Discussion: Uncovering Patterns for Local Explanations in Outcome-based Predictive Process Monitoring

Andrei Buliga[1,2,*], Mozhgan Vazifehdoostirani[3], Laura Genga[3], Xixi Lu[5], Chiara Di Francescomarino[4] and Chiara Ghidini[2]

[1] *Fondazione Bruno Kessler, Trento, Italy*
[2] *Free University of Bozen-Bolzen, Bolzano, Italy*
[3] *Eindhoven University of Technology Eindhoven, Netherlands*
[4] *University of Trento, Trento,Italy*
[5] *University of Utrecht, Utrecht,Netherlands*

Abstract

Explainable Predictive Process Monitoring aims at deriving explanations of the inner workings of black-box classifiers used to predict the continuation of ongoing process executions. Most existing techniques use data attributes (e.g., the loan amount) to explain the prediction outcomes. However, explanations based on *control flow patterns* (such as calling the customers first, and then validating the application, or providing early discounts) cannot be provided. This omission may result in many valuable, actionable explanations going undetected. To fill this gap, this paper proposes PABLO (PAttern Based LOcal Explanations), a framework that generates *local control-flow aware explanations* for a given predictive model. Given a process execution and its outcome prediction, PABLO discovers control-flow patterns from a set of alternative executions, which are used to deliver explanations that support or flip the prediction for the given process execution. Evaluation against real-life event logs shows that PABLO provides high-quality explanations of predictions in terms of *fidelity* and accurately explains the reasoning behind the predictions of the black box models.

Keywords

local explanations, process pattern, explainable AI

1. Introduction

Predictive Process Monitoring (PPM) focuses on predicting the future states of business processes, offering substantial value in areas such as process recommendations, early interventions, and resource planning. Several studies have demonstrated the efficacy of machine learning and deep learning techniques (such as XGBoost and LSTM) in making accurate outcome-based predictions [1].

Such techniques are considered black-box models, since their inner logic is uninterpretable to users [2]. Consequently, previous studies in PPM have employed post-hoc eXplainable Artificial Intelligence (XAI) techniques to provide explanations for these black box models. As a result, a novel subfield called eXplainable Predictive Process Monitoring (XPPM) has emerged, offering explanations tailored to different applications and user's requirements [3]. *Local explanations* may suit domain experts seeking an explanation into *why a prediction was made for a specific case (i.e., an "inquiry trace")*, while *global explanations* may suit process owners who *prioritize strategic decisions over individual assessments* [2].

The frameowrk [1] focuses on providing *local explanations* for process analysis. Local explanation methods are primarily classified in terms *factual* and *counterfactual* explanation methods. Factual explanations reveal the model's reasoning behind specific predictions. These methods highlight the most

EKAPI 2024: 1st International Workshop On Explainable Knowledge Aware Process IntelligencE, June 20-22, 2024, Roccella Jonica (RC), Italy

*Corresponding author.
†These authors contributed equally.
abuliga@fbk.eu (A. Buliga); m.vazifehdoostirani@tue.nl (M. Vazifehdoostirani); l.genga@tue.nl (L. Genga); x.lu@uu.nl (X. Lu); c.difrancescomarino@unitn.it (C. D. Francescomarino); chiara.ghidini@unibz.it (C. Ghidini)
0000-0002-8179-9146 (A. Buliga); 0000-0001-8521-4427 (M. Vazifehdoostirani); 0000-0001-8746-8826 (L. Genga); 0000-0001-9634-5852 (X. Lu); 0000-0002-0264-9394 (C. D. Francescomarino); 0000-0003-1563-4965 (C. Ghidini)

[1]This work is a discussion paper of a conference paper submitted at the Business Process Management (BPM) conference 2024.

influential features [4, 5, 6, 7] or derive factual rules from a surrogate model replicating the behaviour of the black box model in the neighbourhood of a specific case [2]. Counterfactual explanation methods, on the other hand, provide explanations that allow for altering the prediction with the minimum change in the inquiry trace [8]. Previous research pointed out that process analysts often seek insights into actionable modifications within a specific case to achieve desired predictions [9].

Different strategies can be employed for delivering explanations, with some methods outputting counterfactual traces without explaining required changes to achieve desired outcomes [10]. Others, like LORELEY [9], blend factual and counterfactual explanations to derive rules modeling the relation between process features and outcomes. However, these methods focus solely on static trace attributes, lacking *control-flow* aware explanations crucial for operational decisions and process redesign.

The proposed framework, PABLO (**PA**ttern **B**ased **LO**cal Explanations), generates pattern-based explanations for outcome-oriented PPM methods. The main idea of the framework is to uncover predictive process patterns around a single inquiry trace. These patterns can correspond to different activity sequences in different counterfactuals or factual examples, making them challenging to uncover.

The framework consists of four components: (1) the black-box model, (2) the neighbourhood generator, (3) the process pattern discovery, and (4) the surrogate glass-box modelling. We compare our framework against a baseline using 9 event logs obtained from real-life processes. We compute the quality of the generated neighbourhood traces and evaluate the framework's capabilities of approximating both local and global predictive accuracy. The evaluation results suggest that PABLO can more accurately approximate local predictions, while obtaining comparable results for the global predictions.

2. Background

In this section, we recall the basic concepts needed to introduce our framework.

Event, trace, event log Let $\mathcal{AC}$ denote the set of process activities, $\mathcal{C}$ denote the set of case identifiers, $\mathcal{T}$ represent the time domain, and $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_m$ represent sets of additional attributes where $m \in \mathbb{N}$. An event is represented as a tuple $e = (a, c, t, d_1, \ldots, dm)$, where $a \in \mathcal{AC}$, $c \in \mathcal{C}$, $t \in \mathcal{T}$, and $d_i \in \mathcal{D}_i$. A *trace*, $\sigma = \langle e_1, \cdots, e_n \rangle$, is a finite sequence of events where timestamps are non-decreasing. Let $\mathbb{S}$ be the set of all possible traces; an *event log* L is defined as $L = \{\sigma_1, \sigma_2, \cdots, \sigma_n\} \subseteq \mathbb{S}$.

Process pattern A process pattern $P = (N, \mapsto, \alpha, \rho)$ is a directed acyclic graph (DAG), where N is a set of nodes, $\mapsto$ is set of edges, where $\forall n, n' \in N, n \mapsto n'$ denotes there is an edge from n to n', α is a function that assigns a label $\alpha(n) \in \mathcal{AC}$ to any node $n \in N$, and ρ is a function that assigns a label to each edge representing the semantics of the relation between the source and the target node (e.g., directly or eventually follows).

Pattern instance Let P be a process pattern and σ a trace. We define Aligner(P, σ) to be a function that receives σ and P and returns a set of pattern instances PI_σ^P, where each pattern instance corresponds to a list of positions of trace events matching the constraints expressed by the pattern.

3. PABLO

The proposed framework **PABLO** aims to generate control-flow aware local explanations centered around a single inquiry trace. Figure 1 illustrates the four main components of the framework.

The first component, *Predictive Model Training*, takes as input an event log $\mathcal{L}$ and encodes the log to train a predictive black box model b_θ that needs to be explained. Next, the second component *Synthetic Neighbourhood Generation* takes as input the trained predictive model b_θ and the inquiry trace x and generates a set of alternative traces $\mathcal{L}_{syn}$ in the neighborhood of x, containing both traces associated with the same predicted outcome as the inquiry trace and traces associated with the flipped outcome. Using the generated neighbourhood, the third component *Pattern Discovery* discovers *process patterns* that strongly affect the predictions of b_θ. Using the discovered patterns as additional features, the last component *Surrogate Modeling* derives comprehensive explanations.

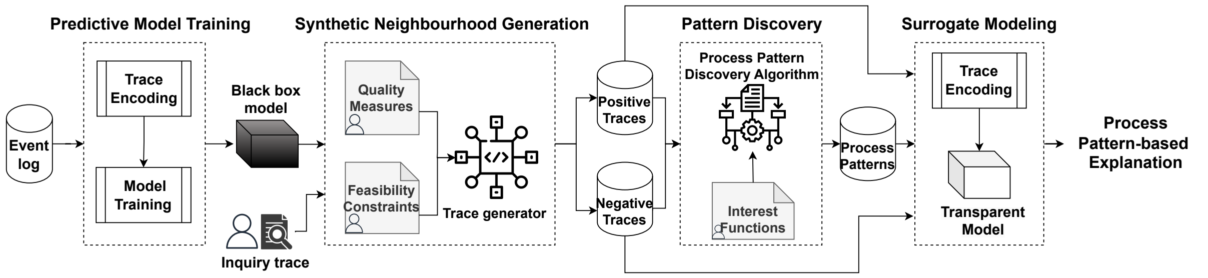

Figure 1: Pattern-based local explanation framework

We emphasize that PABLO's components are method-agnostic, as they can seamlessly accommodate any method chosen by the user. In the following, we discuss each step and show a concrete instantiation of the proposed framework by integrating various proposed techniques for PABLO's components.

Trace encoding and black-box model training Given an event log as input, training a predictive model usually requires preprocessing the log traces to make them compatible with PPM models, using various encoding strategies [1]. In this study, we instantiated two encodings: the simple-index and the simple trace index. Given a prefix trace σ_i, the *simple-index* (SI) encoding returns a vector $g_i =< a_{i_1}, ..., a_{i_2}, ..., a_{i_m} >$, where a_{i_j} denotes the activity i at position j. The *simple-trace index* (STI) encoding, instead, encodes data attributes (trace attributes) too, returning a feature vector $g_i =< s_i^1$, $\ldots, s_i^u, a_{i_1}, a_{i_2}, ..., a_{i_m} >$, with each s_i representing a static feature corresponding to a trace attribute. After encoding the event log, a train-validation-test split is performed, and a black-box predictive model, i.e., eXtreme Gradient Boost (XGBoost) is trained.

Synthetic neighbourhood generation: The second component takes as input the trained predictive model b_θ and the inquiry trace x. Let the prediction of x using the black-box model be $b_\theta(x) = y$, while $\overline{y}$ is the flipped outcome. This component produces a synthetic neighbourhood of traces that either result in $\overline{y}$ (labelled as *positive traces*) or lead to y (labelled as *negative traces*), given the trace x and b_θ. *Quality Measures* and *Feasibility Constraints* need to be defined based on the users' needs and preferences. For example, users might prefer traces that are as close as possible to the given inquiry trace while at the same time minimizing sparsity, i.e., the number of changes made from the inquiry trace to the synthetic trace. Formally, we define $\mathcal{L}_{syn} = \mathcal{L}_{syn}^{=} \cup \mathcal{L}_{syn}^{\neq}$ where traces $\sigma \in \mathcal{L}_{syn}^{=}$ have $b_\theta(\sigma) = b_\theta(x)$, and $\sigma \in \mathcal{L}_{syn}^{\neq}$ such that $b_\theta(\sigma) \neq b_\theta(x)$ as the output of the second component. For this component, we adapted a modification of the Genetic Algorithm (GA) implemented in DiCE [11].

Pattern Discovery The third component of the framework discovers process patterns from the generated synthetic neighbourhood that affect the prediction. These patterns represent a key element of the framework, as they allow to uncover hidden relations between the control flow and the predicted outcome. We deploy IMPresseD [12] as a concrete instantiation for this component. The IMPresseD algorithm iteratively uncovers process patterns, selecting only those dominating others in the Pareto front via the predefined interest functions, extending them by following rules defined in [12].

The IMPresseD algorithm initially converts each trace into a directed acyclic graph (DAG), where each node corresponds to an event and edges model a user-defined notion of order (e.g., the time interval between two events is higher than a threshold). Then, IMPresseD iteratively uncovers process patterns from each trace. It starts with discovering patterns of length-1 (i.e., single activities) and retains only those dominating others in the Pareto front via the predefined interest functions. Then, each non-dominated pattern will be extended to a larger pattern following extension rules defined in [12]. The algorithm stops the discovery procedure after completing predefined extension steps or as requested by the user and returns process patterns as a DAG. Additionally, it provides an encoded event log, recording the frequency of each pattern's occurrence in each trace.

Surrogate Modelling Using the discovered patterns as additional features, the last component *Surrogate Modelling* considers all features available to derive comprehensive explanations. Here, we used a Decision Tree (DT) as a glass-box surrogate for instantiating the fourth component. To include discovered patterns in rule-based explanations, one needs to encode patterns into an understandable format for DT. We encoded the patterns using a frequency-based method [12], measuring $|PI_{\sigma}^{P}|$ for each discovered pattern P and each instance $\sigma \in \mathcal{L}_{syn}$.

As final outputs, PABLO produces (1) comprehensive explanations which emphasize the important control-flow patterns that affect the outcome prediction complemented by (2) concrete alternative executions against the inquiry trace, retrieved from the synthetic neighbourhood. A real-world example of the outputs is shown in the evaluation results.

4. Evaluation

To evaluate the proposed framework, we have designed a *quantitative evaluation* followed by a *qualitative comparison* of PABLO against a baseline where no patterns are discovered. The objectives of the quantitative evaluation is to assess the faithfulness of the explanation generated by PABLO.

4.1. Datasets

We analyzed three event logs commonly used in literature, namely *BPIC12*, *BPIC17*, and *Sepsis*. Using the same labeling strategy used in [1], we obtained 9 datasets, as shown in Table 1.

Table 1
Event logs statistics

Dataset	log	trace#	variant#	event class#	trace att.#	avg. trace length	positive class %	prefix lengths	domain
BPIC12_1							47%		
BPIC12_2	BPIC12	4685	3790	36	1	35	17%	[15,20,25,30]	finance
BPIC12_3							35%		
BPIC17_1							41%		
BPIC17_2	BPIC17	31413	2087	36	3	35	12%	[20,25,30,35]	finance
BPIC17_3							47%		
Sepsis_1							14%		
Sepsis_2		782	709	15	24	14	14%	[5,9,13,16]	healthcare
Sepsis_3							14%		

4.2. Evaluation metrics

Regarding the evaluation objective, we evaluate the mimicking capabilities of the surrogate model $\mathbf{c}$ inferred by PABLO and of the returned explanations against the black box model $\mathbf{b}_\theta$, using the metrics:

- Local Fidelity (LF): $\frac{1}{|\mathcal{L}_{\text{syn}}|}\sum_{\sigma_i \in \mathcal{L}_{\text{syn}}} \mathbb{I}_{(\mathbf{c}(\sigma_i)=\mathbf{b}_\theta(\sigma_i))}$, compares $\mathbf{c}$ and $\mathbf{b}_\theta$ on $\mathcal{L}_{\text{syn}}$
- Global Fidelity (GF): $\frac{1}{|\mathcal{L}|}\sum_{\sigma_i \in \mathcal{L}} \mathbb{I}_{(\mathbf{c}(\sigma_i)=\mathbf{b}_\theta(\sigma_i))}$, compares $\mathbf{c}$ and $\mathbf{b}_\theta$ on $\mathcal{L}$

where $\mathbb{I}(\cdot)$ is the indicator function, returning 1 if the predictions of the derived surrogate model $\mathbf{c}$ and the black box model $\mathbf{b}_\theta$ on σ_i match, and 0 otherwise. $\mathcal{L}_{\text{synth}}$ represents a subset of the synthetic neighbourhood, while $\mathcal{L}$ is a subset of the real event log used to evaluate the black-box model.

5. Results

In this section, we present the results of the experiments carried out to address the research objectives introduced in Section 4. We compare the results of the PABLO framework with the baseline approach through the local and global fidelity metrics. Figure 2 displays the overall results for the two trace encodings used across the different prefix lengths.

By looking at Fig. 2, on average, across both encodings, for the local fidelity (LF), PABLO *always* outperforms the baseline with a range of improvement between 2% and 10%, as well as with an average

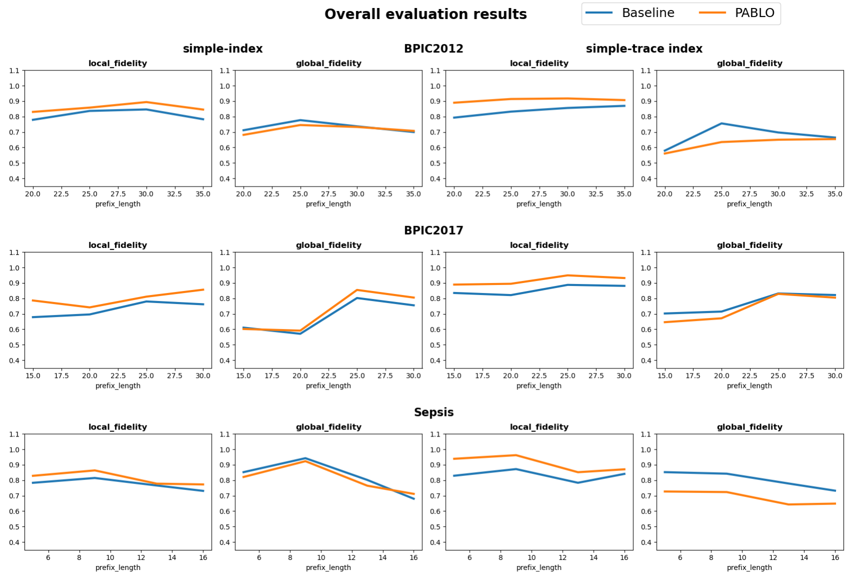

Figure 2: Local and global fidelity across different prefix lengths for both encodings

of 6%. For the SI encoding, we observe an average improvement of 5%, although for the BPIC17_2 and BPIC12_2, we see a much greater improvement of around 10%. For the STI encoding, we observe a greater improvement of around 7%. The largest differences are observed for the *Sepsis* datasets, with an average improvement of 9%. A notable improvement (10%) is also observed in the BPIC17_1 dataset.

For the global fidelity (GF), however, the baseline outperforms PABLO by an average of 3%, although some exceptions exist for both encodings. In the SI encoding, for the BPIC17_2 dataset, PABLO outperforms the baseline by 13%. Another exception is the BPIC12_2 dataset, where PABLO marginally outperforms the baseline. Overall, the difference between the baseline and PABLO are more pronounced for the STI encoding (6% on average), with the most significant differences being registered for the *Sepsis* datasets (8%), except Sepsis_1, where the difference is below the average (3.5%). Meanwhile, for the SI encoding the differences are much less pronounced (1.5% on average).

As depicted in Figure 2, the study investigates the influence of prefix lengths on evaluation metrics, presenting results for both Local Fidelity (LF) and Global Fidelity (GF) across various event logs ($BPIC12$, $BPIC17$, $Sepsis$). The aggregated outcomes for each event log are showcased, considering minimal discrepancies across log variants (Table 1). Regarding Local Fidelity, PABLO consistently outperforms the baseline across different prefix lengths, except for the $Sepsis$ event log with STI encoding at prefix length 13. The findings regarding Global Fidelity offer a more nuanced perspective, showcasing variations in performance between PABLO and the baseline. For instance, in the case of the SI encoding and $BPIC17$ event log, the baseline initially outperforms PABLO for shorter prefixes but is later surpassed by PABLO as prefix length increases. Similar trends are observed across the other event logs, with PABLO demonstrating superior performance for longer prefixes. Notably, this trend is more prominent for larger event logs like $BPIC17$, suggesting PABLO's proficiency in identifying predictive patterns for extended prefixes.

Overall, across the different datasets, prefix lengths, and encodings, PABLO consistently outperforms the baseline in terms of LF, capturing fine-grained process patterns within the synthetic neighbourhood that can more accurately explain the predictions of the black box model. Regarding the GF, in general, the baseline performs better with shorter prefixes, especially when considering trace attributes. However, PABLO's performance gets closer to (or higher than) the baseline with longer prefixes. Broadly, the current analysis lacks conclusive observations in that sense, and a more extensive evaluation is needed.

6. Conclusion

In this research, we introduced the PABLO framework, designed to provide local and control-flow aware explanations for outcome-based PPM. The designed framework addresses the gap in providing meaningful explanations on causes that have led to a specific prediction for a given process execution based on control-flow patterns. Real-life event log evaluation demonstrates PABLO's superiority over the baseline in replicating predictions from the black box model.In future work, we intend to extend the framework for providing global explanations by identifying patterns not only for an inquiry trace but also by giving insights to process owners to make strategic changes to the whole process to achieve more desirable outcomes. Furthermore, we intend to extend our evaluation by involving expert users to assess the trustworthiness of the provided control-flow-aware explanations.

References

[1] I. Teinemaa, M. Dumas, M. L. Rosa, F. M. Maggi, Outcome-oriented predictive process monitoring: Review and benchmark, ACM Transactions on Knowledge Discovery from Data 13 (2019) 1–57.

[2] N. Mehdiyev, P. Fettke, Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring, Interpretable artificial intelligence: A perspective of granular computing (2021) 1–28.

[3] G. Elkhawaga, M. Abu-Elkheir, M. Reichert, Explainability of predictive process monitoring results: Can you see my data issues?, Applied Sciences 12 (2022).

[4] W. Rizzi, C. Di Francescomarino, F. M. Maggi, Explainability in predictive process monitoring: when understanding helps improving, in: International Conference on Business Process Management, Springer, 2020, pp. 141–158.

[5] B. Wickramanayake, C. Ouyang, Y. Xu, C. Moreira, Generating multi-level explanations for process outcome predictions, Engineering Applications of Artificial Intelligence 125 (2023) 106678.

[6] M. Harl, S. Weinzierl, M. Stierle, M. Matzner, Explainable predictive business process monitoring using gated graph neural networks, Journal of Decision Systems 29 (2020) 312–327.

[7] M. Vazifehdoostirani, M. A. Onari, I. Grau, L. Genga, R. Dijkman, Uncovering the hidden significance of activities location in predictive process monitoring, in: Pre-proceedings of the ML4PM 2023 4th international workshop on leveraging machine learning in process mining, 2023, pp. 1–12.

[8] R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, F. Turini, Factual and counterfactual explanations for black box decision making, IEEE Intelligent Systems 34 (2019) 14–23.

[9] T. Huang, A. Metzger, K. Pohl, Counterfactual explanations for predictive business process monitoring, in: EMCIS 2021, Proc., volume 437, 2021, pp. 399–413.

[10] O. Hundogan, X. Lu, Y. Du, H. A. Reijers, Created: Generating viable counterfactual sequences for predictive process analytics, in: International Conference on Advanced Information Systems Engineering, Springer, 2023, pp. 541–557.

[11] R. K. Mothilal, A. Sharma, C. Tan, Explaining machine learning classifiers through diverse counterfactual explanations, in: Proceedings of the 2020 conference on fairness, accountability, and transparency, 2020, pp. 607–617.

[12] M. Vazifehdoostirani, L. Genga, X. Lu, R. Verhoeven, H. van Laarhoven, R. Dijkman, Interactive multi-interest process pattern discovery, in: International Conference on Business Process Management, Springer, 2023, pp. 303–319.