	Local Fidelity
LIME	Local Interpretable Model-Agnostic Explanations
LPMs	Local Process Models
LSTM	Long Short-Term Memory
PABLO	PATtern Based LOcal explanation
PPM	Predictive Process Monitoring
RFE-CV	Recursive Feature Elimination with Cross-validation
TPE	Tree Parzen Estimator
TWP	Time Within Pattern
XAI	eXplainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting
XPPM	eXplainable Predictive Process Monitoring

## Data availability

We have provided a link to access the data and code.

## References

- Adamo, Greta, Borgo, Stefano, Di Francescomarino, Chiara, Ghidini, Chiara, Guarino, Nicola, Sanfilippo, Emilio M., 2017. Business processes and their participants: an ontological perspective. In: *AI\* IA 2017 Advances in Artificial Intelligence: XVIth International Conference of the Italian Association for Artificial Intelligence*, Bari, Italy, November 14-17, 2017, Proceedings 16. Springer, pp. 215–228.
- Augusto, Adriano, Conforti, Raffaele, Dumas, Marlon, La Rosa, Marcello, Maggi, Fabrizio Maria, Marrella, Andrea, Mecella, Massimo, Soo, Allar, 2018. Automated discovery of process models from event logs: Review and benchmark. *IEEE Trans. Knowl. Data Eng.* 31 (4), 686–705.
- Aversano, Lerina, Bernardi, Mario Luca, Cimitile, Marta, Iammarino, Martina, Verdone, Chiara, 2023. A data-aware explainable deep learning approach for next activity prediction. *Eng. Appl. Artif. Intell.* 126, 106758.
- Bradley, Andrew P., 1997. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognit.* 30 (7), 1145–1159.
- Buliga, Andrei, Vazifehdoostirani, Mozghan, Genga, Laura, Lu, Xixi, Dijkman, Remco, Chiara, Di Francescomarino, Ghidini, Chiara, Reijers, Hajo, 2024. Uncovering patterns for local explanations in outcome-based predictive process monitoring. In: *International Conference on Business Process Management*. Springer, pp. 363–380.
- Camargo, Manuel, Dumas, Marlon, González-Rojas, Oscar, 2019a. Learning accurate LSTM models of business processes. In: *Hildebrandt, Thomas, van Dongen, Boudewijn F., Röglinger, Maximilian, Mendling, Jan (Eds.), Business Process Management*. Springer International Publishing, Cham, pp. 286–302.
- Camargo, Manuel, Dumas, Marlon, Rojas, Oscar González, 2019b. Learning accurate LSTM models of business processes. In: *Hildebrandt, Thomas T., van Dongen, Boudewijn F., Röglinger, Maximilian, Mendling, Jan (Eds.), Business Process Management - 17th International Conference, BPM 2019, Vienna, Austria, September 1-6, 2019, Proceedings*. In: *Lecture Notes in Computer Science*, vol. 11675, Springer, pp. 286–302.
- Ceravolo, Paolo, Comuzzi, Marco, De Weerd, Jochen, Di Francescomarino, Chiara, Maggi, Fabrizio Maria, 2024. Predictive process monitoring: concepts, challenges, and future research directions. *Process. Sci.* 1 (1), 1–22.
- Cheng, Qiang, Zhou, Hongbo, Cheng, Jie, Li, Huiqing, 2014. A minimax framework for classification with applications to images and high dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.* 36 (11), 2117–2130.
- Di Francescomarino, Chiara, Ghidini, Chiara, 2022. Predictive process monitoring. In: *Process Mining Handbook*, Springer International Publishing, Cham, pp. 320–346.
- Di Francescomarino, Chiara, Ghidini, Chiara, Maggi, Fabrizio Maria, Petrucci, Giulio, Yeshchenko, Anton, 2017. An eye into the future: Leveraging A-priori knowledge in predictive business process monitoring. In: *Carmona, Josep, Engels, Gregor, Kumar, Akhil (Eds.), Business Process Management*. Springer International Publishing, Cham, pp. 252–268.
- Diamantini, Claudia, Genga, Laura, Potena, Domenico, van der Aalst, Wil, 2016. Building instance graphs for highly variable processes. *Expert Syst. Appl.* 59, 101–118.
- Dogan, Onur, 2023. A process-centric performance management in a call center. *Appl. Intell.* 53 (3), 3304–3317.
- Fang, Wei, Zhang, Qiang, Sun, Jun, Wu, Xiaojun, 2020. Mining high quality patterns using multi-objective evolutionary algorithm. *IEEE Trans. Knowl. Data Eng.* 34 (8), 3883–3898.
- Fani Sani, Mohammadreza, Vazifehdoostirani, Mozghan, Park, Gyunan, Pegoraro, Marco, van Zelst, Sebastiaan J., van der Aalst, Wil M.P., 2023. Performance-preserving event log sampling for predictive monitoring. *J. Intell. Inf. Syst.* 61 (1), 53–82.
- Freytes, Christian Yaphet, Perry Mayrand, Robin, Sawada, Luana Okino, Yan Liang, Thony, Curiel Cid, Rosie E., Burke, Shanna, Loewenstein, David, Duara, Ranjan, Adjouadi, Malek, 2023. Recursive feature elimination with cross validation for alzheimer's disease classification using cognitive exam scores. In: *2023 Intelligent Methods, Systems, and Applications*. IMSA, pp. 327–332.
- Galanti, Riccardo, de Leoni, Massimiliano, Monaro, Merylin, Navarin, Nicolò, Marazzi, Alan, Di Stasi, Brigida, Maldera, Stéphanie, 2023. An explainable decision support system for predictive process analytics. *Eng. Appl. Artif. Intell.* 120, 105904.
- Grinvald, Avihai, Soffer, Pnina, Mokryn, Osnat, 2021. Inter-case properties and process variant considerations in time prediction: A conceptual framework. In: *Augusto, Adriano, Gill, Nurcan, Selmin, Reinhartz-Berger, Iris, Schmidt, Rainer, Zdravkovic, Jelena (Eds.), Enterprise, Business-Process and Information Systems Modeling*. Springer International Publishing, Cham, pp. 96–111.
- Guidotti, Riccardo, Monreale, Anna, Giannotti, Fosca, Pedreschi, Dino, Ruggieri, Salvatore, Turini, Franco, 2019. Factual and counterfactual explanations for black box decision making. *IEEE Intell. Syst.* 34 (6), 14–23.
- Harl, Maximilian, Weinzierl, Sven, Stierle, Mathias, Matzner, Martin, 2020. Explainable predictive business process monitoring using gated graph neural networks. *J. Decis. Syst.* 29 (suppl. 1), 312–327.
- Hsieh, Chihcheng, Moreira, Catarina, Ouyang, Chun, 2021. DiCE4EL: Interpreting process predictions using a milestone-aware counterfactual approach. In: *2021 3rd International Conference on Process Mining, ICPM*, pp. 88–95.
- Huang, Tsung-Hao, Metzger, Andreas, Pohl, Klaus, 2021. Counterfactual explanations for predictive business process monitoring. In: *Di Francescomarino, Chiara, Burattin, Andrea, Janiesch, Christian, Sadiq, Shazia (Eds.), In: EMCIS 2021, Proc.*, vol. 437, Cham, pp. 399–413.
- Hundogan, Olusanmi, Lu, Xixi, Du, Yupei, Reijers, Hajo A., 2023. CREATED: Generating viable counterfactual sequences for predictive process analytics. In: *International Conference on Advanced Information Systems Engineering*. Springer, pp. 541–557.
- Kim, Jongchan, Comuzzi, Marco, Dumas, Marlon, Maggi, Fabrizio Maria, Teinema, Irene, 2022. Encoding resource experience for predictive process monitoring. *Decis. Support Syst.* 153, 113669.
- Klijn, Eva L., Mannhardt, Felix, Fahland, Dirk, 2024. Multi-perspective concept drift detection: Including the actor perspective. In: *Guizzardi, Giancarlo, Santoro, Flávia Maria, Mouratidis, Haralambos, Soffer, Pnina (Eds.), Advanced Information Systems Engineering - 36th International Conference, CAISE 2024, Limassol, Cyprus, June 3-7, 2024, Proceedings*. In: *Lecture Notes in Computer Science*, vol. 14663, Springer, pp. 141–157.
- Leemans, Sander J.J., van Zelst, Sebastiaan J., Lu, Xixi, 2023. Partial-order-based process mining: a survey and outlook. *Knowl. Inf. Syst.* 65 (1), 1–29.
- Leontjeva, Anna, Conforti, Raffaele, Di Francescomarino, Chiara, Dumas, Marlon, Maggi, Fabrizio Maria, 2015. Complex symbolic sequence encodings for predictive monitoring of business processes. In: *Motahari-Nezhad, Hamid Reza, Recker, Jan, Weidlich, Matthias (Eds.), Business Process Management - 13th International Conference, BPM 2015, Innsbruck, Austria, August 31 - September 3, 2015, Proceedings*. In: *Lecture Notes in Computer Science*, vol. 9253, Springer, pp. 297–313.
- Liu, Ying, Stein Dani, Vinicius, Beerepoot, Iris, Lu, Xixi, 2023. Turning logs into lumber: preprocessing tasks in process mining. In: *International Conference on Process Mining*. Springer, pp. 98–109.
- Lu, Xixi, Fahland, Dirk, van der Aalst, Wil M.P., 2015. Conformance checking based on partially ordered event data. In: *Business Process Management Workshops: BPM 2014 International Workshops*. pp. 75–88.

- Maggi, Fabrizio Maria, Dumas, Marlon, García-Bañuelos, Luciano, Montali, Marco, 2013. Discovering data-aware declarative process models from event logs. In: *Business Process Management: 11th International Conference, BPM 2013, Beijing, China, August 26-30, 2013. Proceedings.* Springer, pp. 81–96.
- Mannhardt, Felix, 2018. Multi-perspective process mining. In: van der Aalst, Wil M.P., Casati, Fabio, Conforti, Raffaele, de Leoni, Massimiliano, Dumas, Marlon, Kumar, Akhil, Mendling, Jan, Nepal, Surya, Pentland, Brian T., Weber, Barbara (Eds.), *Proceedings of the Dissertation Award, Demonstration, and Industrial Track At BPM 2018 Co-Located with 16th International Conference on Business Process Management (BPM 2018), Sydney, Australia, September 9-14, 2018.* In: *CEUR Workshop Proceedings*, vol. 2196, CEUR-WS.org, pp. 41–45.
- Márquez-Chamorro, Alfonso Eduardo, Resinas, Manuel, Ruiz-Cortés, Antonio, 2017. Predictive monitoring of business processes: a survey. *IEEE Trans. Serv. Comput.* 11 (6), 962–977.
- Mehdiyev, Nijat, Fettek, Peter, 2021. Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring. *Interpret. Artif. Intell.: A Perspect. Granul. Comput.* 1–28.
- Mothilal, Ramaravind K., Sharma, Amit, Tan, Chenhao, 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In: *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency.* pp. 607–617.
- Munoz-Gama, Jorge, Martin, Niels, Fernandez-Llatas, Carlos, Johnson, Owen A., Sepúlveda, Marcos, Helm, Emmanuel, Galvez-Yanjari, Victor, Rojas, Eric, Martinez-Millana, Antonio, Aloini, Davide, et al., 2022. Process mining for healthcare: Characteristics and challenges. *J. Biomed. Inform.* 127, 103994.
- Navarin, Nicolò, Vincenzi, Beatrice, Polato, Mirko, Sperduti, Alessandro, 2017. LSTM networks for data-aware remaining time prediction of business process instances. In: *2017 IEEE Symposium Series on Computational Intelligence. SSCI*, pp. 1–7.
- Neu, Dominic A., Lahann, Johannes, Fettek, Peter, 2022. A systematic literature review on state-of-the-art deep learning methods for process prediction. *Artif. Intell. Rev.* 55 (2), 801–827.
- Nguyen, Hoang, Dumas, Marlon, La Rosa, Marcello, Maggi, Fabrizio Maria, Suriadi, Suriadi, 2014. Mining business process deviance: a quest for accuracy. In: *OTM Confederated International Conferences “on the Move To Meaningful Internet Systems”*. pp. 436–445.
- Panpanich, Patcharin, Porouhan, Parham, Premchaiswadi, Wichian, 2015. Analysis of handover of work in call center using social network process mining technique. In: *2015 13th International Conference on ICT and Knowledge Engineering. ICT & Knowledge Engineering 2015*, pp. 97–104.
- Pasquadibisceglie, Vincenzo, Appice, Annalisa, Castellano, Giovanna, Malerba, Donato, 2023. Darwin: An online deep learning approach to handle concept drifts in predictive process monitoring. *Eng. Appl. Artif. Intell.* 123, 106461.
- Peeperkorn, Jari, Ortega Vázquez, Carlos, Stevens, Alexander, De Smedt, Johannes, vanden Broucke, Seppe, De Weerd, Jochen, 2023. Outcome-oriented predictive process monitoring on positive and unlabelled event logs. In: Montali, Marco, Senderovich, Arik, Weidlich, Matthias (Eds.), *Process Mining Workshops.* Springer Nature Switzerland, Cham, pp. 255–268.
- Peeva, Viki, Mannel, Lisa L., van der Aalst, Wil M.P., 2022. From place nets to local process models. In: *International Conference on Applications and Theory of Petri Nets and Concurrency.* Springer, pp. 346–368.
- Pesic, Maja, Schonenberg, Helen, Van der Aalst, Wil M.P., 2007. Declare: Full support for loosely-structured processes. In: *11th IEEE International Enterprise Distributed Object Computing Conference. EDOC 2007, IEEE*, 287–287.
- Rama-Maneiro, Efrén, Vidal, Juan C., Lama, Manuel, 2021. Deep learning for predictive business process monitoring: Review and benchmark. *IEEE Trans. Serv. Comput.* 16 (1), 739–756.
- Rizzi, Williams, Di Francescomarino, Chiara, Maggi, Fabrizio Maria, 2020a. Explainability in predictive process monitoring: when understanding helps improving. In: *International Conference on Business Process Management.* Springer, pp. 141–158.
- Rizzi, Williams, Di Francescomarino, Chiara, Maggi, Fabrizio Maria, 2020b. Explainability in predictive process monitoring: When understanding helps improving. In: *Lecture Notes in Business Information Processing*, vol. 392 LNBI, Springer International Publishing, pp. 141–158.
- Selbst, Andrew, Powles, Julia, 2018. Meaningful information and the right to explanation. In: *Conference on Fairness, Accountability and Transparency.* PMLR, 48–48.
- Senderovich, Arik, Di Francescomarino, Chiara, Ghidini, Chiara, Jorbina, Kerwin, Maggi, Fabrizio Maria, 2017. Intra and inter-case features in predictive process monitoring: A tale of two dimensions. In: Carmona, Josep, Engels, Gregor, Kumar, Akhil (Eds.), *Business Process Management.* Springer International Publishing, Cham, pp. 306–323.
- Senderovich, Arik, Di Francescomarino, Chiara, Maggi, Fabrizio Maria, 2019. From knowledge-driven to data-driven inter-case feature encoding in predictive process monitoring. *Inf. Syst.* 84, 255–264.
- Setzu, Mattia, Guidotti, Riccardo, Monreale, Anna, Turini, Franco, Pedreschi, Dino, Giannotti, Fosca, 2021. Glocalx-from local to global explanations of black box ai models. *Artificial Intelligence* 294, 103457.
- Slaats, Tijs, 2020. Declarative and hybrid process discovery: Recent advances and open challenges. *J. Data Semant.* 9 (1), 3–20.
- Sokol, Kacper, Flach, Peter, 2024. Interpretable representations in explainable AI: from theory to practice. *Data Min. Knowl. Discov.* 1–39.
- Stierle, Matthias, Brunk, Jens, Weinzierl, Sven, Zilker, Sandra, Matzner, Martin, Becker, Jörg, 2021. Bringing light into the darkness - a systematic literature review on explainable predictive business process monitoring techniques. In: *ECIS Proceedings.*
- Tax, Niek, Dalmas, Benjamin, Sidorova, Natalia, van der Aalst, Wil M.P., Norre, Sylvie, 2018. Interest-driven discovery of local process models. *Inf. Syst.* 77, 105–117.
- Tax, Niek, Sidorova, Natalia, Haakma, Reinder, van der Aalst, Wil M.P., 2016. Mining local process models. *J. Innov. Digit. Ecosyst.* 3 (2).
- Tax, Niek, Verenich, Ilya, La Rosa, Marcello, Dumas, Marlon, 2017. Predictive business process monitoring with LSTM neural networks. In: Dubois, Eric, Pohl, Klaus (Eds.), *Advanced Information Systems Engineering.* Springer International Publishing, Cham, pp. 477–492.
- Teinema, Irene, Dumas, Marlon, Rosa, Marcello La, Maggi, Fabrizio Maria, 2019. Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. Knowl. Discov. Data* 13 (2), 1–57.
- Vazifheoostirani, Mozghan, Abbaspour Onari, Mohsen, Grau, Isel, Genga, Laura, Dijkman, Remco, 2023a. Uncovering the hidden significance of activities location in predictive process monitoring. In: *International Conference on Process Mining.* Springer, pp. 191–203.
- Vazifheoostirani, Mozghan, Genga, Laura, Dijkman, Remco, 2022. Encoding high-level control-flow construct information for process outcome prediction. In: *2022 4th International Conference on Process Mining. ICPM, IEEE*, pp. 48–55.
- Vazifheoostirani, Mozghan, Genga, Laura, Lu, Xixi, Verhoeven, Rob, van Laarhoven, Hanneke, Dijkman, Remco, 2023b. Interactive multi-interest process pattern discovery. In: *International Conference on Business Process Management.* Springer, pp. 303–319.
- Wickramanayake, Bemali, He, Zhipeng, Ouyang, Chun, Moreira, Catarina, Xu, Yue, Sindhgatta, Renuka, 2022. Building interpretable models for business process prediction using shared and specialised attention mechanisms. *Knowl.-Based Syst.* 248, 108773.
- Wickramanayake, Bemali, Ouyang, Chun, Xu, Yue, Moreira, Catarina, 2023. Generating multi-level explanations for process outcome predictions. *Eng. Appl. Artif. Intell.* 125, 106678.