



Why Should I Trust Your Explanation? An Evaluation Approach for XAI Methods Applied to Predictive Process Monitoring Results

Ghada Elkhawaga , Omar M. Elzeki , Mervat Abu-Elkheir , *Member, IEEE*, and Manfred Reichert 

Abstract—As a use case of process mining, predictive process monitoring (PPM) aims to provide information on the future course of running business process instances. A large number of available PPM approaches adopt predictive models based on machine learning (ML). With the improved efficiency and accuracy of ML models usually being coupled with increasing complexity, their understandability becomes compromised. Having the user at the center of attention, various eXplainable artificial intelligence (XAI) methods emerged to provide users with explanations of the reasoning process of an ML model. Though there is a growing interest in applying XAI methods to PPM results, various proposals have been made to evaluate explanations according to different criteria. In this article, we propose an approach to quantitatively evaluate XAI methods concerning their ability to reflect the facts learned from the underlying stores of business-related data, i.e., event logs. Our approach includes procedures to extract features that are crucial for generating predictions. Moreover, it computes ratios that have proven to be useful in differentiating XAI methods. We conduct experiments that produce useful insights into the effects of the various choices made through a PPM workflow. We can show that underlying data and model issues can be highlighted using the applied XAI methods. Furthermore, we could penalize and reward XAI methods for achieving certain levels of consistency with the facts learned about the underlying data. Our approach has been applied to different real-life event logs using different configurations of the PPM workflow.

Impact Statement—As ML models are used to generate predictions for running business process instances, the outcomes of these models should be justifiable to users. To achieve this, explanation methods are applied on top of ML models.

Manuscript received 21 October 2022; revised 16 February 2023 and 30 August 2023; accepted 14 January 2024. Date of publication 22 January 2024; date of current version 9 April 2024. This article was recommended for publication by Associate Editor Koduvayur P. Subbalakshmi upon evaluation of the reviewers' comments. (*Corresponding author: Ghada Elkhawaga.*)

Ghada Elkhawaga is with the Institute of Databases and Information Systems, Ulm University, 89081 Ulm, Germany, and also with the Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt (e-mail: ghada.el-khawaga@uni-ulm.de).

Omar M. Elzeki is with the Faculty of Computer Science and Engineering, New Mansoura University, Mansoura 35516, Egypt, and also with the Faculty of Computers and Information, Mansoura University, Mansoura 35516, Egypt (e-mail: omar_m_elzeki@mans.edu.eg).

Mervat Abu-Elkheir is with the Faculty of Media Engineering and Technology, German University in Cairo, New Cairo 11835, Egypt (e-mail: mervat.abuelkheir@guc.edu.eg).

Manfred Reichert is with the Institute of Databases and Information Systems, Ulm University, 89081 Ulm, Germany (e-mail: manfred.reichert@uni-ulm.de).

This article has supplementary downloadable material available at <https://doi.org/10.1109/TAI.2024.3357041>, provided by the authors.

Digital Object Identifier 10.1109/TAI.2024.3357041

However, explanations need to be evaluated concerning the valuable information they convey about the predictive model and their ability to encode underlying data facts. In other words, an explainability method should be evaluated concerning its ability to match model inputs to its outputs. Our approach provides a means to evaluate and compare explainability methods concerned with the global explainability of the entire reasoning process of an ML model. Based on experimental settings, where each step of the PPM workflow is changeable, we could study the ability of our approach to evaluate different combinations of data, preprocessing configurations, modeling, and explanation methods. This approach allows an understanding of which PPM workflow configurations increase the ability of an explanation method to make the prediction process transparent to users.

Index Terms—Explainability, functionally grounded evaluation, global explainability methods, predictive process monitoring, process mining, XAI.

I. INTRODUCTION

PREDICTIVE process monitoring (PPM) originates from the research domain of process mining [1]. Process mining aims to provide insights that contribute to enhancing running process executions, prevent an expected inefficiency in future executions, or optimize the order in which tasks are executed in the context of the respective business process. [2] defines PPM as *the set of runtime methods that aim to generate predictive models that may be used for predicting a particular value of a process instance given its partial trace and the event log of historical traces as inputs*. As advances have been made in the machine learning (ML) field with respect to the achieved performance, many PPM approaches tend to use ML models to generate accurate predictions. Given the complementary nature of research, however, these advances as well as challenges will be propagated across disciplines. Consequently, the challenges of ML are propagated to PPM approaches.

One of the main challenges of ML models is the tradeoff between accuracy and complexity. To tackle this challenge, recent research has focused on making the outcomes of predictive models more understandable and making the models themselves more interpretable for users being affected by these models. As a consequence, eXplainable artificial intelligence (XAI) has emerged and numerous proposals have been made to produce explanations that differ with respect to the complexity of the predictive model, the scale, and the size of the predictions to be explained, as well as the experience and knowledge of

the respective user. Several approaches have been proposed to support ML-based PPM by mechanisms to increase user trust in the generated predictions. To achieve this goal, XAI methods are applied to PPM results.

A. Problem Statement

The increasing use of XAI methods necessitates using metrics and techniques to evaluate these methods with respect to specific criteria. Particularly, it is necessary to evaluate the ability of an XAI method to reflect the knowledge learned by an ML model. With the increasing interest in applying XAI methods in PPM, a few proposals [3], [4], [5] were made to evaluate the explanations created for PPM results. This article proposes an approach for evaluating XAI methods with respect to their ability to transfer data facts learned by an ML model about PPM data. The proposed approach provides quantitative techniques to compare global XAI methods when explaining the predictions of one ML model while considering the data patterns that the model learned during the training process. We propose ratios to measure the consistency of an XAI method. In XAI literature, *consistency* is defined as a property of an explanation. According to [6], *consistency* of explanations corresponds to the similarity between the explanations created for similar predictions, which were made by two different ML models. However, it has been demonstrated that sometimes explanations can highlight features that do not have a statistical association with the prediction target [7]. In this article, the term *consistency* denotes the percentage of similarity between the set of essential features generated by an XAI method on one hand, and the principal feature set being crucial for generating predictions on the other. With the principal feature set, we mean features that have the highest influence on prediction results. We provide techniques to conclude the principal feature set from the original data. Furthermore, we provide metrics to compare and rank various XAI methods with respect to their consistency ratios (CRs) among other factors. We believe that our approach represents a starting point for other proposals in the same direction.

B. Contributions

As mentioned, in the PPM context, there is a need to provide mechanisms to quantitatively evaluate not only the resulting explanations but also the methods generating these explanations. To achieve this, we propose an approach to evaluate global model-agnostic XAI methods, which may be used to explain predictions in the PPM context. In summary, we make the following contributions.

- 1) We propose an approach that comprises four phases to compute ratios and metrics to differentiate XAI methods based on the consistency of their explanations with the ground truth learned about a particular event log.
- 2) We introduce the concept of XAI method consistency with underlying principal features and provide means to extract these principal features.
- 3) Based on a number of experiments, we study how the underlying data characteristics can be reflected through the obtained experimental results.

- 4) We implement the proposed approach and enable free access to this implementation for interested researchers to build upon.

Section II summarizes the necessary backgrounds and related research efforts. Our approach and the main contributions of our work are presented in Section III. In Section IV, we discuss the main research questions we try to address as well as the experimental settings we use to examine and apply the approach. The basic observations and discussions of the factors that lead to these observations are presented in Section V. Furthermore, we discuss related research on PPM with explanations in Section VI. The article concludes with a summary and an outlook in Section VII.

II. BACKGROUNDS

Our work follows two main research directions; *PPM* as a use case of process mining and *XAI*. Knowledge and techniques developed in the context of both fields constitute the main building blocks of our approach. This section introduces the backgrounds needed to familiarize the reader with the main concepts and techniques in the two research fields. Finally, we provide some background on feature selection (FS) methods that are used in the context of this work.

A. PPM

1) *Preliminaries*: The main input of a PPM task is an *event log*, which comprises a finite number of *traces* that document several executions of a specific process, i.e., *process instances*. While illustrating the basic notions, we will use activities from the order placement process as our example.

- 1) *Events and activities*: An event corresponds to the execution of a single step of the process and belongs to a single activity class. For example, the process of order placement defines a set of all allowed activities, including user logging (A_1), catalog browsing (A_2), products' choosing (A_3), cart modification (A_4), cart approval (A_5), and payment (A_6). Each of these activities is called an *activity class*. Whenever a user places an order, instances of these activities are executed. In this case, they are called *events*. With each event, there is payload information associated with it. The timestamp (T) of an event, the resource that executed the event, and important notes (text) are examples of payload. Therefore, an event $e_i \in \epsilon$ is a tuple $e_i = (c_i, a_i, t_i, d_{ij})$ with $c_i \in C$ representing a case, i.e., process instance identifier, $a_i \in A$, $t_i \in T$ representing mandatory attributes, and $d_{ij} \in D_j$, $1 \leq j \leq m$ are all additional attributes.
- 2) *Traces and event log*: A trace $\sigma = \langle e_1, e_2, \dots, e_n \rangle$ represents a sequence of executed events recorded for a specific process instance. An *event log* stores all traces of process instances executed in the context of a particular business process. Each process instance is stored over multiple rows in the event log. Each row stores an event along with its payload or associated attributes. Attributes having the same value in all rows of a process instance are *static attributes*. For example, the case ID and customer

TABLE I
EXAMPLE OF A TRACE

Case ID	Event ID	Activity	Timestamp	Resource	Requested Amount	Monthly Cost
C120	e_1	Create application	30 March 2018 10:07:22	John Doe	38 000	1281
C120	e_2	Validate application	30 March 2018 14:12:29	Ben Markus	38 000	231
C120	e_3	Decide	12 April 2018 11:15:30	Jill Adams	38 000	342
C120	e_4	Close application	23 April 2018 15:24:03	John Doe	38 000	1213

name are static attributes. Moreover, attributes whose values correspond to a specific event and change for the rows recording the corresponding process instance are *dynamic attributes*. For example, Table I represents an example of a trace of events, while each has associated dynamic and static attributes.

In this example, the *case ID* is an identifier of the trace, while the *event ID* is an identifier of the event to be represented by a given row. Fields that represent dynamic attributes such as T , *resource*, and *monthly cost* have different values for each row. The other fields, in turn, represent static attributes and have static values along the trace. *Requested amount* is an example of static attributes.

- 3) A *prefix*: It represents a subset of a trace, where $p_n = \langle e_1, e_2, \dots, e_n \rangle$ with $1 \leq n \leq k$ and k being the number of events executed within the trace. Note that a prefix can be extracted from traces following a gap between the events to be included in the prefix. For example, when the gap = 3 and prefix length = 5, then in this case $P_5 = \langle e_1, e_4, e_7, e_{10}, e_{13} \rangle$. An event log that contains prefixes of traces is called *prefix log*.

2) *Predictive Process Monitoring Workflow*: A plethora of research has been conducted to predict relevant information on the future course of a running process, facilitated by the availability of ML and statistical techniques. Prediction tasks include the prediction of the next activity to be executed, performance indicators, e.g., remaining time upon process completion or cost of executing an event, and process outcome [2], [8], [9]. This work focuses on predicting the outcome of a running process instance. Several studies and benchmarks were carried out to characterize the essential phases as well as associated methods and techniques used along the PPM workflow [2], [8], [9]. Our focus is on PPM workflow-related procedures that use an ML model to perform the prediction task. The procedures and steps performed along a PPM workflow do not differ according to different prediction tasks. A PPM workflow is divided into *offline* and *online* stages [9]. The offline stage is illustrated in the left part of Fig. 3. It consists of two basic levels.

- 1) *Event log preprocessing*: At this level, an event log is prepared and transformed into a format suitable as input for the selected ML model. The procedures applied at this level vary in their purpose and applied technique. The first step deals with *prefixing* process instances, i.e., cutting the process instances up to a certain length. The length and gaps used to truncate the instances are predefined to equip the model with enough information to make a prediction, while not exhausting the used computational resources with large amounts of data to be processed. *Bucketing* follows prefixing when preprocessing an event log. It means grouping similar prefixes in a bucket and treating each bucket as a separate sublog. The bucketing

technique defines the similarity criterion applied to group similar prefixes. Single, state, prefix length, clustering, and domain knowledge are well-studied and surveyed bucketing techniques [8], [9]. *Encoding* follows the bucketing step and aims to abstract a trace or a process instance while at the same time extracting features in a format suitable for a predictive ML model [9]. Four techniques are proposed in literature [8], [9] including static, index-based, last-state, and aggregation encoding techniques. In summary, the encoding step aims to convert prefixes p_n into a representative vector of numerical values. To transform the values of each attribute a , encoding strategies differ based on whether the attribute is static or dynamic, and whether it contains numerical or categorical values. Numerical values representing static attributes are passed to the final features' vector as-is. Furthermore, categorical values belonging to static attributes are one-hot encoded. Encoding static attributes is called *static encoding*. Encoding techniques for dynamic attributes differ with respect to how numerical and categorical values are transformed. For example, in *aggregation encoding*, each event belonging to a trace has a separate row. Numerical values are aggregated across the trace using aggregation functions (e.g., sum and mean). Furthermore, values of categorical attributes are either transformed into a boolean representation indicating whether a given value is available for each event or a numerical representation indicating the number of occurrences of the value.

- 2) *Modelling*: In the PPM workflow, this level does not include special steps that differ from the ones of the modeling stage of an ML pipeline. According to [2], [8], [9], a wide range of ML and deep learning models are selected based on the type of the PPM task. [9] obtained accurate results after applying boosting models in the outcome prediction task, while promising results were obtained by [8] after using LSTM models for predicting the remaining time of a process instance. After training the selected model on the encoded training subset of the event log, model performance is evaluated using the testing subset of the same event log. The accuracy of predictions can be measured with well-founded ML techniques, (e.g., AUC, F-score, MAE, and RMSE), and the choice of a specific technique depends on the type of prediction task. At the same time, earliness measures the length of a prefix at which the model achieves an acceptable level of accuracy, with a preference for shorter prefixes [8].

Event log preprocessing and modeling are performed in the offline stage of a PPM workflow. During this stage, completed process instances are used. In the online stage of a PPM workflow, in turn, incomplete process instances are considered. The latter are either obtained as a test subset of event logs or actually constitute running process instances. In the online stage, the corresponding bucket of the running process instance is determined, the instance is encoded according to the technique applied to other instances belonging to the same bucket, and finally, the encoded running instance serves as input to the trained model to generate a prediction.

B. XAI Application

With the emergence of XAI methods that address different explainability contexts and needs [10], [11], [12], [13], [14], a large number of studies investigate the benefits of predictions that are enhanced with explanations and to explore the limitations and pitfalls of the applied XAI methods [6], [15], [16], [17], [18]. According to [16], an explanation is *a means to describe the internals of a system or a model in a way understandable to humans (interpretability) while being accurate at the same time (completeness)*. In [15], explanations are defined as *the degree to which a human observer can understand the reason behind a decision or a prediction made by the model*.

Being able to provide accurate predictions has never been the sole aim of using ML-based systems. Other goals include safety, trust, nondiscrimination, and the capability of improvement. [19] argues that XAI provides a means to confirm the desiderata of ML applications. Corresponding criteria include fairness with respect to certain groups, the privacy of sensitive information within the data, and causality, which implies that the predicted change in output (due to a perturbation) will occur in the real system. To further pursue these goals, XAI methods are applied and placed on top of ML models. Explanations can be provided for single predictions (*local explanations*) or for the entire reasoning process of a predictive model (*global explanations*).

As a response to the wide adoption of XAI approaches, multiple XAI evaluation approaches have been proposed. The latter vary with respect to their goals, techniques, user experience levels, and scale. An evaluation method has a target characteristic against which the performance of an XAI method is evaluated. For example, there exist methods for evaluating an explanation for its *stability* [5], [20], *understandability* [3], *robustness* [21], [22], and *fidelity* [4]. Our proposed approach evaluates an XAI method with respect to the degree of its *consistency* with the underlying data. Consistency as we define it enables differentiating XAI methods regardless of the explained ML model, presuming that the model remains unchanged regarding its type and training parameters. Note that the consistency of an XAI method establishes a link between the evaluated XAI method and the original data from the evaluation perspective, apart from the fact that a global XAI method evaluates the model's understanding of the data, but not the data itself.

An evaluation approach has a specific *grounding*, i.e., the philosophy and the principal components an evaluation approach is based on [19] categorizes evaluation methods into the three grounding types: first, *functionally grounded* methods automatically measure some formal definition or characteristic of an explanation, i.e., *without any human participation*. Second, *human-based* evaluation methods depend on the presence of a *lay-person* who is provided with a simplified form of the task to assess the quality of an explanation without relying on the level of user experience in the domain. Third, *application-based* evaluations present the generated as-is explanations to an *involved practitioner* who is expected to have some domain knowledge and hence can provide key observations and improvement hints. Whenever humans are involved, there is

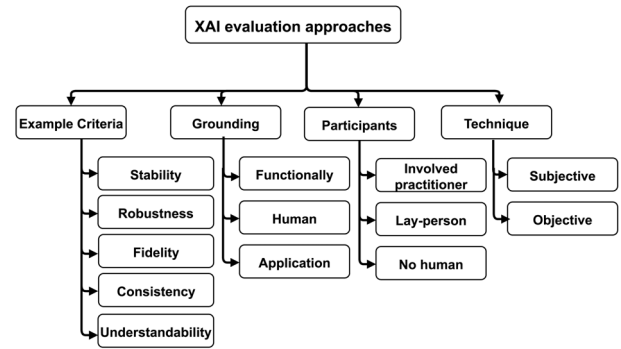


Fig. 1. Categories of XAI evaluation approaches.

a potential that the evaluation is done in a qualitative manner yielding *subjective* opinions on the XAI method. Whenever the evaluation method does not depend on humans, it is possible to apply a quantitative method in an *objective* manner. Fig. 1 summarizes the grounding for evaluating the XAI methods.

Due to space constraints, we do not discuss evaluation approaches in this article, other than in the form of related works (cf. Section VI). We refer interested readers to corresponding surveys on evaluation methods [23], [24]. Our approach is an objective functionally grounded approach that evaluates global XAI methods concerning their consistency based on the facts learned about the original data.

C. FS

An essential component of our approach is FS. FS is an implicit step performed when training a predictive model. Apart from model training, FS is regarded as an important data-cleaning procedure that is carried out to reduce data dimensionality. Furthermore, it enables extracting information that is expected to have high predictive power and hence can leverage the efficiency of the prediction model [25]. FS methods aim to remove irrelevant and redundant features that do not profoundly contribute to the generation of predictions. As an example consider a dataset with several highly correlated features. This correlation implies the cooccurrence of certain values of the features and, hence, the redundancy of information in the patterns learned by the predictive model. In this case, it is sufficient to have only one of the correlated features. On the other hand, a feature whose values are not correlated to prediction values is considered irrelevant as its presence does not lead to different predictions. FS methods fall into three categories [26].

- 1) *Wrapper-based feature selection (WFS)*: WFS methods use a classifier to choose the best subset of features that increase classifier performance. *Sequential selection* algorithms are the most common form of WFS methods. They start with an empty feature subset and incrementally add features one by one until a stopping condition is met (*forward selection*). Alternatively, WFS methods may start with the entire feature set and eliminate features one by one until the stopping condition is met (*backward selection*). While depending on a classifier and its performance as an objective function, there is a risk to the

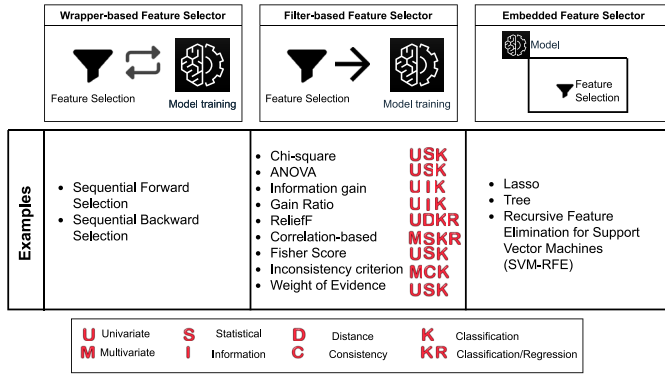


Fig. 2. FS methods.

validity of the selected feature subset since a classifier is prone to overfitting. Moreover, for each feature subset at each step, a new model is trained, and as a result, the execution process becomes computationally expensive.

- 2) *Filter-based feature selection (FSS)*: A suitable ranking criterion is used to order the features according to their prediction power and dependency on the target prediction. FSS methods are independent of the predictive model choice and are computationally inexpensive. However, FSS methods that analyze the predictive power of features independently from other features have the potential to disregard features that are important in combination with others.
- 3) *Embedded feature selection (EFS)*: Methods from this category use a classifier with a ranking function as an intrinsic part of its working mechanism, e.g., decision trees or linear models. However, being dependent on a classifier for choosing the most relevant feature subset also means being prone to any bias regarding the choices made by the respective classifier.

Fig. 2 provides various examples of methods falling into the three presented FS families. The FSS category contains most FS methods. FS methods may be characterized according to specific criteria [27], e.g.,

- 1) *Number of analysed features*. An FS technique may either be *uni* or *multivariate*. On one hand, univariate methods analyze the relevance of a feature in isolation from other features. Note that this complicates finding redundant features. On the other, multivariate methods analyze an entire feature subset [28].
- 2) *Ranking criteria*. The measurements or metrics used to evaluate the relevance of a given feature compared to others include, but are not limited to, the distance between samples, statistical analysis of the predictability power of feature values, information gain, and class consistency.
- 3) *Analysis goal*. Some filter methods are suitable to be used in the context of either classification or regression tasks, while others are suitable for both tasks.

FS methods enable reducing a highly dimensional dataset to its important features that are expected to be used by a predictive model. Subsequently, these methods are useful to the proposed

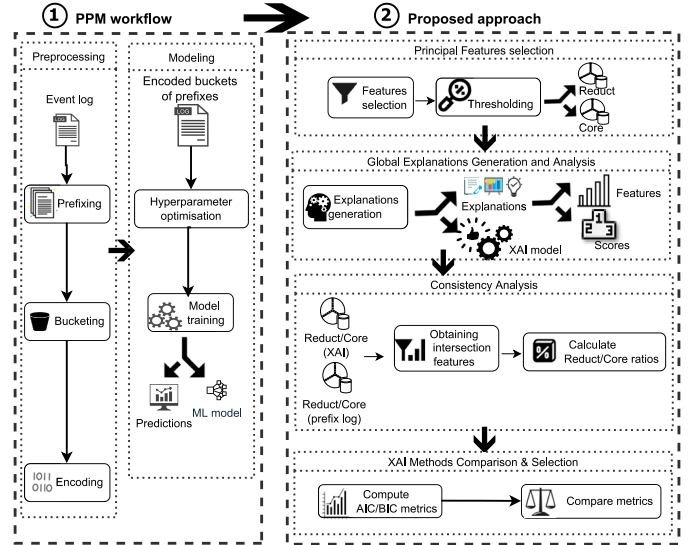


Fig. 3. Proposed XAI evaluation approach.

approach since they can be applied to reduce the input event log to the most crucial features.

III. PROPOSED XAI EVALUATION APPROACH

Our proposed evaluation approach has four phases that can be organized in a linear workflow. The first phase starts with the analysis of event logs to select the minimum set of features without which no reliable predictions would be possible. The second phase adds global XAI methods on top of modeling and prediction outcomes. The third phase then computes the metrics to measure the consistency percentage of outcomes from the previous phase. Finally, the fourth phase computes comparative metrics used to select an XAI method from a set of methods applied under the same experimental settings. The mechanism followed in the four phases of the approach necessitates their execution in a post hoc manner, i.e., after the modeling and prediction generation workflow is executed. An exception is the first phase, which solely depends only on the availability of preprocessed event logs, and hence can be interleaved within a PPM workflow. Fig. 3 shows the proposed evaluation approach preceded by the PPM workflow phases.

A. Phase 1: Principal Features Selection

This phase is the most crucial one to realize the goal of the approach, as its outcomes are supposed to reflect the ground truth extracted from the underlying data. With the term “ground truth” we mean basic data elements that can be crucial to generating the outcomes of a predictive model, and hence, are expected to be used by a predictive model and be reflected through the explanations of an XAI method. This phase is concerned with defining feature subsets that are relevant and nonredundant to the process of generating predictions. To identify the ground truth’s basic elements, we consider the following proposition:

Proposition 1: A finite set of features used by a predictive model can be abstracted by subsets of features that have a deterministic relation to the prediction to be generated.

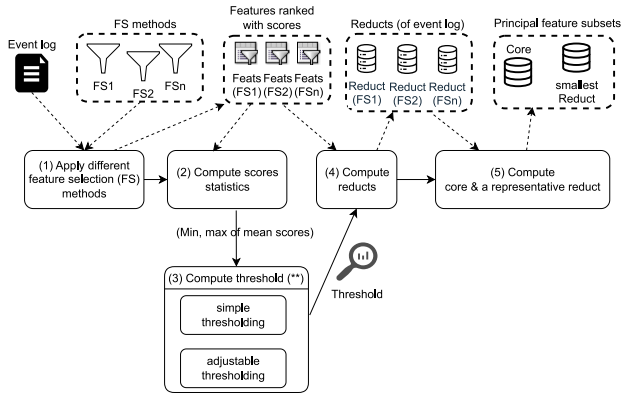


Fig. 4. Phase 1: principal features selection.

Kommiya Mothilal et al. [12] define *necessary feature subsets* as a minimal feature subset whose values have a coexistence relation to predictions. This implies that each change to feature values in the necessary subset will inevitably lead to a remarkable change in the prediction. We denote this subset as the *principal features*. In phase 1, an extraction process is executed to obtain principal features whose values have a deterministic role in achieving a prediction outcome. The obtained principal features are not mainly used to improve the performance of the applied ML model, as this phase is executed independently of the model.

In our approach, we differentiate between two types of principal features, i.e., *reduct* and *core*. A reduct is a *subset of features that are sufficient to define the basic concepts of the feature set*. A reduct represents a form of the feature set that is less redundant in terms of the information it represents about the event log. Consequently, multiple reducts may exist for a single feature set, depending on the reduct extraction method and the predictive power of the features in the subset together. A core is a *subset of the reduct, with features, being critical to a prediction task and cannot be eliminated when reducing features*. Features in the core have more predictive power compared to the remaining features, while excluded features are assumed to have less potential to improve the accuracy of a predictive model. In contrast, the accuracy of a predictive model might decline whenever a feature is removed from the core. The core is calculated by assuming the intersection of multiple reducts computed of a feature set using different FS techniques. The additional features existing in a reduct complement those in a core to improve the performance of a predictive model. Fig. 4 demonstrates the steps of phase 1. Furthermore, we provide a pseudocode for computing principal core sets in the supplementary materials (cf. Algorithm 1).

To analyze the entire feature set and to determine the predictive power of a certain feature in achieving the prediction target, we employ FS methods. Note that these methods are applied to the training subset of the event log. In phase 1, we employ different FS methods belonging to different FS categories and differing in the used techniques (cf. Fig. 4, step 1). The main criterion that qualifies an FS method to be applicable is its ability to return importance or relevance scores for the entire

Algorithm 1 Compute Adjustable Threshold

Input: Set of all $Reduct_{FS}$, $Threshold_{Old}$

Output: $ThresholdsList$

- 1: Compute mean of scores foreach $Reduct \in Reduct_{FS}$
 - 2: $MinScore = \min (MeanScores)$
 - 3: $MaxScore = \max (MeanScores)$
 - 4: $Interval = MaxScore - MinScore$
 - 5: *Divide the Interval into equal steps*
 - 6: $stepsCount = Num\ of\ steps\ in\ the\ interval$
 - 7: **for** s in range (1, $stepsCount$) : **do**
 - 8: $Threshold_{new} = Threshold_{Old} - (step * s)$
 - 9: $ThresholdsList.append(Threshold_{new})$
-

feature set, not just a predefined number of features. The independence of a specific ML model is another crucial criterion. Any FS method that has the potential to satisfy both criteria is a candidate for application in this phase.

Provided that the applied FS methods differ in their scoring techniques, the retrieved scores are normalized for the sake of comparability (cf. Fig. 4, step 2). Normalizing the scores results in all scores being within the range of 0,1. Note that the scores are shifted before being normalized in order to mitigate the effect of negative scores in case there are any. When shifting the scores, the absolute value of the minimum negative score is added to all the scores, in order to shift the whole data with the same amount before being normalized. The retrieved scores serve two important roles. The first is being a means to rank features according to their relevance to the prediction target. The second is to provide a threshold to define a reduct for each FS method based on feature scores provided by the applied FS method. As a result, each applied FS method will provide a reduct of the feature set (cf. Fig. 4, step 3).

As the applied FS methods rank the features instead of eliminating redundant ones, a mechanism is needed to reduce the number of features denoted as the reduct. Therefore, for each FS method, we derive a mean threshold based on the mean of scores provided by the method (cf. Fig. 4, step 2). To define a universal threshold that may be used across the scores of all FS methods, the minimum mean score is used as the threshold. Note that this scenario follows the “simple thresholding” substep in Fig. 4. Features scoring higher than the defined threshold are included in the reduct produced by the relevant FS method. Consequently, not all reducts are expected to be of the same size (cf. Fig. 4, step 4). Afterwards, the core set is the result of intersecting all reducts (cf. Fig. 4, step 5). The notion of a core and its calculation process are obtained from rough sets theory [29]

$$Core_{Eventlog} = \cap Reduct_{FS}$$

where $Reduct_{FS}$ is the set of all reducts obtained using the applied FS methods.

In our core notion, one of the most important conditions of a core is that there can be no empty core set, not under any given situation.

Proposition 2: An empty core set indicates that no feature holds valuable information for the predictive model, as a result of disagreement of FS methods.

To ensure that the reduct obtained by applying each FS method shares some knowledge with the other applied FS methods, the core set should not be empty. In turn, this condition ensures that there is obvious knowledge in the data that is to be learned by the ML model. This knowledge may further be reflected by an XAI method. Our goal is to capture this essential knowledge (in the form of the core set) and use it later in comparing several XAI methods in terms of their ability to communicate this knowledge.

In order to overcome an empty core output, whenever it occurs, the applied reduct selection threshold has to be less restrictive. Our approach is to define an adjustable threshold as illustrated in Algorithm 1, whenever a simple threshold leads to an empty core situation. Following step (2) in Fig. 4, the mean of scores obtained by each FS method is calculated. Our goal is to give more room for more features to be included in a reduct. To reach this goal, we define an interval under the previously applied threshold (line 4 in Algorithm 1). The defined interval is equal to the difference between the maximum and minimum mean scores and is divided into equal sections. The minimum threshold obtained should not be less than the old one by more than the defined interval. A new list of thresholds is computed by reducing the old threshold by a step or more. A new threshold is picked up randomly from the new list. The new threshold is not less than the old one by more than the value of the interval. A new set of reducts is computed based on the new threshold and subsequently a new core. This procedure continues until we obtain a nonempty core set.

After obtaining a nonempty core, the smallest reduct set is selected as the reduct representing the event log together with the core set, and both will be used in upcoming phases (cf. Fig. 4, step 5)

$$\text{Reduct}_{\text{Eventlog}} = \text{shortest}(\text{Reduct}_{\mathcal{FS}})$$

where $\text{Reduct}_{\mathcal{FS}}$ is the set of all reducts obtained using the applied FS methods. The shortest reduct is selected as it is guaranteed to contain the common features between all reducts, i.e., the core set, and at the same time represents the most important information contained in the entire feature set of the event log in a concise form. In Algorithm 1 (cf. the supplementary materials), we summarize the complete procedure followed to obtain a reduct and a core representing the principal features. According to [14], the achievement of some characteristics of an XAI method on a local scale does not imply their achievement in global explanations. As we compute the proposed principal feature subsets based on the whole data available in an event log, it is more consistent to study how they are reflected by global XAI methods.

B. Phase 2: Global Explanations Generation and Analysis

This phase is concerned with obtaining the artifacts to be evaluated. As we want to evaluate the consistency of XAI methods outcomes with respect to principal features, evaluated

XAI methods are expected to explain predictions in terms of features that influence a predictive model reasoning process. Consequently, XAI methods applied in this phase fall into the *feature attributions* category, i.e., they rank features based on their importance for making a prediction. As the proposed approach is meant to be a post hoc one, phase 2 is executed on top of a trained predictive model.

After explaining a predictive model with the selected XAI methods, each method returns a set of features along with their associated scores. However, these scores do not always represent the importance of a feature. Instead, each XAI method returns scores that represent a criterion used in the ranking mechanism of the method. For example, accumulated local effects (ALE) [10] returns scores that represent expected prediction change in response to changing the feature value. Another example is SHAP values, that are resulting after applying SHAP method, [13] represent the contribution of a feature value in reaching a specific prediction for a single process instance. A transformation technique is applied to convert feature scores into a form representing the importance of a feature in driving the reasoning process of the predictive model over the whole event log.

For example, we use entropy to obtain conclusive scores of the set of features after applying ALE. In the case of SHAP, scores are calculated for each process instance separately. The latter values should be scaled up to be representative on a global scale rather than locally to a specific process instance. Following the procedure defined by the author of SHAP [30], We aggregated scores of a specific feature over the whole event log into one score representing the contribution of such feature in the general prediction process. In permutation importance (perm) [18], the method returns importance scores indicating the average prediction error after shuffling the values of the feature for a number of predefined times.

The n - and m - top-ranked features are obtained from the features returned by each XAI method, where n and m are the sizes of the reduct and core obtained in phase 1. The extracted feature subsets are called $\text{reduct}_{\text{XAI}}$ and core_{XAI} , respectively, and they represent the principal features obtained using a certain XAI method. This phase concludes with a number of reducts and cores equivalent to the number of applied XAI methods.

C. Phase 3: Consistency Analysis

The main focus of phase 3 is to measure the percentage of consistency between the principal features and the important features concluded by each XAI method. In other words, the CR denotes the knowledge that an XAI method can reflect in the generated explanation when being applied to a specific ML model. Therefore, we introduce a metric to enable evaluate the degree to which an XAI method can reflect the ground truth found in the underlying data. The following equation shows how to compute the CR metric:

$$\text{CR} = \frac{\sum \text{Scores}(\text{Target}_{\text{XAI}} \cap \text{Target}_{\text{Eventlog}})}{\sum \text{Scores}(\text{Target}_{\text{Eventlog}})}. \quad (1)$$

Seeking shared knowledge requires obtaining the intersection between the principal features and the explanation generated

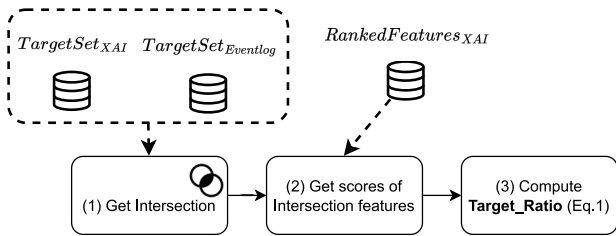


Fig. 5. Steps of phase 3.

by an XAI method. The *consistency_ratio* can be measured using either the *reduct_ratio* or the *core_ratio* depending on which CR shall be computed. Regarding the *reduct_ratio*, the reduct sets of the complete event log and the XAI method to be evaluated are used and, similarly, in the case of computing the *core_ratio*. Therefore, in (1), the notion Target is varying based on the target ratio to be computed. If the ratio to be computed is the *reduct_ratio*, then an intersection between the $Reduct_{XAI}$ and $Target_{Eventlog}$ is obtained, and likewise in the case of having the *core_ratio* as our Target. Scores of intersection features at the numerator part are the scores obtained by the XAI method. (1) is an adapted version of the recall equation applied to measure the fidelity of explanations in [14]. The introduced change in (1) is the usage of the features scores, not the features themselves. The proposed change enables differentiating XAI methods in case of having equally sized reducts or cores. Furthermore, using scores instead of the volume of the intersection set enables preserving the magnitude of features' influence even if different XAI methods have the same intersection features.

The summation of scores at the denominator is equated to the length of the target corresponding set representing the principal features of the event log. We choose the length to be the value of the denominator since this set represents the most relevant and least redundant feature subset. Therefore, each feature scores 1.0 in terms of its importance, where a score of 1.0 denotes the complete realization of the principal feature. The value of the CR lies between 0.0 and 1.0. The higher an XAI method scores in this measure is considered a better score. A high core ratio indicates the ability of an XAI method to reflect the most crucial factors contributing to the prediction process. At the same time, a high reduct ratio indicates the ability of an XAI method to reflect a high percentage of the features that define a form of the event log with the minimum redundant information. Fig. 5 shows the basic steps followed in phase 3.

An XAI method shall reflect the reasoning process of a predictive model. Therefore, whenever its outcomes are to be evaluated, it may be argued that this is an evaluation of the reasoning process itself. However, we can argue for the suitability of our approach to evaluate an XAI method. First, multiple XAI methods can be compared as long as the underlying settings of an experiment, i.e., all the PPM workflow choices regarding data, preprocessing, and modeling, are kept stable and unchangeable across different experiments. In this way, the only variable is the XAI method, so any difference in the evaluation results can be discussed in light of the properties of the applied XAI method. Second, we need to consider a studied pitfall of many

XAI methods, namely the instability of results under multiple executions of the same XAI method under the same conditions [20], [31]. This pitfall implies a potential for obtaining different ratios whenever the proposed approach is adopted to evaluate different executions of the same XAI method without changing the underlying settings.

D. Phase 4: XAI Methods Comparison and Selection

Individual ratios calculated in the previous phase represent indicators of the consistency level achieved by a certain XAI method. In phase 4, we introduce evaluation metrics that enable comparing several XAI methods based on the consistency of their conclusions about the important factors that affect the outcomes of a predictive model. The first proposed metric is inspired by Akaike information criterion (AIC) [32], which is designed to select a predictive model with minimal prediction error. However, we adapt the AIC to select an XAI method with minimal inconsistency with underlying data facts learned in phase 1 of this approach. In other words, the metric proposed in this phase selects an XAI method that maximizes the agreement with the ground truth. The proposed metric takes the form of the following equation:

$$AIC_{Consistency} = -2 * \log_2(\overline{CR}) + 2 * \overline{K}. \quad (2)$$

K is the number of features of the intersection feature subset. However, to compute $AIC_{Consistency}$ to favor the XAI method with minimal inconsistency, we use complements of CR and K . With a log function applied to the CRs, a small change in the values of the ratios introduces a remarkable difference in the resulting AIC values. In addition to $AIC_{Consistency}$, we propose another metric called $BIC_{Consistency}$. $BIC_{Consistency}$ is an adaptation of the Bayesian information criterion (BIC) [33], which is similar to AIC in its usage as a model selection metric, except that BIC penalizes complex ML models depending on more parameters. Again, we adapt the metric computations to be based on the complement of the number of intersection features. The following equation represents the form of $BIC_{Consistency}$:

$$BIC_{Consistency} = -2 * \log_2(\overline{CR}) + \overline{K} * \log_2(N). \quad (3)$$

N is the number of process instances contained in the event log under analysis. For an XAI method to be selected, it has to achieve low values of $AIC_{Consistency}$ and $BIC_{Consistency}$ metrics. The proposed metrics in this phase facilitate differentiating the target methods based on the tradeoff between the CR, (i.e., the method fitness function in this context), and the method coverage range (represented by the number of process instances and the number of features).

IV. EXPERIMENTS

To prove the applicability of the proposed approach, we performed experiments with different settings for different purposes.

A. Research Questions

In our pursuit of an approach for quantitatively evaluating global XAI methods in the context of PPM, we aim to measure consistency as a quality of the produced explanations, regardless of the executed ML model, despite considering the features that are important from the perspective of the model. However, in order to consider the influence of PPM workflow, we perform experiments to answer the following research questions:

- 1) *RQ1*: Is there an observable effect of using different PPM workflow settings on the results of the applied consistency metrics?

We need to examine the effect of underlying choices made in the context of the PPM workflow, e.g., preprocessing choices, on obtained results. We need to understand whether the underlying choices can propagate their characteristics to the applied XAI methods and whether this is discoverable through the performed experiments. For example, consider the situation when applying different preprocessing configurations and having all other choices stable. Obtaining distinguishable results in terms of consistency metrics can initiate further exploration of how preprocessing configurations can affect explainability results. With stable choices, we mean having the selected FS methods, ML model, and XAI method unchangeable during several experiments on the same event log when changing the preprocessing configuration at each experiment.

- 2) *RQ2*: How are the proposed metrics and ratios dependent on each other?

The proposed ratios are based on computations of importance scores provided by an XAI method. Furthermore, AIC and BIC metrics penalize/ favor an XAI method for different reasons. We want to understand how these claims are proven by experiments. We need to investigate the ability of both metrics to distinguish and compare the evaluated XAI methods. Furthermore, we need to study the factors that influenced the ability of the metrics to differentiate the XAI methods.

B. Settings

We conducted experiments on three real-life event logs which are publicly available from the 4TU Centre for Research Data [34]. We adopted the same labeling functions applied in [9] to classify each process instance into one of two classes, i.e., a binary classification task. The three basic event logs [34] used are as follows.

- 1) *Sepsis*. This event log belongs to the healthcare domain and reports on cases of patients with sepsis as a life-threatening condition. It further reports on their relevant diagnostic paths. Labeling of this event log defines whether the patient is admitted to the intensive care unit.
- 2) *Traffic fines*. This event log is a governmental one extracted from an Italian information system for managing road traffic fines. Hence, a process instance in this event log contains information about a fine, amount, and payment method. The event log is labeled to indicate whether a fine is fully paid or paid in installments.

TABLE II
STATISTICS OF THE THREE EVENT LOGS

Event Log	#Traces	#Max Prfx Len	%Pos Class	#Static Cols	#Dynamic Cols	#Cat Cols	#Num Cols	#Cat Levels Static Cols	#Cat Levels Dynamic Cols
Sepsis	776	60	14	24	13	28	14	76	39
Traffic fines	129615	20	45.5	4	14	13	11	54	173
BPIC2017	31413	180	41	3	20	12	13	6	682

- 3) *BPIC2017*. This event log documents the loan application process in a Dutch financial institution. The event log is labeled to identify whether a loan application is accepted.

Table II shows basic statistics of the event logs used in our experiments. We expect that differences may enable more variability in the results. Different data characteristics have implications for the characteristics of the preprocessed event logs. Consider, for example, the relation between the max prefix length of an event log and the number of prefix logs after applying prefix bucketing.

1) *Preprocessing*: As discussed in Section II-A, bucketing and encoding are necessary preprocessing steps that need to be performed to transform an event log into a format compatible with the requirements of ML models. According to the benchmarks available in [8], [9], in the PPM literature, only five bucketing techniques are available. Moreover, four encoding techniques exist that are applicable according to these benchmarks. We applied *single* and *prefix-based* bucketing techniques in association with aggregation and index encoding techniques, respectively. In single bucketing, all prefix traces are grouped within a single bucket, unlike in prefix-based bucketing when prefixes of traces are grouped based on their lengths [8]. We applied prefix-based bucketing with a gap of five. The latter setting means that for each prefix length of 1 to the maximum trace length (cf. Table II), separated by a gap of five activities, a separate bucket is created, prefixes of this length are grouped, and consequently, a separate ML model is trained. We apply a gap of five, as it is not desirable to overload the resulting logs with more features (in case of a shorter gap size) on one hand and to avoid losing valuable information about activities (in case of a longer gap size) on the other. As a result, we obtained three subevent logs from *sepsis* (lengths in 1, 6, and 11), two from *traffic fines* (lengths in 1 and 6), and four from *BPIC2017* (lengths in 1, 6, 11, and 16) after bucketing the prefixes based on their lengths. *Aggregation* and *index-based* techniques are selected as the encoding techniques. Both agree on the way static attributes are processed with numerical columns being encoded as-is, while one-hot encoding is used to encode categorical columns [9]. However, note that aggregation and index-based encoding differ in how they process the dynamic attributes of a process instance. In aggregation encoding [8], aggregation functions (e.g., sum and average) are applied on numerical columns, whereas either a boolean function (occurred or not) or a frequency-based function (number of times a value occurs) is applied to categorical columns. In contrast, in index-based encoding [8], numerical columns are encoded as-is, while a separate column is created for each subcategory in each column with a value of 0 or 1 indicating the absence/presence of this value along all process instances. Hence, *single-aggregation* and *prefix-index* preprocessing configurations are applied to the

TABLE III
SEARCH SPACES FOR HYPERPARAMETERS OF THE EXECUTED ML MODELS

ML Model	Hyperparameter	Search Space
Logit	Regularization (c)	$2^x, x \in [-5, 5]$
XGBoost	Learning rate	$x \in [0, 1]$
	Min child weight	$x \in [1, 6]$
	Subsample	$x \in [0.5, 1]$
	Max tree depth	$x \in [4, 30]$
	Colsample by tree	$x \in [0.5, 1]$
	n estimators	500

TABLE IV
AUC SCORES OF USED PREDICTIVE MODELS

		Sepsis			Traffic_Fines			BPIC2017		
XGBoost	Single_agg	0.91374			0.73918			0.86429		
	Prefix_index	1	6	11	1	6	1	6	11	16
		0.9273	0.90799	0.6061	0.4771	0.8259	0.5380	0.6644	0.8765	0.94365
Logit	Single_agg	0.8788			0.7949			0.8244		
	Prefix_index	1	6	11	1	6	1	6	11	16
		0.9093	0.9185	0.8292	0.55499	0.8011	0.54597	0.7144	0.8753	0.9351

Categorized according to preprocessing configurations used.

event logs. Our choices of preprocessing configurations and predictive models are influenced by the best-performing configurations as reported in [9]. Furthermore, we choose *single-aggregation* and *prefix-index* configurations as they have the least information lossy techniques (i.e., index encoding) or the most comprehensive techniques that enable the input of various sizes of prefixes to the same predictive model (i.e., single bucketing). We use both configurations on the *sepsis*, *traffic fines*, and *BPIC2017* event logs.

2) *Predictive Models*: As ML models, we selected logit [35] and XGboost [36]. The former is selected as it is simple and interpretable, whereas the latter is selected due to its outstanding performance, as reported by many studies [9]. We optimize the hyperparameters of both models using the TPE algorithm. We perform three-fold cross-validation to choose the best-performing hyperparameters. Table III represents the search space of the hyperparameters of each model.

After training the selected ML models on 80% of each event log as the training set, the two models achieved reasonable AUC scores (see Table IV). Note that, AUC is an accuracy measure used to differentiate classifiers in terms of how well the negative and positive classes are separated for the decision index [37]. The higher the AUC score becomes, the better a classifier is.

3) *FS Methods*: As for the chosen FS methods, we selected seven methods that meet certain criteria. The ability of the method to rank all features based on their predictive power, rather than returning a predefined number of selected features is an important selection criterion. Furthermore, the selected FS methods are not biased toward a certain ML model. Hence, WFS methods are excluded from the selected FS methods. We picked methods that accept a prefit ML model, whenever inputting an ML model constitutes a prerequisite. The latter criterion ensures that the output feature subset will be aligned with the patterns and internal analysis made by the ML model

when generating predictions. To adhere to these prerequisites, as embedded methods, we selected lasso and tree implemented in the Scikit-learn library. We input the prefit models specified in the previous subsection as inputs to lasso and tree. This procedure ensures that the ground truth obtained after applying these two FS methods reflects important features according to the ML models used in the prediction process. Furthermore, by using the pretrained models, we use the same settings the model used in ranking the features during the training process. For example, in the case of tree selector, the input model is XGBoost with the feature importance setting being set to the default, i.e., *gain*. From the filter methods category, we selected Information gain [38], gini-index [38], TuRF (as one of the ReliefF versions) [39], information value (IV) [40], and chi-square [41] and ANOVA [42] interchangeably based on the underlying nature of features.

4) *XAI Methods*: Our proposed approach aims to evaluate model-agnostic XAI methods. Therefore, we selected three different XAI methods of this category to rank important features in the context of the performed experiments. The selected methods can provide a list of features together with scores indicating their rank. The selected XAI methods are SHAP [13], perm [18], and ALE [10]. SHAP is based on game theory and computes Shapely values by assuming the presence/absence of a player in a game and examining all possible settings to evaluate players' (features) contributions to the achieved score (prediction). Perm calculates feature importance by shuffling values of a feature and monitoring the effect of this change on the generated predictions. ALE divides the values of a feature into quantiles. Based on the conditional distribution of the feature, the difference in predictions is analyzed for samples with similar values of the feature.

5) *Hardware and Software Tools*: All experiments were run using Python 3.6 and the scikit-learn library on a 96-core Intel(R) Xeon(R) Platinum 8268 @2.90GHz with 768GB of RAM. The code of executed experiments, in addition to execution results, are available through our Github repository¹ to enable open access for interested practitioners.

V. OBSERVATIONS AND DISCUSSION

After applying the approach proposed in Section III, within experiments designed as specified in Section IV, we could make several observations that we trace back to their influencing factors in the following.

A. Observations

We obtained six values for each event log after applying the proposed approach to the defined experiments. These values are *reduct ratio* and *core ratio*, and the *AIC* and *BIC* values corresponding to each of these ratios. Detailed results are reported in Table V(a)–V(l) and plotted in Fig. 2 in the supplementary materials. Table V(a)–V(i) reports on the result corresponding to each event log preprocessed with *prefix-index* configuration.

¹https://github.com/GhadaElkhawaga/XAI_predictivemonitoring_Consistency.git

When considering the number of features in core intersection sets, we observe the superiority of SHAP when explaining XGBoost predictions. In turn, perm started to obtain the highest number of features in event logs with longer prefixes of *BPIC2017*, when explaining logit predictions [cf. Table V(b)–V(d)].

In most cases, ALE is scoring the highest reduct ratios, despite not having the highest number of features in the respective intersection set. This observation can be explained by having high entropy values associated with the features in the intersection set. Note that such high entropy values compensate for the lower number of features when computing the respective ratio. Consequently, ALE could beat SHAP, which has the highest number of features.

As stated in phase 4, AIC tends to penalize the method with minimal inconsistency, i.e., it selects the method with the highest ratio. Consequently, XAI methods scoring the highest reduct/core ratios are expected to have the lowest AIC value.

Observation (3). Unexpectedly, we observed no relation between reduct/core ratios and their respective AIC values. Observation 3 is not valid in few cases. As an example, consider ALE results on *sepsis* with prefix length = 6 [cf. Table V(h)]. In all obtained results, including the latter exception, AIC values are the lowest for XAI methods with the highest number of features in the intersection sets.

Observation (4). BIC metric is expected to penalize methods with the minimum number of features in the intersection sets. This implies selecting the XAI method that captures the largest amount of ground truth, i.e., the XAI method with the highest number of intersection features. This hypothesis is met by the results of all XAI methods on all event logs under all performed experimental settings. Interestingly, *observations 2, 3, and 4* are tightly related to answer *RQ2*.

Observation (5). The empty intersection core set is an interesting observation made in results associated with *sepsis* and *traffic fines* event logs when preprocessed using *single-aggregation* configuration. These observations are made clearer in their respective plots [cf. Fig. 1(h) and 1(i) in the supplementary materials]. In the case of *traffic fines* event log, it has relevantly shorter prefix lengths (cf. Table II). Shorter prefixes, and consequently fewer features, provide lower chances for an XAI method to achieve an intersection with the underlying data. The same case does not hold in other event logs with longer prefixes, e.g., *BPIC2017* preprocessed using *single-aggregation* configuration and does not apply to *traffic fines* prefixes that are preprocessed using *prefix-index* configuration.

B. Discussion

1) *Volume of Features in the Intersection Sets:* As shown in Fig. 1(a)–1(l) (in the supplementary materials) and as stated in the last two rows of each table from Table V, the number

TABLE VI
CALCULATED RATIOS

(a) Traffic Fines (Prefix Len=1)

Measurement	Logit_SHAP	Logit_perm	Logit_ALE
Reduct_ratio	0.52941	0.23529	0.23529
AIC_reduct	34.17493	52.77404	52.77404
BIC_reduct	268.75714	433.97015	433.97015
#Feats_reduct_intersection	18	8	8

(b) Traffic Fines (Prefix Len=6)

Measurement	Logit_SHAP	Logit_perm	Logit_ALE
Core_ratio	0.08333	0.08333	0.08333
AIC_core	44.25106	44.25106	44.25106
BIC_core	288.29172	288.29172	288.29172
#Feats_core_intersection	2	2	2

(c) Sepsis (Prefix Len=1)

Measurement	XGBoost_SHAP	XGBoost_perm	XGBoost_ALE
Core_ratio	0.33333	0.22222	0.22222
AIC_core	13.16993	14.72514	14.72514
BIC_core	56.82667	65.65801	65.65801
#Feats_core_intersection	3	2	2

Based on numbers of intersection features.

of features in the intersection sets of XAI methods that explain predictions of XGBoost are relatively higher than their counterparts of logit, regardless of the applied preprocessing configuration. Furthermore, the number of features increases as the prefix length increases in event logs preprocessed using *prefix-index* configuration. In [31], the authors run experiments to explore the data characteristics of the three event logs used in this article. High multicollinearity between the features is discovered in event logs that were preprocessed using *prefix-index* configuration, while not being completely absent from *single-aggregation* preprocessed event logs. *Sepsis* event log shows high multicollinearity in general, approaching complete collinearity, regardless of the preprocessing configuration applied. XGBoost is not supposed to be affected by multicollinearity. In boosting-based models like XGBoost, whenever collinearity exists between a subset of the features, the model chooses one feature as the data splitting criterion, to which it assigns the entire importance score [36]. In contrast, logit assigns similar coefficients to collinear features. With the philosophy of applied FS methods in phase 1 being to reduce irrelevant and redundant features, we find XAI methods built on top of XGBoost models can capture higher numbers of principal features compared to their counterparts built on top of logit models.

According to (1), large intersection sets are effective in the case when the XAI method assigns high scores to features in these sets. High scores enable obtaining high reduct/core ratios, especially in the case of an event log with a small reduct/core. The number of features alone cannot be used as a distinguishing factor between XAI methods applied to predictions of the same predictive model. As shown in Table VI(a)–VI(c), we tried to calculate reduct/core ratios solely based on the number of features in the intersection set of each XAI method, rather than their respective scores. All results shown in Table VI indicate the inability to distinguish XAI methods whenever the number of features at the intersection set is the same. The observed effects of using the scores of features on the obtained ratio values

contribute partially to answer *RQ2*. As stated by observation (5), results associated with *traffic fines* and *sepsis* event logs preprocessed using *single-aggregation* configuration indicate empty core sets [cf. Table V(g)–V(i), V(k), V(l), and Fig. 1(h) and 1(i) in the supplementary materials], except in few cases of the evaluated XAI methods. These empty sets might be the result of having shorter prefixes in case of *traffic fines* event log. These shorter prefixes provide less number of features after preprocessing using *single-aggregation* configuration. Having fewer features subsequently results in fewer chances for an XAI method to achieve an intersection with the underlying data, especially when taking into account the random effect of XAI methods that apply shuffling, e.g., permutation importance. Another factor that might influence this phenomenon is the class imbalance in *sepsis* as indicated in Table II. Despite having relatively high-performing predictive models as indicated in Table IV, the XAI methods used to query these models for the most important features are unable to make conclusions about features that are responsible for the generated predictions, as the model itself is not able to learn obvious decisive patterns from data. Consequently, important features as concluded by the XAI methods in this case are not aligned with important and principal features as concluded by the FS methods.

2) *Effect of Preprocessing Configurations on Explanations Consistency*: The number of process instances for which predictions are generated is considered a secondary factor in our analysis. While it is expected to have a direct effect on the predictive model accuracy, or in the process of evaluating the predictive power of a feature, it does not have a direct effect on the XAI method. Therefore, we do not observe any effect of the bucketing technique on the resulting ratios and metrics values. The encoding technique used, i.e., whether index or aggregation, affects the resulting feature vector size, especially in the case of event logs with a remarkable imbalance in the number of categorical versus numerical attributes. As discussed in [9], aggregation encoding performs aggregations on attributes' columns to summarize them in terms of aggregation functions (e.g., sum, average, etc.) applied on numerical attributes or frequency of occurrence or boolean functions applied on categorical attributes. Furthermore, in index encoding, a separate column is created for each value in each attribute column associated with each event in a given process instance. The latter encoding can result in a dimensionality explosion as the number of columns increases as the number of values in an attribute differs with each process instance. However, from Fig. 2(c)–2(e) (supplementary materials), it may be concluded that the reduct/core ratios decrease as the prefix length increases. The only exception is observed in ALE reduct ratios on both ML models in *BPIC2017* [cf. Fig. 2(c) in the supplementary materials]. At the same time, the number of features at the intersection sets does not decrease as the prefix length increases [cf. Table V(a)–V(i)]. This may indicate that the difference between scores associated with the top n -features shrinks, and features become similar in the scores concluded by each XAI method. Again, having almost similar importance scores can justify the increasing AIC scores as the number of intersection features increases. The observed increase is not

accompanied by an increase in the reduct ratio (for example in the case of ALE when it has a high reduct ratio based on a small number of intersection features with high importance scores). The concluded relation between importance scores and AIC scores contributes to the answer of *RQ2*. The effect of the applied encoding technique together with the used XAI method can be observed in ALE results. Despite scoring well in terms of reduct ratios, ALE, as indicated before, does not score well in terms of other metrics, nor in terms of the number of features at the intersection sets. This observation can be explained in terms of the encoding technique which increases the number of categorical columns. Computing ALE effects for categorical attributes can be criticized for being inaccurate since values of these features do not maintain order [18], [31]. This discussion concludes the answer of *RQ1*.

VI. RELATED WORK

Several approaches were proposed over the past few years to apply explainability as a complementary part of the PPM workflow. Explainability is integrated into the PPM workflow to increase user trust in the generated predictions of a business process. While models based on deep learning get more complex as their prediction accuracy increases, several approaches are introduced to make the outcomes of these models more understandable. To predict the next activity transparently, [43] uses weights generated in attention-based neural networks to highlight different factors contributing to reaching a prediction. [43] examines the applicability of their proposal through training three different LSTM-based attention models to predict the next activity and the next activity associated with its executing resource and the remaining time till the end. Using the three different models, [43] could highlight factors contributing to reaching the prediction at different levels of detail, i.e., interpreting using influencing events only, or using influencing events and their associated event attributes.

[44] proposes an approach based on integrating layerwise relevance propagation (LRP) [11] into an LSTM-based model used to predict the next activity. In a related context, [45] uses gated graph neural networks (GGNN) to predict the outcome of a running process instance in addition to a relevance score corresponding to each of the activities preceding the predicted outcome. In the same direction, [46] proposes an approach based on explaining the outcome of a process instance in terms of If – Then rules learned by a neuro-fuzzy network and output in human-interpretable form. In an attempt to tailor XAI methods to suit the specific nature of PPM event logs, [47] proposes an approach to generate counterfactual explanations. The latter are generated using genetic algorithms while taking process constraints into account in order to ensure the generation of realistic process instances.

With a direct application of XAI methods to understand the attributes contributing to different prediction tasks, i.e., remaining time, activity occurrence, and case total cost, [48] applied SHAP [13] to explain local predictions generated by an LSTM model. In [31], [49], different levels and types of XAI methods are examined to discover how data characteristics

and underlying model sensitivities could be propagated and highlighted in generated explanations. Building on the ability of post hoc XAI methods to highlight features contributing to a prediction, [50] uses LIME [14] and SHAP to identify features contributing to false predictions. Knowing the features affecting the accuracy of predictions, [50] shuffles the values of these features. Furthermore, [50] retrains the model on the new event log in order to neutralize patterns constituted of these features and hence improve the accuracy of the predictive model.

Without the direct application of out-of-the-box XAI methods, [51] leverages the idea of a decompositional explanation. The proposal in [51] is based on decomposing the prediction (in this case, cycle time) into a weighted sum of the predicted cycle times of activities to be performed till the end of the running process instance. To achieve this, [51] uses two ML models, one for predicting the cycle time of each activity, and another one to predict the branching possibilities of each decision point. This idea is enabled by using *flow analysis* techniques and [51] argues that it is a white-box interpretable one.

Unfortunately, only a few proposals exist that are concerned with proposing evaluation approaches for XAI methods when applied to PPM results. However, as more attention is drawn to the importance of explaining PPM results, it is expected that evaluating XAI methods will gain more interest. For example, [4] proposes an approach to evaluate local XAI methods for their fidelity, i.e., the ability of the XAI method to mimic the behavior of the explained ML model in the vicinity of the explained process instance. However, in this approach, the authors evaluate the internal fidelity of the XAI method, i.e., the similarity between the decision-making process of the explainer proxy model and the explained complex black box model, rather than the similarity between the decisions made by the two models. Perturbation of feature values based on a newly generated uniform distribution is applied to replace the current feature vector of the explained process instance with a new one. For the new feature vector, new predictions are generated and the error in predictions is considered the fidelity measure of the proxy model created by the XAI method.

[5] introduces another evaluation approach to measure the stability of generated local explanations. According to [20], stability of explanations means the similarity between explanations generated for the same data sample under the same conditions. [5] proposes and applies two metrics to evaluate the stability of the top-K feature subset, and their relevant weights after explaining predictions for certain process instances multiple times. [52] introduces four out-of-the-box metrics that are imported from relevant XAI evaluation research. The metrics are applied the same way to process mining-related data attributes. Different attributes in process mining data should be studied separately as they differ in their characteristics, and subsequently in the magnitude of their effect on the generated predictions and explanations. [3] conducts experiments that are more user-oriented in evaluating XAI methods, and studying whether the resulting explanations are understandable and how effective they are in the decision-making process. Participants in user evaluations carried out in this study are working in the PPM field and others from the ML field. This study concludes

that comprehension and usage levels of these explanations varied among participants based on their domain knowledge and experience. While quantitative evaluations confirm technical characteristics or requirements of explanations, qualitative studies are needed to confirm the usefulness of explanations in achieving the goals they are generated for.

VII. CONCLUSION

In this article, we propose an approach for evaluating global model-agnostic XAI methods that use feature attributions to explain the reasoning process of an ML model. Our goal is to evaluate these XAI methods with respect to how consistent their explanations are with the basic concepts extracted from the underlying data. Using experiments on real-life predictive monitoring event logs, we provide a functionally grounded evaluation that was able to uncover the effect the applied preprocessing configurations have on the generated explanations. Furthermore, we could identify the way the sensitivities of a predictive model can be reflected in generated explanations. We could uncover how these sensitivities have the potential to affect the conformance of the explanations to ground truth extracted from the underlying data.

Our approach came to its limits whenever a large event log was used. However, in the future, we plan to extend our proposal to include multivariate feature analysis in order to study the effect of feature interactions. In addition, we plan to perform more experiments using more choices of preprocessing configurations and ML models. To this point, we proposed a flexible framework that can be extended to evaluate any model-agnostic global XAI method.

REFERENCES

- [1] W. M. P. van der Aalst, "Process mining: A 360 degree overview," in *Process Mining Handbook (LNBIP)*. New York, NY, USA: Springer-Verlag, 2022, pp. 3–34.
- [2] A. E. Márquez-Chamorro, M. Resinas, and A. Ruiz-Cortés, "Predictive monitoring of business processes: A survey," *IEEE Trans. Services Comput.*, vol. 11, no. 6, pp. 962–977, Nov./Dec. 2018.
- [3] W. Rizziet al., "Explainable predictive process monitoring: A user evaluation," 2022, *arXiv:2202.07760*.
- [4] M. Velmurugan, C. Ouyang, C. Moreira, and R. Sindhgatta, "Evaluating fidelity of explainable methods for predictive process analytics," in *Proc. Lecture Notes Bus. Inf. Process., Intell. Inf. Syst.*, vol. 424, S. N. A. Korthaus, Ed., New York, NY, USA: Springer-Verlag, 2021, pp. 64–72.
- [5] M. Velmurugan, C. Ouyang, C. Moreira, and R. Sindhgatta, "Evaluating stability of post-hoc explanations for business process predictions," in *Proc. Int. Conf. Service-Oriented Comput.*, H. Hacid, O. Kao, M. Mecella, N. Moha, and H.-y. Paik, Eds., Cham, Switzerland: Springer-Verlag, 2021, pp. 49–64.
- [6] D. V. Carvalho, E. M. Pereira, and J. S. Cardoso, "Machine learning interpretability: A survey on methods and metrics," *Electronics*, vol. 8, no. 8, 2019, Art. no. 832.
- [7] R. Wilming, C. Budding, K.-R. Müller, and S. Haufe, "Scrutinizing XAI using linear ground-truth data with suppressor variables," *Mach. Learn.*, vol. 111, no. 5, pp. 1903–1923, 2022.
- [8] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, and I. Teinemaa, "Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring," *ACM Trans. Intell. Syst. Technol.*, vol. 10, pp. 1–34, Jul. 2019.
- [9] I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi, "Outcome-oriented predictive process monitoring: Review benchmark," *ACM Trans. Knowl. Discov. Data*, vol. 13, pp. 1–57, Mar. 2019.
- [10] D. Apley and J. Zhu, "Visualizing the effects of predictor variables in black box supervised learning models," *J. Roy. Statist. Soc. B*, vol. 82, no. 4, pp. 1059–1086, 2020.

- [11] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, "Layer-wise relevance propagation for neural networks with local renormalization layers," in *Proc. Artif. Neural Netw. Mach. Learn. (ICANN) (LNCS)*, vol. 9887, Cham, Switzerland: Springer-Verlag, 2016, pp. 63–71.
- [12] R. Kommiya Mothilal, D. Mahajan, C. Tan, and A. Sharma, "Towards unifying feature attribution and counterfactual explanations: Different means to the same end," in *Proc. 2021 AAAI/ACM Conf. AI, Ethics, and Soc.*, New York, NY, USA: ACM, 2021, pp. 652–663.
- [13] S. Lundberg and S. Lee, "A unified approach to interpreting model predictions," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 4768–4777.
- [14] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?" in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, B. Krishnapuram, M. Shah, A. Smola, C. Aggarwal, D. Shen, and R. Rastogi, Eds., New York, NY, USA: ACM, 2016, pp. 1135–1144.
- [15] A. B. Arrieta et al., "Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, no. 3, pp. 82–115, 2020.
- [16] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. Data Sci. Adv. Analytics (DSAA)*, Piscataway, NJ, USA: IEEE Press, 2018, pp. 80–89.
- [17] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–42, 2019.
- [18] C. Molnar, *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. GitHub, 2020. Accessed: Feb. 2, 2024. [Online]. Available: <https://christophm.github.io/interpretable-ml-book/>
- [19] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," 2017, *arXiv:1702.08608v2*.
- [20] G. Visani, E. Bagli, F. Chesani, A. Poluzzi, and D. Capuzzo, "Statistical stability indices for time: Obtaining reliable explanations for machine learning models," *J. Oper. Res. Soc.*, vol. 73, no. 1, pp. 91–101, 2021.
- [21] D. Alvarez-Melis and T. S. Jaakkola, "On the robustness of interpretability methods," 2018, *arXiv:1806.08049v1*.
- [22] C. Hsieh et al., "Evaluations and methods for explanation through robustness analysis," in *Proc. 9th Int. Conf. Learn. Representations (ICLR)*, 2021.
- [23] S. Mohseni, N. Zarei, and E. D. Ragan, "A multidisciplinary survey and framework for design and evaluation of explainable AI systems," *ACM Trans. Interact. Intell. Syst.*, vol. 11, nos. 3–4, pp. 1–45, 2021.
- [24] J. Zhou, A. H. Gandomi, F. Chen, and A. Holzinger, "Evaluating the quality of machine learning explanations: A survey on methods and metrics," *Electronics*, vol. 10, no. 5, 2021, Art. no. 593.
- [25] A. O. Balogun et al., "Impact of feature selection methods on the predictive performance of software defect prediction models: An extensive empirical study," *Symmetry*, vol. 12, no. 7, 2020, Art. no. 1147.
- [26] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Comput. Elect. Eng.*, vol. 40, no. 1, pp. 16–28, 2014.
- [27] A. Jovic, K. Brkic, and N. Bogunovic, "A review of feature selection methods with applications," in *Proc. 38th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, pp. 483–519.
- [28] V. Bolón-Canedo, N. Sánchez-Marroño, and A. Alonso-Betanzos, "A review of feature selection methods on synthetic data," *Knowl. Inf. Syst.*, vol. 34, no. 3, pp. 483–519, 2013.
- [29] Z. Pawlak, *Rough Sets: Theoretical Aspects of Reasoning About Data* (Theory and Decision Library), vol. 9. Dordrecht, The Netherlands: Springer-Verlag, 1991.
- [30] S. Lundberg, "SHAP issues: How to extract the most important feature names?" 2019. Accessed: Jan. 17, 2024. [Online]. Available: <https://github.com/slundberg/shap/issues/632>
- [31] G. Elkhawaga, M. Abuelkheir, and M. Reichert, "XAI in the context of predictive process monitoring: An empirical analysis framework," *Algorithms*, vol. 15, no. 6, 2022.
- [32] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, Dec. 1974.
- [33] S. I. Vrieze, "Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC)," *Psychological Methods*, vol. 17, no. 2, pp. 228–243, 2012.
- [34] "4TU. Centre for Research Data," Accessed: Jan. 17, 2024. [Online]. Available: <https://data.4tu.nl/>
- [35] M. Maalouf, "Logistic regression in data analysis: An overview," *Int. J. Data Anal. Techn. Strategies*, vol. 3, no. 3, pp. 281–299, 2011.
- [36] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, New York, NY, USA: ACM, pp. 785–794.
- [37] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.
- [38] L. E. Raileanu and K. Stoffel, "Theoretical comparison between the Gini index and information gain criteria," *Ann. Math. Artif. Intell.*, vol. 41, no. 1, pp. 77–93, 2004.
- [39] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *J. Biomed. Inform.*, vol. 85, pp. 189–203, 2018.
- [40] E. Zdravevski, P. Lameski, and A. Kulakov, "Weight of evidence as a tool for attribute transformation in the preprocessing stage of supervised learning algorithms," in *Proc. Int. Joint Conf. Neural Netw.*, Piscataway, NJ, USA: IEEE Press, 2011, pp. 181–188.
- [41] R. Cao, W. González Manteiga, and J. Romo, *Nonparametric Statistics*, vol. 175. Cham, Switzerland: Springer-Verlag, 2016.
- [42] H. R. Lindman, *Analysis of Variance in Experimental Design* (Springer Texts in Statistics). New York, NY, USA: Springer-Verlag, 1992.
- [43] R. Sindhgatta, C. Moreira, C. Ouyang, and A. Barros, "Exploring interpretable predictive models for business processes," in *Proc. Bus. Process Manage.: 18th Int. Conf. (BPM)*, Seville, Spain, Germany: Springer-Verlag, Sep. 13–18, 2020, pp. 257–272.
- [44] S. Weinzierl, S. Zilker, J. Brunk, K. Revoredo, M. Matzner, and J. Becker, "XNAP: Making LSTM-based next activity predictions explainable by using LRP," 2020, *arXiv:2008.07993*.
- [45] M. Harl, S. Weinzierl, M. Stierle, and M. Matzner, "Explainable predictive business process monitoring using gated graph neural networks," *J. Decis. Syst.*, vol. 29, no. sup1, pp. 312–327, 2020.
- [46] V. Pasquadibisceglie, G. Castellano, A. Appice, and D. Malerba, "FOX: A neuro-fuzzy model for process outcome prediction and explanation," in *Proc. 3rd Int. Conf. Process Mining (ICPM)*, 2021, pp. 112–119.
- [47] T.-H. Huang, A. Metzger, and K. Pohl, "Counterfactual explanations for predictive business process monitoring," in *Proc. Eur. Mediterranean Middle Eastern Conf. Inf. Syst.*, M. Themistocleous and M. Papadaki, Eds., Cham, Switzerland: Springer-Verlag, 2022, pp. 399–413.
- [48] R. Galanti, B. Coma-Puig, M. de Leoni, J. Carmona, and N. Navarin, "Explainable predictive process monitoring," in *Proc. 2nd Int. Conf. Process Mining (ICPM)*, 2020, pp. 1–8.
- [49] G. Elkhawaga, M. Abuelkheir, and M. Reichert, "Explainability of predictive process monitoring results: Can you see my data issues?" *Appl. Sci.*, vol. 12, no. 16, 2022.
- [50] W. Rizzi, C. Di Francescomarino, and F. M. Maggi, "Explainability in predictive process monitoring: When understanding helps improving," in *Business Process Management Forum*, D. Fahland, C. Ghidini, J. Becker, and M. Dumas, Eds., Cham, Switzerland: Springer-Verlag, 2020, pp. 141–158.
- [51] I. Verenich, M. Dumas, M. La Rosa, and H. Nguyen, "Predicting process performance: A white-box approach based on process models," *J. Softw. Evolution Process*, vol. 31, no. 6, 2019.
- [52] A. Stevens and J. De Smedt, "Explainability in process outcome prediction: Guidelines to obtain interpretable and faithful models," *Eur. J. Oper. Res.*, 2023.