# Towards a Business-Oriented Approach to Visualization-Supported Interpretability of Prediction Results in Process Mining

Ana Rocío Cárdenas Maita[1] [a], Marcelo Fantinato[1] [b], Sarajane Marques Peres[1] [c]
and Fabrizio Maria Maggi[2] [d]

[1] *School of Arts, Science and Humanities, University of Sao Paulo, Rua Arlindo Bettio 1000, Sao Paulo, Brazil*
[2] *Faculty of Computer Science, Free University of Bozen-Bolzano, Bozen-Bolzan, Italy*

Keywords: Process Mining, Event Logs, Explainable Machine Learning, XAI, Interpretable Machine Learning, Predictive Process Mining.

Abstract: The majority of the state-of-the-art predictive process monitoring approaches are based on machine learning techniques. However, many machine learning techniques do not inherently provide explanations to business process analysts to interpret the results of the predictions provided about the outcome of a process case and to understand the rationale behind such predictions. In this paper, we introduce a business-oriented approach to visually support the interpretability of the results in predictive process monitoring. We take as input the results produced by the SP-LIME interpreter and we project them onto a process model. The resulting enriched model shows which features contribute to what degree to the predicted result. We exemplify the proposed approach by visually interpreting the results of a classifier to predict the output of a claim management process, whose claims can be accepted or rejected.



## 1 INTRODUCTION

Operational dashboards in business process management (BPM) are traditionally used to monitor the performance of ongoing or recently completed process cases (or instances). Business process monitoring dashboards allow process domain experts to intervene to fix or redirect the running instance or plan future interventions to optimize or improve the process (Dumas et al., 2018). Process experts need to make decisions during process monitoring aiming at optimizing the outcome of running cases or achieving a more general business goal. As a result, these decisions may significantly affect the outcome of cases (Aalst, 2016). In this context, the status of running processes cases or statistical performance information, usually available in current BPM systems, may not be sufficient to support decision making.

For this reason, recent research in process mining (Aalst, 2016) has sought to apply machine learning techniques to predict, from historical process data, the evolution of running process cases, thus supporting process analysts in decision making during process monitoring (Márquez-Chamorro et al., 2017; Mehdiyev and Fettke, 2021). These machine learning techniques can be used to predict, e.g., the positive or negative outcome of a case, the time remaining to complete a case, the next activity to be performed, or the resources to be used to perform an activity (Kim et al., 2022; Verenich et al., 2019b; Polato et al., 2018; Teinemaa et al., 2019; Maggi et al., 2014; Mehdiyev and Fettke, 2021). However, as in other areas of machine learning application, many of the techniques used do not inherently provide conditions for business analysts to interpret the prediction results in a way to understand the reasons for the predictions made (Belle and Papantonis, 2021; Holzinger, 2018; Márquez-Chamorro et al., 2017).

In fact, many machine learning techniques that solve complex problems provide opaque decision models (Barredo Arrieta et al., 2020), whose decision strategy is encoded in complex nonlinear functions associated with a large parametric space. Such predictive models are commonly applied as black box models. In a predictive model used as a black box, users do not understand its internal mechanisms and cannot extract knowledge about the decision process by look-

[a] https://orcid.org/0000-0001-9879-3229
[b] https://orcid.org/0000-0001-6261-1497
[c] https://orcid.org/0000-0003-3551-6480
[d] https://orcid.org/0000-0002-9089-6896



Maita, A., Fantinato, M., Peres, S. and Maggi, F.
Towards a Business-Oriented Approach to Visualization-Supported Interpretability of Prediction Results in Process Mining.
DOI: 10.5220/0011976000003467
In *Proceedings of the 25th International Conference on Enterprise Information Systems (ICEIS 2023) - Volume 1*, pages 395-406
ISBN: 978-989-758-648-4; ISSN: 2184-4992

ing only at its input parameters and outcomes (Belle and Papantonis, 2021; Ribeiro et al., 2016). However, in many real-world scenarios, e.g., healthcare and finance, explanations of why a model gives certain predictions are sorely needed, while in other domains they would be at least useful (Holzinger, 2018).

As for predictive process monitoring (PPM), knowing, e.g., which activities of a running process case determine whether its final outcome will be positive or negative may be crucial to apply corrective actions to that case or to ensure that such events do not reoccur. Thus, having information about why a machine learning technique is predicting a certain outcome for a process case may be more beneficial than just getting high-accuracy predictions. However, few studies on PPM have addressed interpretability (or explainability) for prediction results aided by machine learning techniques. Some recent studies on this topic are presented by Warmuth and Leopold (2022); Wickramanayake et al. (2022); Rizzi et al. (2020); Galanti et al. (2020) and Weinzierl et al. (2020).

LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016) is one of the best-known techniques for interpreting machine learning results. Despite being easily implemented and widely used, we claim LIME is primarily designed for data scientists. As a result, LIME produces and presents interpretability data in a way that does not make it easy for business experts to understand that data and hence to interpret the prediction results. Process domain experts would hardly be able to directly understand LIME results to then interpret prediction results produced by machine learning-supported PPM.

For example, consider the illustrative process model shown in Figure 1. Assume that a predictor was created to predict the outcome of a new process instance, whose output can be positive or negative, considering the path the instance takes through the process, i.e., the instance trace. LIME could be run to explain why the prediction results for certain instances are being performed considering the activities that were performed by the process instance under analysis. The result produced by LIME would seems like the one shown in Figure 2. This standard output of the LIME method is generic for any type of predictor, for any application domain. Note that the information presented in Figure 2, especially the way it is presented on the Y axis, is not easy, straightforward to interpret for a process domain expert. For example, per Figure 2, the non-execution of activity *E* contributes positively to the prediction of the positive result for that instance, while the execution of activity *B* contributes positively. Ideally, this information should be presented in a specific way for that application domain, i.e., on the business process model itself.

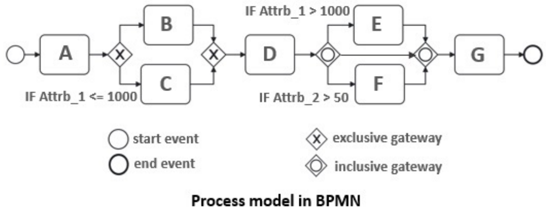


Figure 1: Illustrative example of a business process model.

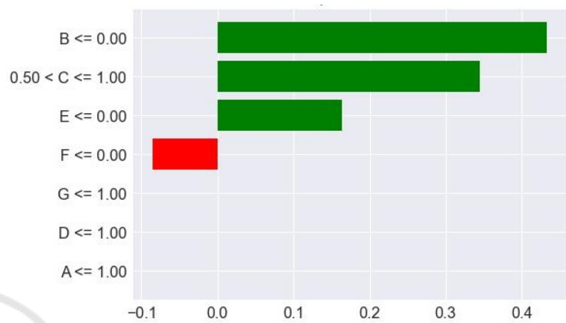


Figure 2: Illustrative example of LIME result for a positive-case outcome.

In this work, we introduce VisInter4PPM – a business-oriented approach to visually support the interpretability of PPM results. VisInter4PPM relies on the SP-LIME (Ribeiro et al., 2016), which is derived from LIME. We propose to graphically represent, through the activities in the process model, which features contribute to what degree to a predicted result. This graphic representation must be based on the results produced by SP-LIME. Data to support the interpretation of prediction results can be viewed per case, using the LIME outcome directly, or globally, building a combined interpretation of multiple SP-LIME outcomes through post-processing. We exemplify this approach by visually interpreting the results of a classifier to predict the outcome of a claim management process, whose claims can be accepted or rejected. To the best of our knowledge, our work provides a novel manner to view interpretability data in machine learning-supported PPM by combining the post-processing of the outcome of an interpretability method with data expressed graphically in a process model.

The remainder of this paper is organized as follows: Section 2 presents the theoretical background. Section 3 details our proposed approach, whereas Section 4 reports the conducted experiment. Section 5 discusses related work. Finally, Section 6 concludes the paper and spells out directions for future work.

## 2 THEORETICAL BACKGROUND

We introduce here our theoretical background, which includes an overview of process mining and PPM, and the concept of interpretable predictive models.

### 2.1 Process Mining and PPM

Process mining (Aalst, 2016) is a bridge between data science and process science, which, among other things, enables organizations to effectively use information on process executions from event logs to monitor and optimize processes across the BPM lifecycle.

The different solvable tasks in process mining consider an event log from one or more of four perspectives (Aalst, 2016): *(1)* control flow, which assumes as a source of information the process logic represented by the trace associated with each case; *(2)* performance, which allows discovering knowledge regarding the execution time of activities and cases; *(3)* resource, which provides an organizational and sociometric analysis that relates resources and how they are distributed within the work logic associated with the process; and *(4)* case, which considers the properties of the case providing contextualized analysis on the business underlying the process under analysis.

As summarized by de Sousa et al. (2021), following Aalst (2016)'s definitions, process mining work relies primarily on the concepts of event, case, trace, log, and attribute. An *event e* is the occurrence of a business process activity at a given time, performed by a given resource, at a given cost. A *case c* corresponds to a process instance and comprises events such that each event relates exactly to a case. A *trace* $\varsigma$ is a mandatory *attribute* of a case and corresponds to a finite sequence of events such that each event appears only once. An *event log L* is a set of cases such that each event appears only once in the entire event log. Each event in the event log comprises a set of attributes such as identifier, timestamp, activity, resource, and cost. Cases can also have non-mandatory *attributes*, often related to domain-specific data.

This work is particularly interested in the analysis from a flow-of-control perspective. For this perspective, the notions of simple trace and simple event log can be used. According to Aalst (2016), a simple trace $\sigma$ is a finite sequence of activity names $A$, i.e. $\sigma \in A^*$, and a simple event log $l$ is a multi-set of simple traces over $A^*$. Thus, an event log $L$, as illustrated in Figure 3, is represented as a simple event log $l = [\langle \text{A, B, D, E, G} \rangle^{70}, \langle \text{A, B, D, E, F, G} \rangle^{201}, \langle \text{A, B, D, F, E, G} \rangle^{50}, \langle \text{A, C, D, G} \rangle^{47}, \langle \text{A, C, D, F, G} \rangle^{132}]$.

Predictive process monitoring (PPM) aims to predict the behavior, performance, and results of business process at runtime. PPM triggers alerts on the execution of running cases, provides early advice so users can guide ongoing process executions towards achieving business goals (Maggi et al., 2014). There are two most common types of PPM, depending on the type of target variable: regression problems (such as estimating the time to complete a case) for continuous variables, and classification problems (such as predicting the next event or the case outcome) for discrete variables (Mehdiyev et al., 2020). PPM can help determine the performance of a given process execution (a so-called process case, e.g., an order, a purchase request, or a claim) against its performance measures and performance goals (Dumas et al., 2018).

| ID_case | ID_event | Timestamp | Activity_name | Attrb_1 | Attrb_2 |
|---|---|---|---|---|---|
| 1 | 1 | 10-28-2021 | A | 1547 | 40 |
| | 2 | 10-29-2021 | B | 1547 | 40 |
| | 3 | 10-30-2021 | D | 1547 | 40 |
| | 4 | 11-03-2021 | E | 1547 | 40 |
| | 5 | 11-042021 | G | 1547 | 40 |
| 2 | 6 | 10-15-2021 | A | 950 | 38 |
| | 7 | 10-16-2021 | C | 950 | 38 |
| | 8 | 10-17-2021 | D | 950 | 38 |
| | 9 | 10-19-2021 | G | 950 | 38 |
| | 10 | 10-31-2021 | A | 1547 | 55 |
| | 11 | 11-01-2021 | C | 1547 | 55 |
| | 12 | 11-05-2021 | D | 1547 | 55 |
| | 13 | 11-06-2021 | E | 1547 | 55 |
| | 14 | 11-07-2021 | F | 1547 | 55 |
| | 15 | 11-10-2021 | G | 1547 | 55 |
| | ... | | ... | | |
| 500 | 2702 | 06-11-2022 | A | 1600 | 38 |
| | 2703 | 06-12-2022 | B | 1600 | 38 |
| | 2704 | 06-14-2022 | D | 1600 | 38 |
| | 2705 | 06-15-2022 | E | 1600 | 38 |
| | 2704 | 06-16-2022 | G | 1600 | 38 |

**Event log (*L*)**

Figure 3: Illustrative example of an event log snippet with respect to the business process model in Figure 1.

PPM can support process analysts in various contexts. For instance, Robeer (2018) proposed results-oriented PPM, which refers to classifying each ongoing case of a process according to a certain set of possible categorical outcomes, to predict the remaining incomplete case processing time and satisfy a certain customer delivery. Maggi et al. (2014), in turn, proposed a compliance monitoring approach in which predictions and recommendations are made based on what activities to perform and what input data values to provide to minimize the likelihood of violating business constraints.

PPM seeks to provide process analysts with meaningful information about what they are interested in analyzing to make the best decision in order to meet business goals, in terms of key performance indicators, service level agreements, and satisfactory deliverables. Although the results produced by the machine learning-based predictive models currently used have a satisfactory accuracy, the origin and reasoning of these results cannot be easily interpreted by process analysts (Márquez-Chamorro et al., 2017). Thus, the process analyst should simply rely on a predictive model with high accuracy and follow the algorithm's suggestion without knowing the details of how the prediction was performed. Mainly in high-risk processes, e.g., healthcare processes, or processes for which execution time is a critical factor, e.g., financial processes, the process analyst should always be provided with as much information as possible considering the details about the decision made by the algorithm used for predictions.

## 2.2 Interpretable Predictive Models

An interpretable machine learning system is capable of explaining its decisions in such a way that humans can understand the full logic behind those decisions (Roscher et al., 2020; Sagi and Rokach, 2020). We are here especially interested in a class of machine learning models – the predictive models.

A criterion used to classify interpretability methods refers to the way in which the predictive model outcome is obtained. According to this criterion, the interpretability method can be local, if it explains the behavior of the predictive model for a given instance, or global, if it seeks to do it for the model as a whole (Belle and Papantonis, 2021). A local interpretation method explains individual predictions (Ribeiro et al., 2016). A sensitivity analysis can be used to inspect how the outcome of a model locally depends on different input parameters (Roscher et al., 2020). For example, suppose there is a black box predictive model where you can enter data points and get the model's predictions. Changes to parameters can be made as many times as necessary to understand why the model made a prediction for a given piece of data.

Ribeiro et al. (2016) proposed the Local Interpretable Model-agnostic Explanations (LIME) algorithm designed to faithfully and locally approximate an interpretable model over the interpretable representation[1]. The LIME procedure is split into the following steps:

1. Selecting an instance $x$ for which an explanation of the prediction provided by the original predictive model $f$ (seen as a black box) is desired, and creating $x'$ by mapping $x$ to the interpretable representation space.
2. Randomly and uniformly perturbing instances in the neighborhood of $x'$ resulting in the dataset $Z'$, and weighting the new instances in $Z'$ according to their similarity to $x'$.
3. Retrieving the weighted instance dataset $Z$ by mapping $Z'$ to the original representation space, and getting the predictions from $f$ for $Z$.
4. Training the local interpretable predictive model $g$ over $Z'$ using the labeling obtained for $Z$.
5. Explaining the prediction of $x'$ using the local interpretable predictive model $g$.

While LIME works locally on specific instances, submodular pick LIME (SP-LIME) (Ribeiro et al., 2016) works globally to evaluate and assess the predictive model as a whole. SP-LIME aims to provide a global understanding of the predictive model by explaining a set of individual instances. Its purpose is to select a set of diverse, representative instances from the dataset and apply LIME to them to provide interpretations. Representative records are chosen non-redundantly aiming to cover as many relevant features as possible; features that explain many different instances have higher importance scores.

# 3 PROPOSED APPROACH

VisInter4PPM (visual interpretability for PPM) is a business-oriented approach designed to visually support the interpretability of results in PPM.

The approach is split into two parts. The first concerns the creation of the non-interpretable predictive model and the application of SP-LIME in that model to create the local approximate predictive model (cf. Figure 4). The second refers to the visual projection of SP-LIME explanations onto the process model, which can be applied locally, to explain the prediction at the instance (or case) level, or globally, to provide a global explanation of the learning achieved by the predictive model $f$ (cf. Figure 5).

From the flow shown in Figure 4, the approach requires *(i)* filtering the event log $L$ from the analysis of the perspectives of interest, *(ii)* creating the predictor, and *(iii)* applying SP-LIME (cf. Section 2.2). In the current version of the proposed approach, only the control flow perspective is being addressed. After obtaining the explanations from SP-LIME, the explanations produced must be adequate to enable the analysis of a business analyst, according to the flow proposed in Figure 5.

According to Figure 5, SP-LIME explanations for a given instance are projected onto the process model by coloring the activities in the model[2]. Each dimension of the SP-LIME interpretable feature space refers

[1] An interpretable representation is one that can be understood by humans, regardless of the actual features used by the prediction model (Ribeiro et al., 2016), and that explains the predictions of any classifier or regressor.

[2] Color graphical projection onto the process model is

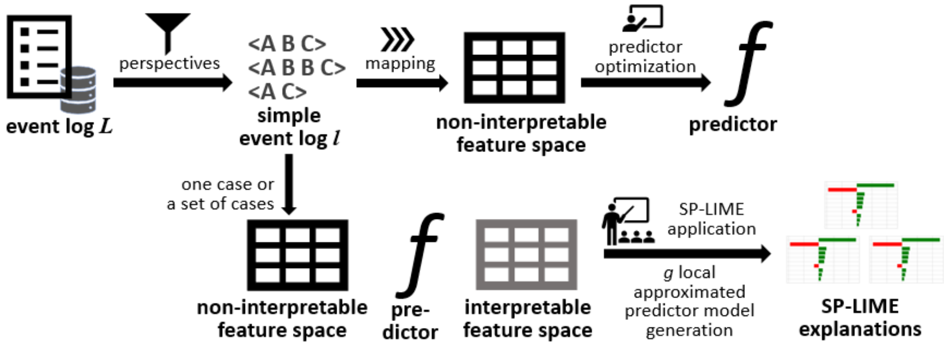


Figure 4: VisInter4PPM – creating SP-LIME explanations.

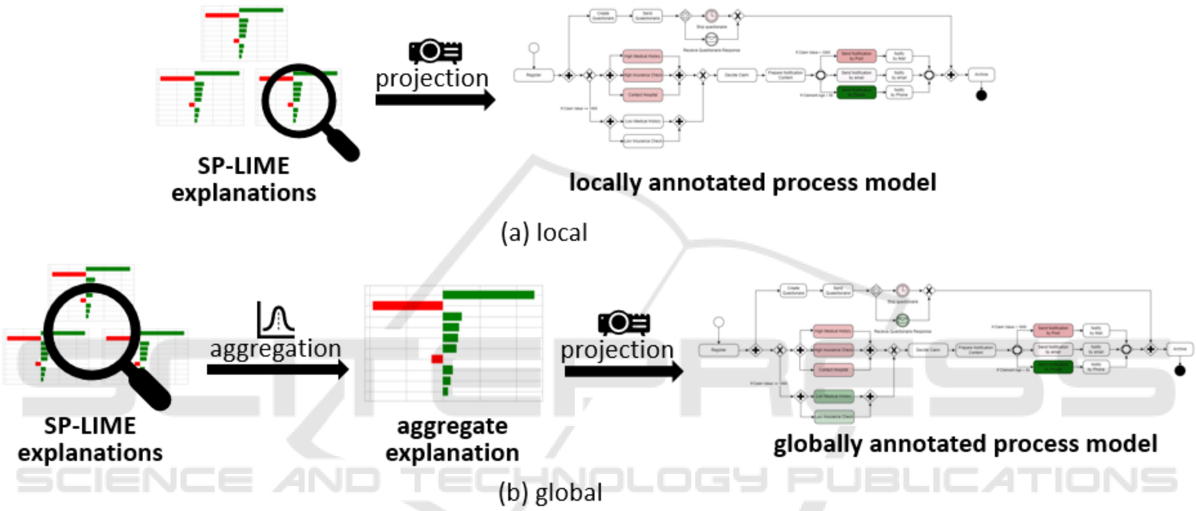




Figure 5: VisInter4PPM – projecting explanation onto process model.

to an activity of the process model, and the occurrence or not of the activity values the corresponding dimension as 1 or 0, respectively. Explanations about the occurrence of an activity are projected onto the process model in intensities of red and green, where red means a negative influence on the predictor's decision and green means a positive influence on the predictor's decision. The intensity of the influence of the activity's occurrence on the predictor's decision (according to the weight assigned by SP-LIME to the corresponding feature) determines the intensity of the color used to color the activities in the process model. The darker the color, the greater the influence. Explanations about the non-occurrence of an activity are disregarded, as the interpretation of a prediction can be focused only on the positive or negative influence that an activity exerts when it is performed.

Figure 5 highlights two possibilities for applying the interpretable predictive model. For the local interpretation (cf. Figure 5(a)), the user can choose specific instances, among those offered by SP-LIME, for which they want to view and analyze the individual explanations. As for the global interpretation (cf. Figure 5(b)), an aggregation of explanations is projected onto the process model, providing a general explanation for all instances offered by SP-LIME. Aggregation is performed through the average calculated on the weights assigned by SP-LIME to each dimension of the interpretable space. The average represents the common behavior captured from several instances, which can highlight different features, in order to unify the representativeness of each instance.

## 4 EXPERIMENTAL STUDY

We report here the application of the proposed approach in an example scenario through an experimental study. We present the business process and event

not yet automated.

log used, the settings and execution of the experiment, the results achieved, and the respective analysis.

## 4.1 Business Process and Event Log

The event log refers to an illustrative health insurance claim management process in a travel agency that has been used in process mining studies (Maisenbacher and Weidlich, 2017; Rizzi et al., 2020). We adapted the process model to merge the alternative activities *Accept claim* and *Reject claim* into only one activity *Decide claim*, as the occurrence of that original activities is correlated with the outcome target for the predictive model. The resulting business process comprises 19 activities (including two intermediate events). Figure 6 shows the adapted process modeled in business process model and notation (BPMN).

The business process begins with the claim registration. Then, alternative activities are carried out to analyze the registered claim, depending on the claim value. After analysis, a decision is made on the claim and the claimant is notified. Notification may be by post, email, or telephone, depending on the claim value and claimant age. In parallel, a questionnaire is sent to the claimant, who has a deadline to respond to it. Finally, the claim is archived. The synthetic event log created comprises 35,358 events and 3,200 cases with a maximum case length of 16 events.

## 4.2 Preprocessing

In this experiment, we assume the travel agency wants to predict whether a claim will be accepted (positive-case outcome) or rejected (negative-case outcome) for a running process case. Thus, we have a categorical prediction problem (a classification problem) to solve.

To train the classification model, we labeled the event log as reported in Table 1. As a result, the event log has 1,326 accepted cases and 1,874 rejected cases. In this event log, there are six variants of simple traces associated to accepted cases, and ten variants of simple traces associated to rejected cases.

The following pre-processing steps were carried out before training the classifier:

1. The original event log[3] was filtered according to the attribute *Activity* to support the control-flow analysis; and a simple event log $l$ was created as a multiset of simple traces.
2. An alternate simple event log $l'$ was created on top of $l$ considering one occurrence of each simple trace variant (disregarding data about the frequency of simple traces in $l$), resulting in a simple event log with 16 simple traces.
3. A frequency-based encoding was applied into the simple event log, following the procedure suggested by Rizzi et al. (2020); i.e., each trace was represented as a feature vector in which each feature represents an activity and is valued with the number of occurrences of that specific activity in the trace. As there is no loop in the process model under analysis, the result was a binary encoding.
4. To apply SP-LIME independently to each prediction class existing in $l'$, two subsets of instances were created: ${l_r}'$, referring to the subset of instances associated with the target *rejected*; and ${l_a}'$, referring to the subset of instances associated with the target *accepted*.

Table 1: Rules for labeling the event log.

| ID | Constraint | Label |
|---|---|---|
| 1 | claim value $>$ 1000 AND<br>(claimant age $\leq$ 50 OR<br>receive questionnaire response $=$ false) | rejected<br>(false) |
| 2 | claim value $>$ 1000 AND<br>(claimant age $>$ 50 AND<br>receive questionnaire response $=$ true) | accepted<br>(true) |
| 3 | claim value $\leq$ 1000 AND<br>skip questionnaire $=$ false | accepted<br>(true) |
| 4 | claim value $\leq$ 1000 AND<br>skip questionnaire $=$ true | rejected<br>(false) |

## 4.3 Experiment Setup

The experiment aimed to apply the proposed approach to visually project onto the process model which activities most influence the prediction of the case outcome as provided by the classifier model.

Following the strategy depicted in Figure 4 and Figure 5, the following steps were performed:

1. Construction of the classifier using the k-NN algorithm and the event log $l'$. k-NN was chosen due to its low parameterization and training complexity, as the goal of this experiment focuses on the explanation visualization. k-NN was run using the Euclidean distance with $k = 3$ (value chosen via tests with $k$ ranging in $[2, 15]$). The event log $l'$ was chosen aiming to isolate trace frequency bias on classifier decisions[4]. The classifier was

[3]Event logs and codes are available at https://github.com/double-blind.

[4]The event log $l$ could be used alternatively. However, k-NN should be properly adapted to deal with unbalanced

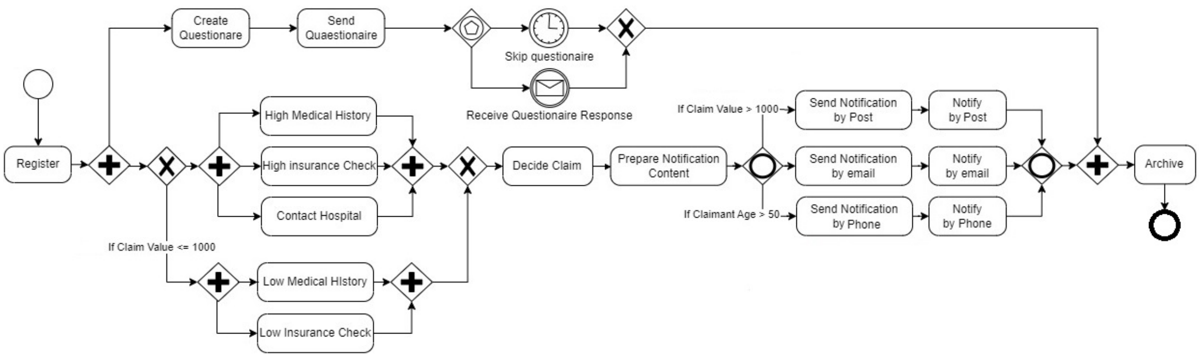


Figure 6: Health insurance claim management process in a travel agency, adapted from (Rizzi et al., 2020).

built overfitted over the full event log $l'$ to create a global environment for viewing explanations of the process under analysis in this experiment.

2. Application of SP-LIME to generate explanations for the decisions of the k-NN classifier. SP-LIME receives as input the previously trained classifier, the set of instances for which explanations must be obtained ($l_r{}'$ or $l_a{}'$), the number of features to be considered in the explanations ($\#f$), and the number of explanation instances to be produced ($\#r$). These following values were chosen: $\#f =$ total number of activities existing in $l'$, to allow the explainer to explore the entire interpretable representation space, and $\#r = 5$ for each application scenario ($l_r{}'$ e $l_a{}'$), which is an arbitrary value chosen for purposes of exploratory study.
3. Implementation of the interaction explanations with the explanations provided by SP-LIME, considering the visual projection of the explanations applied locally or globally. For the former, the business analyst must choose an instance $x$ for which they want a visualization is desired, i.e., let $x = user\_pick(l_r{}')$ or $user\_pick(l_a{}')$. Explanations for a set of instances $X$ are aggregated (via the aggregation function *mean*), where $X$ comprises all instances $x$ of $l_r{}'$ or all instances $x$ of $l_a{}'$, for which the business analyst is interested in the interpretation of the prediction.

## 4.4 Results and Discussion

Since the k-NN classifier is overfitted on $l'$, the way to assess its quality is via resubstitution error $\varepsilon$. According to Han et al. (2012),

> "*(...) if we were to use the training set (instead of a test set) to estimate the error rate of a model, this quantity is known as the resubstitution error. This error estimate is optimistic of the true error rate because the model is not tested on any samples that it has not already seen.*"

Thus, the error value does not reflect a generalization measure, but only shows the upper limit for the learning effectiveness achieved by k-NN over $l'$. In this experiment, $\varepsilon = 1 - FScore = 0.0625$. The classifier was unable to perfectly approximate the decision surface to $l'$, making a classification error for the following simple trace:

$\sigma = \langle$ *Register, Create Questionnaire, Send Questionnaire, Receive Questionnaire Response, High Medical History, High Insurance Check, Contact Hospital, Decide Claim, Prepare Notification Content, Send Notification by Post, Notify by Post, Send Notification by Phone, Notify by Phone, Archive* $\rangle$

***True label**: TRUE; **k-NN prediction**: FALSE.*

Figure 7 shows the result of post-processing the SP-LIME outcomes. The first block refers to the negative class (rejected claims), while the second block refers to the positive class (accepted claims). For each block, there are the five instances for which local explanations were created, followed by the aggregation (via *mean*) of those five explanations, feature by feature. The colors are applied proportionally, that is, the intensity of the color is proportional to the weight assumed by the feature in the SP-LIME explanation; the darkest tone refers to the highest absolute value.

As illustrative example, Figure 8 and Figure 9 show two representative instances (from Figure 7) in the original LIME format. These two instances are highlighted in Figure 7 surrounded by dashed lines. These examples show the potential difficulty for one to understand this data in order to interpret the classifier outcomes. On the left side of the chart, there are the relevant features chosen by LIME. The feature name is accompanied by the range of values it must take to have the relevance associated with that predic-

neighborhoods resulting from the inheritance of weights in the datapoints, resulting from the unbalance of $l$.

| SP-LIME index | Register | Create Question-naire | Send Question-naire | Skip Question-naire | Receive Question-naire Response | High Medical History | High Insurance Check | Contact Hospital | Low Medical History | Low Insurance Check | Decide Claim | Prepare Notifica-tion Content | Send Notifica-tion by Post | Notify by Post | Send Notifica-tion by e-mail | Notify by e-mail | Send Notifica-tion by Phone | Notify by Phone | Archive | KNN prediction |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | 0,0524 | 0,0544 | 0,0518 | | | | | 0,0557 | 0,0561 | | | | | | 0 |
| 2 | | | | | | | | | -0,0555 | -0,0581 | | | | | 0,0079 | 0,0052 | -0,0986 | -0,0902 | | |
| 3 | | | | | -0,3411 | 0,0554 | 0,0624 | 0,0512 | | | | | 0,0517 | 0,0508 | -0,0005 | 0,0045 | | | | |
| 4 | | | | | | 0,0561 | 0,0516 | 0,0444 | | | | | 0,0536 | 0,0468 | -0,0034 | 0,0052 | -0,1035 | -0,1045 | | |
| 5 | | | | | | 0,0559 | 0,0550 | 0,0510 | | | | | 0,0519 | 0,0469 | | | -0,1066 | -0,0984 | | |
| Mean | | | | | -0,3411 | 0,0550 | 0,0558 | 0,0496 | -0,0555 | -0,0581 | | | 0,0532 | 0,0501 | 0,0013 | 0,0050 | -0,1029 | -0,0977 | | |
| 1 | | | | | | | | | 0,0230 | 0,0221 | | | | | | | 0,1035 | 0,1014 | | 1 |
| 2 | | | | | | -0,0180 | -0,0176 | -0,0177 | | | | | -0,0272 | -0,0205 | 0,0076 | 0,0103 | 0,1210 | 0,1095 | | |
| 3 | | | | | | | | | 0,0178 | 0,0136 | | | | | | | | | | |
| 4 | | | | | | | | | 0,0187 | 0,0136 | | | | | -0,0003 | 0,0024 | 0,1108 | 0,1070 | | |
| 5 | | | | | | | | | 0,0133 | 0,0196 | | | | | 0,0002 | 0,0064 | | | | |
| Mean | | | | | | -0,0180 | -0,0176 | -0,0177 | 0,0182 | 0,0172 | | | -0,0272 | -0,0205 | 0,0025 | 0,0064 | 0,1118 | 0,1060 | | |

Figure 7: Result of post-processing the SP-LIME outcomes (considering only feature values = 1).

tion explanation. In this experiment, there are only two options for values to be assumed by the features – 0 (which means that the activity is not executed) or 1 (which means that the activity is executed). Features accompanied by "<= 1.00" or "> 0" refer to executed activities (i.e., the corresponding activity is present in the simple trace under analysis), while the others refer to those not executed. As for the right side of the chart, the size of the green and red bars represent the influence that the corresponding feature had on the predicted result. Green bars refer to positive influence, while red bars refer to negative influence. Both Figure 8 and Figure 9 present interpretability values for the negative class, i.e., the green bars refer to the positive influence of a given feature (representing either the execution or the non-execution of an activity) for the prediction of a *rejected claim*. Only features referring to activities executed (i.e., whose value = 1) are mapped with the data in Figure 7.

Figure 10 and Figure 11 show the locally annotated process models corresponding to both instances presented in Figure 8 and Figure 9, respectively. These annotated process models were created by visually projecting the values in Figure 7 in the original process model (cf. Figure 6). The exact same color intensity scale is used. Understanding this visual information is easier for a business analyst. Figure 10, e.g., shows the influence of the execution of activities for the rejection of the claim associated with instance SP-LIME index = 2. One can observe that, for this instance, there are four activities whose execution negatively influenced this rejection, as they are colored in red, while there are two activities whose execution positively influenced, as they are colored in green. Again, color intensity represents a greater or lesser influence, whether positive or negative. Colorless activities either were not carried out or did not exert any influence, whether positive or negative. In Figure 11, one can observe seven activity occurrences positively influencing the rejection, while only one negatively influencing rejection.

As each analyzed instance can offer different points of view on the influence of each process activity, a global view can be more useful for the business analyst. Figure 12 shows the globally annotated process model resulting from the aggregation of the interpretability values for each model feature, by averaging the feature values for the five most representative instances chosen by SP-LIME. One can see that, overall, the occurrence of *Receive questionnaire response* has the greatest negative influence for a claim to be rejected; i.e., when this event occurs, the claim will likely not be rejected. Moreover, when *Low medical history* and *Low insurance check* are carried out, as well as when *Send notification by phone* and *Notify by phone* are carried out, the claim is also likely not to be rejected, although less likely. On the other hand, when *High medical history*, *High insurance check*, and *Contact hospital* are carried out, as well as *Send notification by post* and *Notify by post*, the claim is also likely to be rejected. In addition, when *Send notification by email* and *Notify by email* are carried out, there is a chance, albeit small, that the claim will be rejected.

Similarly, Figure 13 shows the overview for predicting positive cases, i.e., accepting claims. The globally annotated process models in Figure 12 and Figure 13 are partially complementary, as one represents the positive cases and the other the negative ones. However, the classification is not fully binary; for example, the existence of an *OR* gateway adds complexity to decisions. For Figure 13, the occurrences *Send notification by phone* and *Notify by phone* are those that most increase the chance of the request being accepted.

## 5 RELATED WORK

In 2017, Márquez-Chamorro et al. (2017) presented a survey to understanding the state-of-the-art in predictive monitoring of business processes. The authors reported that, until that time, few initiatives were concerned with the interpretability of the predictive mod-

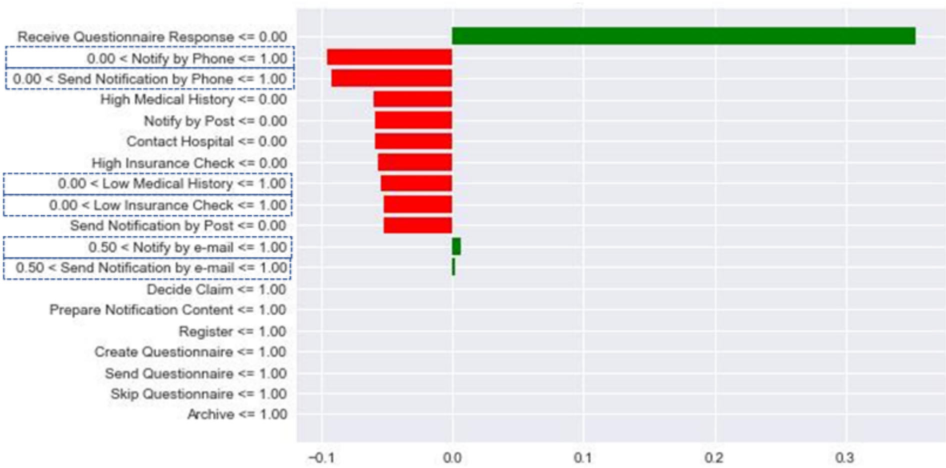


Figure 8: A representative instance in the original LIME format (SP-LIME index = 2 for negative class).

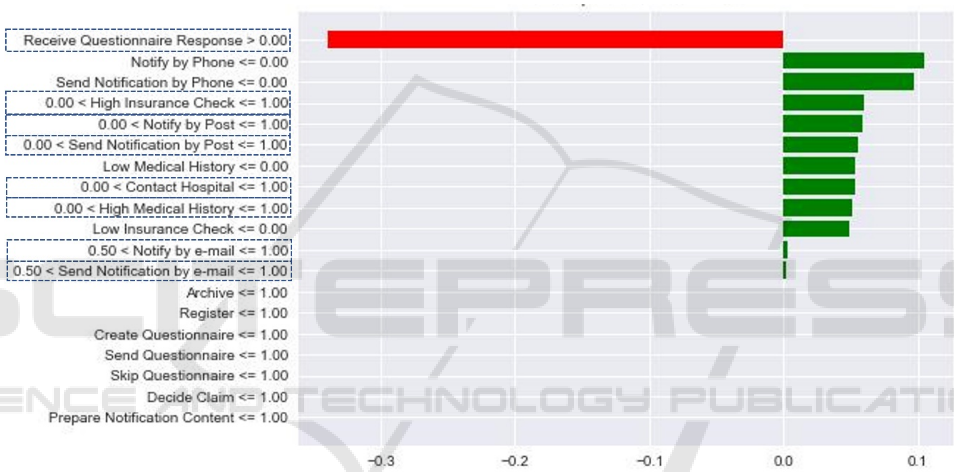


Figure 9: A representative instance in the original LIME format (SP-LIME index = 3 for negative class).



els. Among the 41 studies analyzed, three explicitly mention the concern with proposing interpretable predictors. The authors point out most works focusing on classification tasks do not deal with process-conscious methods, which prevents them from directly bringing useful interpretations and explanations to a business analysis. Finally, these authors warn that "*little attention has been given to providing recommendations and explaining the prediction values to the users so that they can determine the best way to act upon*". However, from 2017 to nowadays, this research gap has been filled. Through an exploratory study, we identified recent initiatives dealing with the interpretability in business process monitoring.

Frameworks to discover the set of attributes that most influence a predictor is studied in Galanti et al. (2020); Mehdiyev and Fettke (2021); Weinzierl et al. (2020). These works differ in the approach used to obtain the explanations, the prediction tasks involved, and the way the explanations are returned to a user: Galanti et al. (2020) use game theory (Shapley Values) to get explanations to the prediction model, instantiate their framework for predicting the remaining time, activity occurrences and case costs, and offer explanations formatted as tables and heatmaps related to characteristic's values and weights; Mehdiyev and Fettke (2021) apply surrogate decision trees to explain the decision of neural network-based predictors applied for predicting next activity, outcome cases, and service level agreement violations, and return the explanations using the hierarchical tree structure and IF-THEN rules; Weinzierl et al. (2020) use a layer-wise relevance propagation method on predictors built with long short-term memory neural networks, apply the approach to the next task prediction problem, and provide the results as heatmaps to show the relevance of the input activities.

The proposition of white-box predictors is also re-

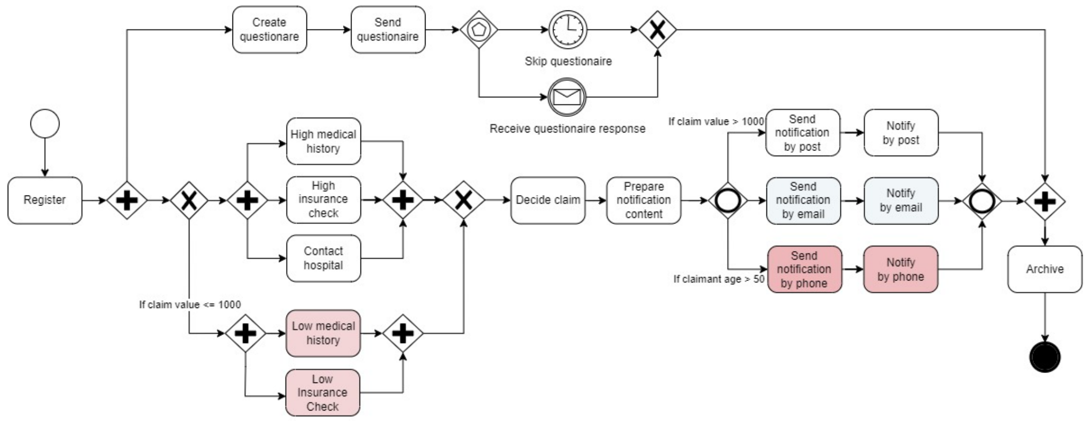


Figure 10: Example of locally annotated process model (SP-LIME index = 2 for negative class).

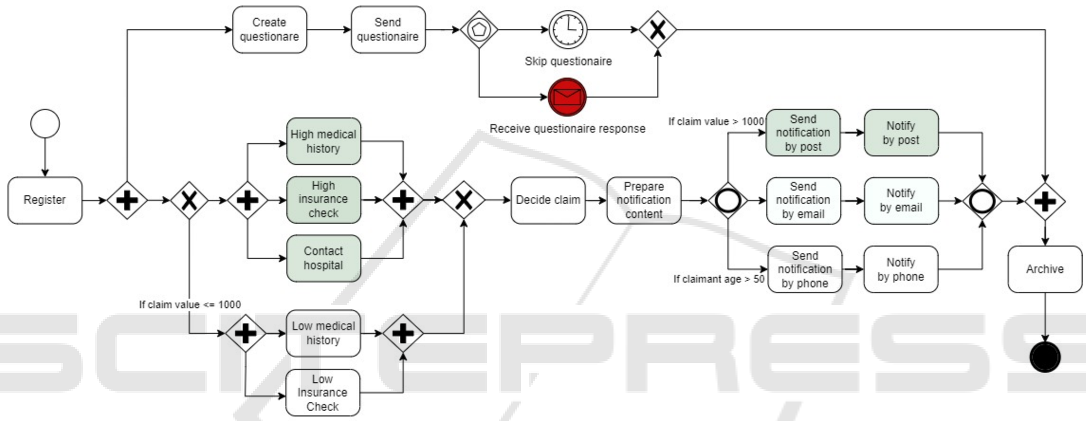


Figure 11: Example of locally annotated process model (SP-LIME index = 3 for negative class).



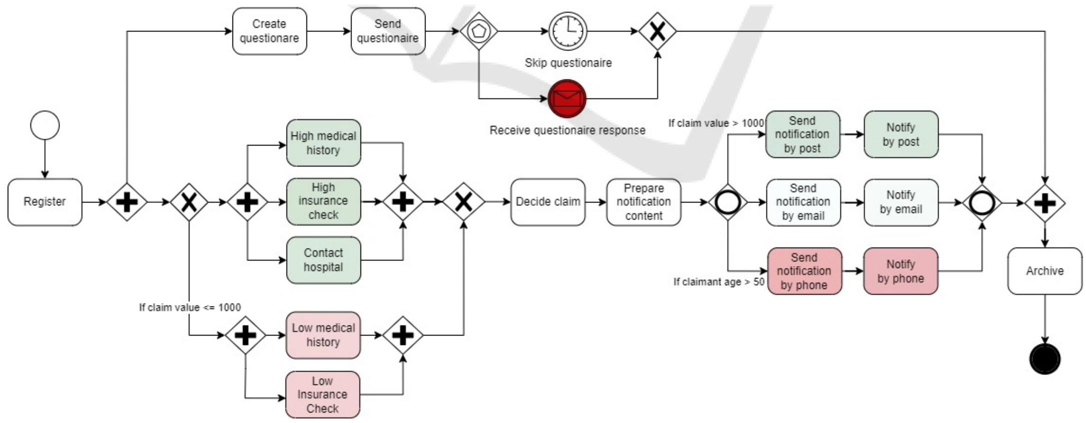


Figure 12: Globally annotated process model (negative class).

ceiving attention from the predictive process monitoring area. The framework presented by Verenich et al. (2019a) first predicts a performance indicator at the level of activities and then aggregates these predictions at the level of a process instance through flow analysis techniques. Wickramanayake et al. (2022), in turn, introduce two new interpretable attention-based models, as they incorporate interpretability straight into the structure of a process predictive model. In this sense, the predictors themselves can inform what the resulting prediction is and why it was got.

Different lines of study are presented by Rizzi et al. (2020) and Warmuth and Leopold (2022). In Rizzi et al. (2020), the authors apply the clas-

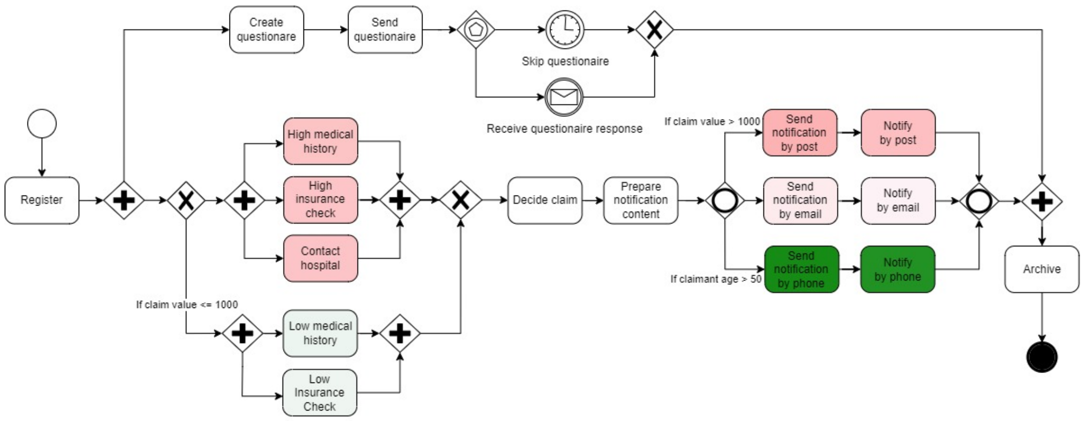


Figure 13: Globally annotated process model (positive class).

sic interpretability approach using approximate interpretable models (LIME and SHARP) to explain uninterpretable models. The differential of this work is the interpretations are employed to enhance the training of the original predictor. They identify the most common features that induce the predictor to make mistakes. Then, they alter such features to reduce their impact on the result, thereby improving the predictor accuracy. Finally, Warmuth and Leopold (2022) use textual information combined with non-textual data as a basis for constructing explanations. As a result, besides returning explanations based on the influence of features for decision making, the authors can also highlight such influence within a textual description of the context associated to the business process.

The studies presented here motivate the use of interpretability techniques or white-box models, emphasizing the need to provide value-added and business-oriented information, endowing a business analyst to make robust and justifiable decisions in predictive process monitoring field. However, none of them present the visualization of explanations at the top of the process model associated with the business, positioning the approach discussed herein as a possibility to fill a gap in the process mining practice.

## 6 CONCLUSION

In this paper, we introduce a business-oriented approach to visualization of explanations derived from the use of SP-LIME over a classification model. When SP-LIME results are projected onto the process model, a business analyst can quickly identify the activities that directly intervened, and to what extent, in the decision provided by a predictive model. By making information accessible to the business analyst, process mining approaches help to avoid unfair and inaccurate actions within the organizational context, and promote the transparency of the decision-making process.

The VisInter4PPM approach was introduced in this paper and instantiated in an experiment considering a synthetic event log, the control flow perspective of analysis, and the SP-LIME method. The next steps in the development of VisInter4PPM include the incorporation of visualization elements enabling other perspectives of analysis (performance, resource and case), commonly used in the process mining field, the use of other methods for explaining predictors, and the experimentation with real-world event logs.

## REFERENCES

Aalst, W. M. P. (2016). *Process Mining – Data Science in Action*. Springer.

Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Inf. Fusion*, 58:82 – 115.

Belle, V. and Papantonis, I. (2021). Principles and practice of explainable machine learning. *Front. Big Data*, page 39.

de Sousa, R. G., Peres, S. M., Fantinato, M., and Reijers, H. A. (2021). Concept drift detection and localization in process mining: An integrated and efficient approach enabled by trace clustering. In *Proc. of the 36th Annual ACM Symp. Appl. Comput*, page 364–373.

Dumas, M., Rosa, M. L., Mendling, J., and Reijers, H. (2018). *Fundamentals of Business Process Management*. Springer, 2nd edition.

Galanti, R., Coma-Puig, B., de Leoni, M., Carmona, J., and Navarin, N. (2020). Explainable predictive process monitoring. In *Proceedings of the 2nd International Conference on Process Mining*, pages 1–8.

Han, J., Kamber, M., and Pei, J. (2012). *Data mining concepts and techniques, third edition*. Morgan Kaufmann Publishers, 3rd edition.

Holzinger, A. (2018). From machine learning to explainable ai. In *Proc. of 2018 World Symp. on Digital Intell. Syst. Mach.*, pages 55–66.

Kim, J., Comuzzi, M., Dumas, M., Maggi, F. M., and Teinemaa, I. (2022). Encoding resource experience for predictive process monitoring. *Decis. Support Syst*, 153:113669.

Maggi, F. M., Francescomarino, C. D., Dumas, M., and Ghidini, C. (2014). Predictive monitoring of business processes. In *Int. Conf. Adv. Inf. Syst. Eng.*, pages 457–472.

Maisenbacher, M. and Weidlich, M. (2017). Handling concept drift in predictive process monitoring. In *Proc. of 2017 IEEE Int. Conf. Serv. Comput.*, pages 1–8.

Márquez-Chamorro, A. E., Resinas, M., and Ruiz-Cortés, A. (2017). Predictive monitoring of business processes: A survey. *IEEE Trans. Serv. Comput.*, 11(6):962–977.

Mehdiyev, N., Evermann, J., and Fettke, P. (2020). A novel business process prediction model using a deep learning method. *Bus. & Inf. Syst. Eng.*, 62(2):143–157.

Mehdiyev, N. and Fettke, P. (2021). Explainable artificial intelligence for process mining: A general overview and application of a novel local explanation approach for predictive process monitoring. In *Proc. of Interpretable Artif. Intell.: A Perspective of Granular Comput.*, pages 1–28.

Polato, M., Sperduti, A., Burattin, A., and de Leoni, M. (2018). Time and activity sequence prediction of business process instances. *Comput.*, 100(9):1005–1031.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?" explaining the predictions of any classifier. In *Proc. of the 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Min.*, pages 1135–1144.

Rizzi, W., Di Francescomarino, C., and Maggi, F. M. (2020). Explainability in predictive process monitoring: When understanding helps improving. In *Proc. of the Bus. Process Manage. Forum*, pages 141–158.

Robeer, M. J. (2018). Contrastive explanation for machine learning. Master's thesis, Utrecht University.

Roscher, R., Bohn, B., Duarte, M. F., and Garcke, J. (2020). Explainable machine learning for scientific insights and discoveries. *IEEE Access*, 8:42200–42216.

Sagi, O. and Rokach, L. (2020). Explainable decision forest: Transforming a decision forest into an interpretable tree. *Inf. Fusion*, 61:124–138.

Teinemaa, I., Dumas, M., Rosa, M. L., and Maggi, F. M. (2019). Outcome-oriented predictive process monitoring: Review and benchmark. *ACM Trans. Knowl. Discovery Data*, 13(2):1–57.

Verenich, I., Dumas, M., La Rosa, M., and Nguyen, H. (2019a). Predicting process performance: A white-box approach based on process models. *J. of Software: Evol. Process*, 31(6).

Verenich, I., Dumas, M., Rosa, M. L., Maggi, F. M., and Teinemaa, I. (2019b). Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. *ACM Transactions on Intelligent Systems and Technology*, 10(4).

Warmuth, C. and Leopold, H. (2022). On the potential of textual data for explainable predictive process monitoring. In *Proc. of 3rd Int. Workshop on Leveraging Mach. Learn. in Process Min.*, pages 1–12.

Weinzierl, S., Zilker, S., Brunk, J., Revoredo, K., Matzner, M., and Becker, J. (2020). Xnap: Making lstm-based next activity predictions explainable by using lrp. In *Proc. of Workshop on Artif. Intell. Bus. Process Manage.*, pages 129–141.

Wickramanayake, B., He, Z., Ouyang, C., Moreira, C., Xu, Y., and Sindhgatta, R. (2022). Building interpretable models for business process prediction using shared and specialised attention mechanisms. *Knowledge-Based Syst.*, 248.