

# Explainable AI Enabled Inspection of Business Process Prediction Models

Chun Ouyang · Renuka Sindhgatta ·  
Catarina Moreira

Received: date / Accepted: date

**Abstract** Modern data analytics underpinned by machine learning techniques has become a key enabler to the automation of data-led decision making. As an important branch of state-of-the-art data analytics, business process predictions are also faced with a challenge in regard to the lack of explanation to the reasoning and decision by the underlying ‘black-box’ prediction models. With the development of interpretable machine learning techniques, explanations can be generated for a black-box model, making it possible for (human) users to access the reasoning behind machine learned predictions. In this paper, we aim to present an approach that allows us to use model explanations to investigate certain reasoning applied by machine learned predictions and detect potential issues with the underlying methods thus enhancing trust in business process prediction models. A novel contribution of our approach is the proposal of model inspection that leverages both the explanations generated by interpretable machine learning mechanisms and the contextual or domain knowledge extracted from event logs that record historical process execution. Findings drawn from this work are expected to serve as a key input to developing model reliability metrics and evaluation in the context of business process predictions.

**Keywords** business process prediction · interpretable machine learning · explainable AI · predictive process monitoring · event logs

## 1 Introduction

Business processes form a lifeline of business within and across organisations. Executions of day-to-day business processes involve a wide range of stakeholders and are supported by a variety of information systems. Data generated by

---

C. Ouyang, R. Sindhgatta, C. Moreira  
Queensland University of Technology, Brisbane, Australia  
E-mail: {c.ouyang,renuka.sr,catarina.pintomoreira}@qut.edu.au

process executions are stored in event logs. In recent years, machine learning techniques are being applied to construct *business process prediction models*. These models aim at predicting future states of a business process by learning from process execution history recorded in event log data. Typical examples of business process predictions include predicting outcomes that the execution of a business process may lead to (Teinemaa et al., 2019) and the remaining time till the completion of process execution (Verenich et al., 2019).

However, similarly to other state-of-the-art data analytics, business process predictions are also faced with a challenge in regard to the lack of explanation to the reasoning and decision of its prediction models. While existing studies focus on applying advanced, complex learning techniques that can achieve high accuracy of process predictions, these techniques often have sophisticated internal computations that are largely recalcitrant to explanation and thus are often applied and recognised as a ‘black-box’.

The recent body of literature in machine learning has emphasised the need to understand and trust the predictions (e.g., Lakkaraju et al. (2019); Rudin (2019)), and methods and techniques have been proposed for explaining black-box models. These are known as *interpretable machine learning* or, in a broader context, *explainable AI* (XAI). In particular, explanations from a black-box model make it possible for (human) users to confirm multiple prerequisites of a machine learning model (Doshi-Velez and Kim, 2017), which are fairness, reliability and robustness, causality, privacy and trust. In their widely-cited survey of XAI methods, Guidotti et al. (2018) point out that the “applications in which black box decision systems can be used are various, and each approach is typically developed to provide a solution for a specific problem”.

In this paper, we aim to present an approach that allows us to use model explanations to investigate certain reasoning applied by machine learned predictions and detect potential issues with the underlying methods thus enhancing trust in business process prediction models. On the one hand, it is challenging to generate explanations of a black-box model which takes as input the event log data that are complex and capture multi-dimensional (temporal) sequences of process execution. On the other hand, event logs often record rich information about process execution, and by applying the existing process mining techniques (van der Aalst, 2016) relevant process knowledge can be extracted from event logs, which provide contextual or domain knowledge useful for understanding model explanations. A novel contribution of our approach is the proposal of model inspection that leverages both the explanations generated by technical interpretable machine learning mechanisms and the contextual or domain knowledge extracted from event logs. Findings drawn from this work are expected to serve as a key input to developing model reliability metrics and evaluation in the context of business process predictions.

Next, Section 2 provides background information with a review of related work. Section 3 proposes our approach to support inspection of business process prediction models. Section 4 presents model inspection and analysis in the context of two existing predictive process monitoring benchmarks and using three real-life event logs. Finally, Section 5 concludes the paper.

## 2 Background and Related Work

### 2.1 Business Process Event Log

Business process execution involves multiple perspectives and data generated along process execution are recorded in *event logs*. Fig. 1 depicts an example of a simplified hospital process and its multiple execution perspectives, and Fig. 2 shows a fragment of an event log that records information about the process execution. The example is inspired by the information captured by a real-life event log recorded by the information systems in a Dutch academic hospital<sup>1</sup>. The process consists of several medical activities: consultation, test, diagnosis, treatment and check-up. Each instance of process execution (*a.k.a.* case) captures a hospital episode of a specific patient and can be identified by the patient reference number recorded in the hospital's IT systems.

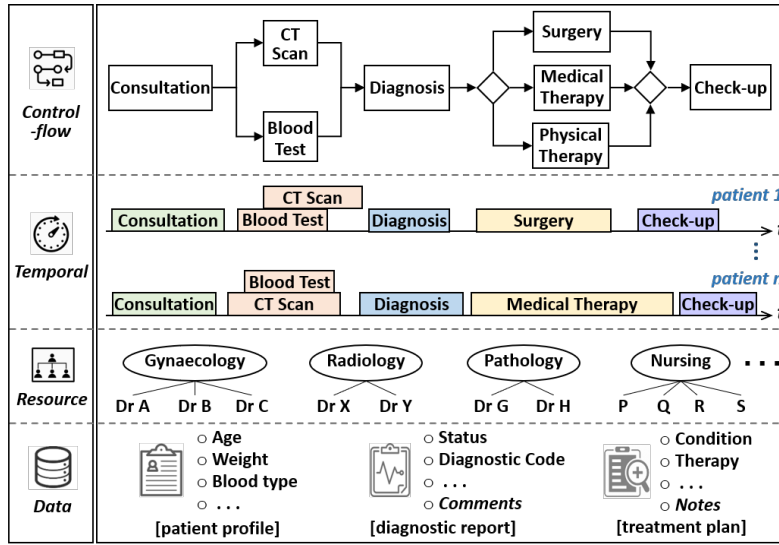
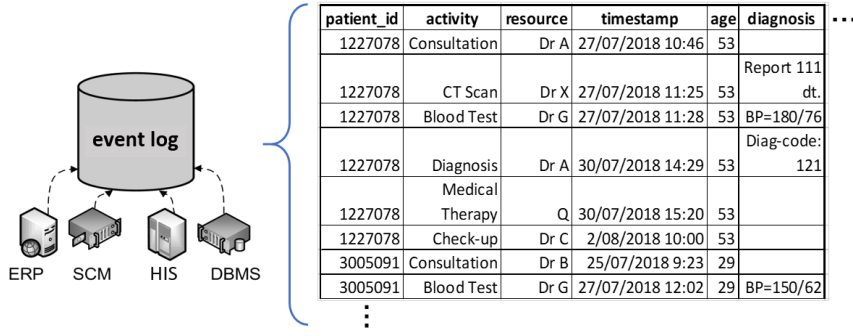


Fig. 1 A simplified hospital process and multiple perspectives of process execution

An event log comprises *events* and *event attributes* (see Definitions 1 and 2). Examples of events and event attributes are shown respectively as rows and columns in Fig. 2. An event log has a multi-dimensional data structure capturing multiple perspectives of process execution. It consists of: (1) *control-flow* data (describing the activities and their occurrences); (2) *temporal* data (representing an ordered sequence of timestamps relevant to activity occurrences); (3) *resource* data (which is tabular data recording the resources and their attributes and may be extracted from HR databases); and (4) *case* data (which is

<sup>1</sup> Refer to Business Process Intelligence Challenge (BPIC) 2011 at <https://www.win.tue.nl/bpi/doku.php?id=2011:challenge>.



**Fig. 2** A fragment of an event log that records execution of the process in Fig. 1

tabular data recording the process instances and their attributes, e.g., patient age group, blood type, disease associated with a hospital episode). In addition, an event log may also contain event attributes that record textual data (corresponding to such as notes or comments relevant to process execution).

**Definition 1 (Event log)** Let  $\mathcal{E}$  be the set of all event identifiers (or *events* in short),  $\mathcal{A}$  the set of all attribute names,  $\mathcal{V}$  the set of all attribute values. An *event log* can be defined as  $L = (E, A, \gamma)$ , where  $E \subseteq \mathcal{E}$  is the set of events,  $A \subseteq \mathcal{A}$  is the set of event attributes, and  $\gamma \in E \rightarrow (A \rightarrow \mathcal{V})$  specifies event attribute value bindings. Each event  $e \in E$  has event attributes  $\text{dom}(\gamma(e))$ . For each  $a \in \text{dom}(\gamma(e))$ ,  $\gamma_a(e) = \gamma(e)(a)$  is the value of attribute  $a$  for event  $e$ .

**Definition 2 (Event attribute)** Let  $\text{Case} \subseteq \mathcal{V}$  be the set of case identifiers,  $\text{Act} \subseteq \mathcal{V}$  the set of activity names,  $\text{Time} \subseteq \mathcal{V}$  the set of timestamps,  $\text{Res} \subseteq \mathcal{V}$  the set of resource identifiers, and  $\text{Data} \subseteq \mathcal{V}$  the set of case data values. An event log  $L = (E, A, \gamma)$  has three *mandatory* event attributes  $\{\text{case}, \text{act}, \text{time}\} \subseteq \text{dom}(\gamma(e))$  and an arbitrary finite number of additional event attributes  $\{\text{res}, \text{data}_1, \dots, \text{data}_n\} \subseteq A$  (where  $n \in \mathbb{N}$ ). For each  $e \in E$ :

- $\gamma_{\text{case}}(e) \in \text{Case}$  is the case to which event  $e$  belongs,
- $\gamma_{\text{act}}(e) \in \text{Act}$  is the activity that  $e$  refers to,
- $\gamma_{\text{time}}(e) \in \text{Time}$  is the time at which  $e$  occurred,
- $\gamma_{\text{res}}(e) \in \text{Res}$  is the resource that executed  $e$  (if  $\text{res} \in \text{dom}(\gamma(e))$ ), and
- $\gamma_{\text{data}_i}(e) \in \text{Data}$  (where  $i \leq 1 \leq n$ ) is some case data associated with  $e$  (if  $\text{data}_i \in \text{dom}(\gamma(e))$ ).

## 2.2 Business Process Prediction

Business process event logs comprising historical process execution are widely used for predicting how a business case, such as a loan application, will unfold until its completion at run time. Such predictions intend to provide better service to customers by mitigating delays or unexpected outcomes. There has

been a growing interest in building complex machine learning models for predicting business process behaviour, such as the next event in a case, the time for completion of an event, and the remaining execution trace of a case. Early work focuses on extracting features from business process event logs (Leontjeva et al., 2015). As most machine learning algorithms need to be configured with certain hyperparameters for better performance based on the characteristics of the event log, a framework for tuning the hyperparameter using genetic algorithms has been proposed (Francescomarino et al., 2018). A systematic review and taxonomy of methods predicting the business process outcome has been presented with a comparative experimental evaluation of eleven representative methods on nine real-world event logs (Teinemaa et al., 2019). Another study presents a systematic review and taxonomy for predicting the remaining time of a business process instance and provides an experimental evaluation of representative methods on sixteen real-world datasets (Verenich et al., 2019).

In this work, we focus on the above two benchmark studies by Teinemaa et al. (2019) and Verenich et al. (2019) that evaluate various techniques used in the context of predictive process monitoring. Existing methods on predicting future behaviour of business process mainly apply supervised machine learning algorithms. A supervised learning algorithm uses a corpus of input, output pairs (or training data) to learn a hypothesis that can predict the output for a new or unseen input (test data). The input and output are derived from the event logs. The output in the two benchmarks are: an outcome of a case defined by using a labelling function, and the remaining time for a case to complete. In both benchmark studies, evaluation of different business process prediction techniques focuses on model accuracy. For outcome-oriented prediction, this is measured by the *area under the ROC curve* (AUC) metric, and for remaining time prediction, this is measured by the *mean absolute error* (MAE) metric. Higher the AUC, better the model is at predicting the outcomes. MAE for remaining time prediction is the absolute difference between the actual remaining time and predicted remaining time and a lower MAE indicates better performance of the prediction model.

### 2.3 Black Box Explanation

In recent years, the topic of *explainable* and *interpretable machine learning* (Guidotti et al., 2018) has gained more and more attention. Use of machine learning in critical decision-making processes has urged the need to provide the reasoning that leads to predictions made by these algorithms. Model interpretability can be addressed by having *interpretable models* or providing *post hoc interpretations*. An *interpretable model* (Lipton, 2018), is able to provide transparency at the level of entire model, the level of individual components, and the level of the learning algorithm. For example, both linear regression models and decision tree models are interpretable models, while neural network models are complex and hard to follow and hence have low transparency.

Another distinct approach to address model interpretability is via *post hoc interpretation*. Here, explanations and visualisations are extracted from a learned model, that is, *after* the model has been trained, and hence are *model agnostic*. Existing techniques can be divided into two categories: partial dependence models and surrogate models. More specially, *surrogate models* use the input data and a black box model (i.e., a trained machine learning model) and emulate the black box model. In other words, they are approximation models that use interpretable models to approximate the predictions of a black box model, enabling a decision-maker to draw conclusions and interpretations about the black box (Molnar, 2018). Interpretable machine learning algorithms, such as linear regression and decision trees are used to learn a function using the predictions of the black box model. This means that this regression or decision tree will learn both well classified examples and misclassified ones. Measures such as mean squared error are used to assess how close the predictions of the surrogate model approximate the black box. As a result, the explanations derived from the surrogate model reflect a local and linear representation of the black box model.

Explanations from a black-box model permits us to confirm multiple prerequisites of a machine learning model (Doshi-Velez and Kim, 2017), which are *fairness, reliability and robustness, causality, privacy and trust*. In this work, we aim to present an approach that allows us to use model explanations to investigate certain reasoning applied by machine learned predictions and detect potential issues thus enhancing *trust* in business process prediction models.

### 3 Approach

In this section, we propose an approach to support XAI-enabled inspection of business process prediction models. Fig. 3 depicts an overview of the approach.

A standard workflow for building business process prediction models based on machine learning (Verenich et al., 2019) is illustrated at the top of Fig. 3. At first, *prefixes* are extracted for each trace (denoted as  $\sigma$ ).  $\sigma$  is an ordered sequence of  $n$  events  $\langle e_1, \dots, e_n \rangle$  and corresponds to a set of  $n$  prefixes  $\{\langle e_1, \dots, e_i \rangle \mid 1 \leq i \leq n\}$ . We use  $prefix(\sigma, i)$  to denote the sequence of the first  $i$  events extracted for  $\sigma$  and thus  $i$  specifies the prefix length. Next, the prefixes extracted for all instances of process execution are grouped into *buckets* based on their similarities (such as prefix length or state of process execution) and the prefixes in each bucket are then encoded into feature vectors. Different techniques can be applied for bucketing and encoding prefixes and will be discussed in Section 4.1. The buckets of feature vectors are ready as input for a machine learning (ML) algorithm. Since the future state for each trace is known from event log data, pairs of the encoded prefix and the future state are used to train a ML-based prediction model for each bucket. The resulting prediction model is assessed for its performance using conventional metrics in machine learning (such as accuracy, precision, recall, F1 score).

Since business process prediction models are mostly built upon advanced machine learning algorithms to achieve better model performance, they become black-box models. A novel contribution of our approach is the proposal of process prediction model inspection as depicted in Fig. 3. A key step is to generate explanations about the process predictions made by black-box models using interpretable ML techniques. For example, it is important to understand the reasons for machine learned process predictions (Guidotti et al., 2018). Often the output produced by a technical interpretation mechanism requires validation with a human expert or via applying relevant contextual or domain knowledge before it can be used to derive meaningful insights. In the field of business process studies, event logs record process execution history and by using process mining techniques (van der Aalst, 2016) various process knowledge can be extracted from event log data, which can serve as contextual or domain knowledge. A typical example is discovery of the (logged) execution paths of process activities (i.e., control-flow of process execution) in the form of a process graph. Hence, we propose the inclusion of another important step in our approach, which focuses on extracting process knowledge relevant to explanations of process prediction. Both the technical interpretation of black-box predictions and relevant process knowledge extracted from event log data are used for inspection of process prediction models. The expected findings from such model inspection include detection of potential issues associated with a process prediction model, diagnosis of the issues, explanations about the issues and/or reasons that may have caused the issues, etc.

We consider that detection of model-associated issues will serve as a key input to developing model reliability metrics and evaluation, and diagnosis and explanations of the issues will help with model tuning for improving the reliability and robustness of process prediction models in our future work.

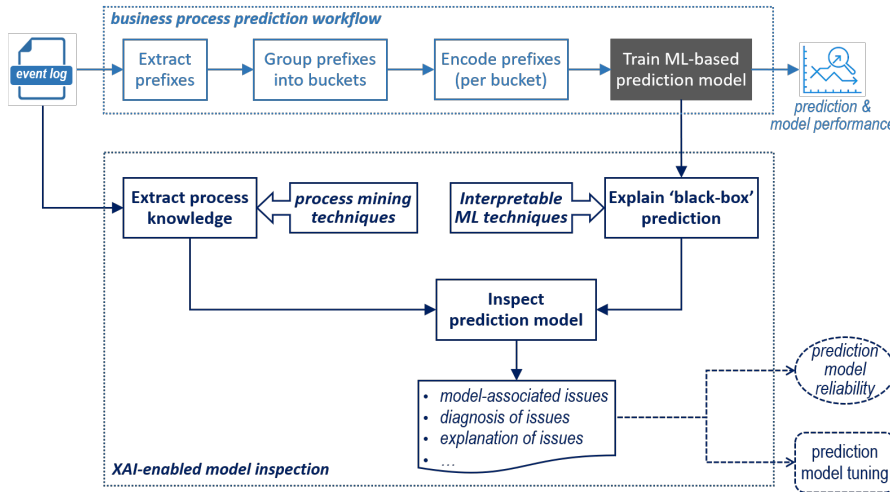


Fig. 3 Approach for XAI-enabled inspection of business process prediction models

## 4 Model Inspection and Analysis

Following the approach proposed in the previous section, we carry out inspection of business process prediction models used for predicting process outcome and remaining time, and analyse the inspection results to draw findings about the prediction models. The source code of model inspection and analysis is available <https://git.io/Je186> (regarding process outcome prediction) and <https://git.io/Je1XZ> (regarding process remaining time prediction).

### 4.1 Design and Configuration

In the context of two existing predictive process monitoring benchmarks (see Section 2.2), combinations of various bucketing, encoding, and supervised learning algorithms have been evaluated for predicting process outcome (Teinmaa et al., 2019) and remaining time (Verenich et al., 2019), respectively. In this study, we decide to choose the following techniques, given the fact that the methods built upon a combination of these techniques have better performance (i.e., high AUC values or low MAE values) as compared to others according to the benchmark evaluation.

*Bucketing techniques:* i) **single bucket**, where all prefixes of traces are considered in a single bucket, and a single prediction model is trained; and ii) **prefix length bucket**, where each bucket contains the prefixes of a specific length, and one prediction model is trained for each possible prefix length.

*Encoding methods:* i) **static encoding**, where the case attributes of each event that remain the same through out the prefixes of a trace is added as a feature *as-is*. The categorical attributes are one-hot encoded, where each categorical attribute value is represented as a feature that takes a binary value of 0 or 1; and ii) **aggregation encoding**, where the prefixes in each bucket are transformed by considering only the frequencies of event attributes (such as activity, resource) and computing four features to capture the numeric event attributes (maximum, mean, sum and standard deviation), and note that this way the order of the events in a prefix is ignored.

*Machine learning algorithms:* We choose *gradient boosted trees* (Friedman, 2001) (specifically **XGBoost**), which is used in both benchmarks and outperformed the other machine learning techniques (e.g., random forest, support vector machines, logistic regression). Note that the benchmark study by Verenich et al. (2019) considers *Long Short Term Memory* (LSTM) as a deep learning algorithm used for remaining time prediction. Given that deep learning-based models require different feature encoding from traditional machine learning algorithms, LSTM is yet not considered in this study.



*Interpretable ML techniques:* To make sure that the output and performance of a business process prediction model under inspection remain intact, we apply *post hoc interpretation* to derive explanations for a process prediction model built upon the above design and configuration.

When generating explanations of process prediction models, we propose that **global explanations** can be derived as the first step to gain an overall understanding of a model. These explanations may already help reveal certain issues associated the model. In this study, we conduct *permutation feature importance measurement* supported by gradient boosted trees to obtain relevant global explanations about a process prediction model. More specifically, the feature importance value generated by XGBoost is used to explain the impact of different features to the overall predictions made by a prediction model.

Next, we are interested in deriving **local explanations** about specific process predictions. In this study, we choose a representative local surrogate technique, known as *Local Interpretable Model-Agnostic Explanations* (**LIME**) (Ribeiro et al., 2016), which can explain the predictions of any classification or regression algorithm by approximating it locally with a linear interpretable model. We use LIME to generate local explanations useful in interpreting the prediction for a particular trace of process execution. The explanations by LIME are represented in terms of feature attribution, i.e., the features that have impacted a specific model learned prediction and the value of feature attribution measuring such impact.

*Process mining platform:* We leverage existing process mining techniques for discovery of process knowledge. The discovered control-flow of process execution is often specified in process graphs. Statistical/descriptive analysis can be applied in the context of process execution. A variety of process mining techniques are supported by existing process mining software platforms. We choose DISCO<sup>2</sup> as the process mining platform in our experiments.

## 4.2 Datasets and Notations

We present model inspection and analysis on three real-life event logs that are representative of our experiments and produce interesting results and findings. The event logs, known as BPIC 2011<sup>3</sup>, BPIC 2012<sup>4</sup> and BPIC 2015<sup>5</sup>, are from the Business Process Intelligence Challenge initiative and publicly available at 4TU.ResearchData. They were used for evaluation of business process predictions in the two benchmark studies (Teinemaa et al., 2019; Verenich et al., 2019). Below are brief descriptions of these three event logs as well as certain

<sup>2</sup> A commercial tool supporting process mining for professionals and providing academic licenses free of charge (<https://fluxicon.com/disco/>)

<sup>3</sup> [https://data.4tu.nl/articles/dataset/Real-life\\_event\\_logs\\_-\\_Hospital\\_log/12716513/1](https://data.4tu.nl/articles/dataset/Real-life_event_logs_-_Hospital_log/12716513/1)

<sup>4</sup> [https://data.4tu.nl/articles/dataset/BPI\\_Challenge\\_2012/12689204](https://data.4tu.nl/articles/dataset/BPI_Challenge_2012/12689204)

<sup>5</sup> [https://data.4tu.nl/collections/BPI\\_Challenge\\_2015/5065424](https://data.4tu.nl/collections/BPI_Challenge_2015/5065424)

notations to be used in the experiments. A summary of statistics of the three event logs used for model inspection and analysis is provided in Table 1.

*BPIC 2011*: The event log contains cases of patient visits to the Gynaecology department of a Dutch hospital (referred to as *bpic2011*). Process remaining time prediction using the log is considered for model inspection and analysis.

*BPIC 2012*: The event log records the execution of a loan application process at a Dutch financial institution. For process outcome prediction, each trace in the log is labelled as “*accepted*”, “*declined*”, or “*cancelled*” (based on whether the trace contains the occurrence of activity `O_ACCEPTED`, `O_DECLINED`, or `O_CANCELLED`). One of the three logs – *bpic2012\_1* which concerns loan acceptance – is considered.

*BPIC 2015*: There are five event logs recording the execution of a permit application process at five Dutch municipalities, respectively. For process outcome prediction, the rule stating “every occurrence of activity `01_HOOFD_020` is eventually followed by activity `08_AWB45_020_1`” is used to label the outcome of a trace as *positive* (Teinemaa et al., 2019). The log for one of the municipalities – *bpic2015\_5* – is considered.

**Table 1** Statistics of the three event logs used for model inspection and analysis

event log	# cases	# events	# activities	# static attributes	# dynamic attributes	min length	max length	avg length
bpic2011	1,140	149,730	251	6	14	1	1,814	25.0
bpic2012_1	4,685	186,693	36	1	10	15	175	35.0
bpic2015_5	1,051	54,562	217	18	12	5	134	50.0

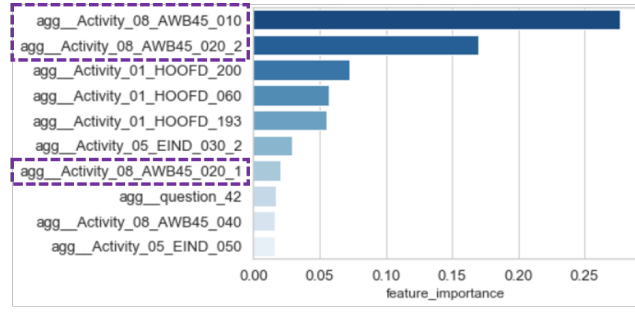
*Notations*: The feature representations in both local and global explanations used in model inspection in the next section are to be read as follows:

- `agg__[Activity] | [Resource]` represents the frequency of an activity executed by a resource) in a trace via aggregation encoding (**agg**).
- `static__[attribute]` represents the case attribute for a trace (of which the value does not change during the execution of the trace).

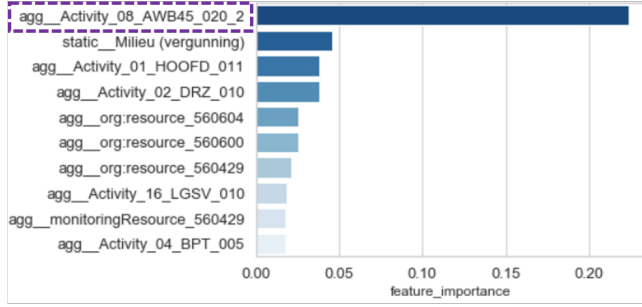
## 4.3 Analysis and Discussion

### 4.3.1 Detecting Data Leakage

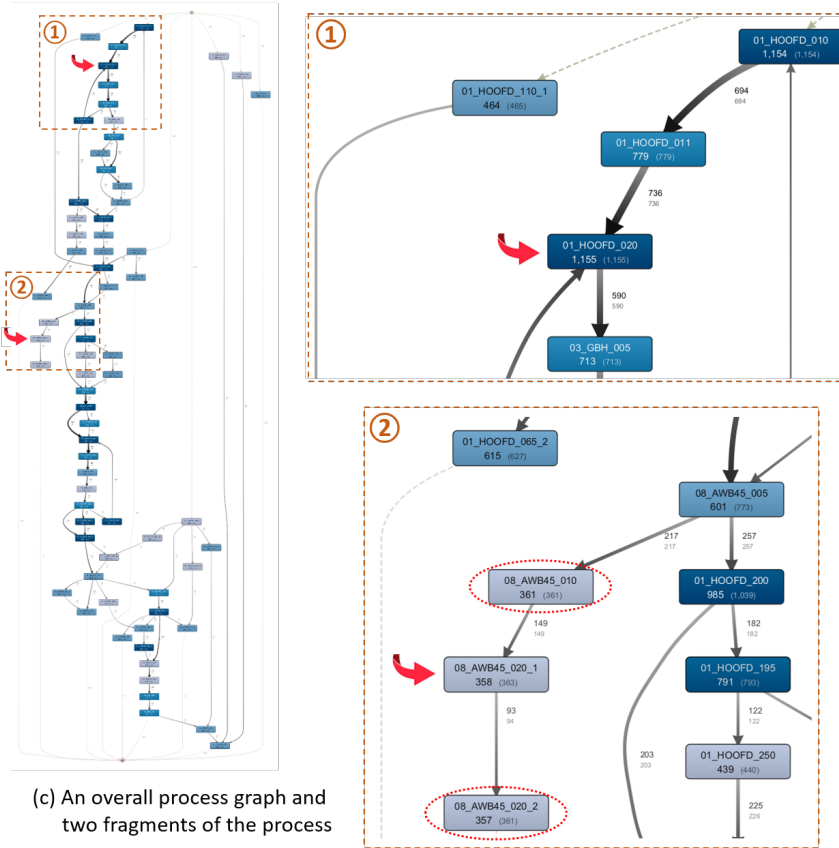
We inspect the business process prediction models trained for process outcome prediction using event log *bpic2015\_5*. The following two combinations of bucketing and encoding methods are applied respectively with XGBoost: i) single bucket and aggregation encoding (*single\_agg*) and ii) prefix length bucket and aggregation encoding (*prefix\_agg*).



(a) Global explanation (single\_agg)



(b) Global explanation (prefix\_agg; bucket of prefix length 10)



**Fig. 4** Inspecting process outcome prediction models for *bpic2015-5* event log with XG-Boost: (a) global explanation of the model using *single\_agg* method; (b) global explanation of the model trained for bucket of prefix length 10 using *prefix\_agg* method; and (c) an overall process graph and two fragments of the process captured by the event log.

Fig. 4(a) shows the global explanation for the model using *single\_agg* method. The occurrences of three activities 08\_AWB45\_010, 08\_AWB45\_020\_2 and 08\_AWB45\_020\_1 are identified among the important features for outcome prediction. However, the occurrence of 08\_AWB45\_020\_1 is part of the outcome to be predicted (as described in Section 4.2). Fig. 4(b) shows the global explanation for the model trained for the bucket of prefix length 10 using *prefix\_agg* method with XGBoost. The activity 08\_AWB45\_020\_2 is identified as the most important feature used for prediction. Fig. 4(c) depicts an overall process graph of the process execution captured by event log *bpic2015\_5* and zoom-in views of two fragments of the process: one contains activity 01\_H00FD\_020<sup>6</sup>, the other involves activities 08\_AWB45\_010, 08\_AWB45\_020\_1 and 08\_AWB45\_020\_2.

For both prediction models, global explanations together with the knowledge of the business process (discovered from event log data via process mining techniques) can be used to identify potential issues with the prediction model under inspection. As shown in Fig. 4(c), each activity in the process graph is also annotated with case frequency – the digit displayed under the activity name – which specifies the number of cases involving the occurrence of the activity. A statistical analysis over the process graph indicates that: i) activity 08\_AWB45\_020\_2 is executed after 08\_AWB45\_020\_1 in 68% of the cases, and ii) activity 08\_AWB45\_010 occurs prior to 08\_AWB45\_020\_1 in 50% of the cases.

All these observations reveal that the prediction model of *single\_agg* with XGBoost exhibits a problem of **data leakage** (Kaufman et al., 2011), “where information about the label of prediction that should not legitimately be available is present in the input”. As such, the features that occur along with or after the activity used as the label have likely influenced the model predictions.

#### 4.3.2 Investigating Feature Relevance

The purpose of this analysis is to reason the relevance of the features identified important for process prediction. Feature importance in global explanations and feature impact to predictions of traces in local explanations are valuable inputs to investigating feature relevance. At the same time, relevant knowledge about process execution is also needed, and often domain knowledge may also be required in order to understand how relevant a feature is to a given context (such as medical scenarios). Below, we discuss two examples of model inspection for analysing feature relevance: i) process outcome prediction using event log *bpic2012\_1* and ii) process remaining time prediction using log *bpic2011*.

*Analysis I:* We inspect the prediction model trained for process outcome prediction of event log *bpic2012\_1* using a *single\_agg* method and a single classifier XGBoost (trained for a single bucket containing all traces of all prefix lengths).

Starting with global explanations, Fig. 5(a) shows the importance of the features used by the model, in which the occurrences of activities O\_SENT\_BACK,

<sup>6</sup> As described in Section 4.2, the occurrence of activity 01\_H00FD\_020 eventually to be followed by the occurrence of activity 08\_AWB45\_020\_1 is part of the rule for labelling process outcome for event log *bpic2015\_5*.

W\_Validate Request, A\_DECLINED, O\_DECLINED, A\_CANCELLED and A\_APPROVED are the top six important features<sup>7</sup>. We then generate local explanations for two randomly chosen traces for which the model correctly predicted the positive outcome (i.e., loan accepted). Fig. 5(b) and (c) illustrate local explanations for traces with prefix lengths of 5 and 25, respectively. In both, the first three important features are given by the *absence of occurrences of specific activities*, and these activities are among the top six important features in the global explanation (Fig. 5(a)). Also, taking as an example the most important feature in both local explanations, the absence of the occurrence of activity A\_CANCELLED has a significant impact on the prediction of loan being accepted, and this seems reasonable for such outcome prediction.

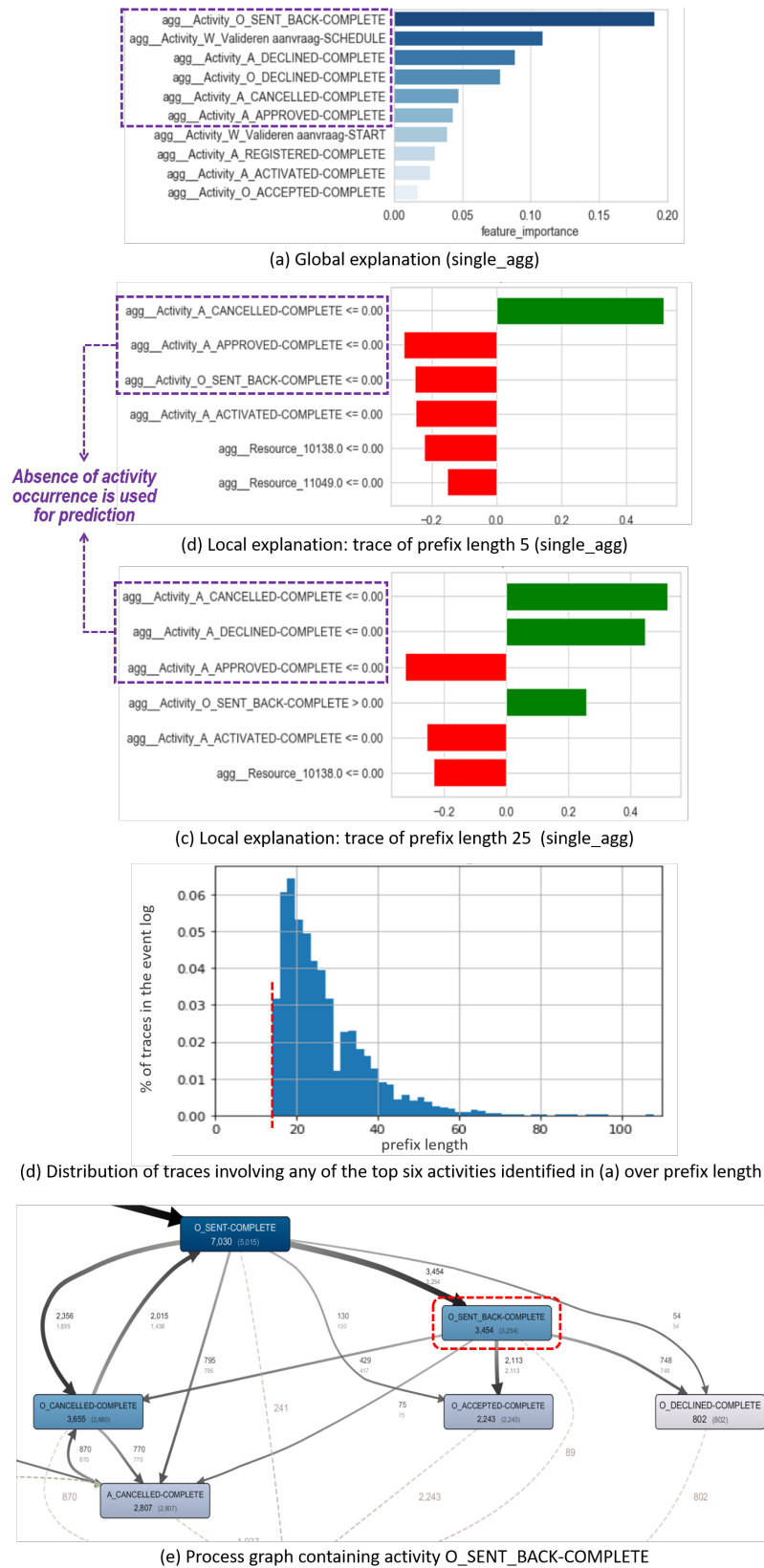
We are interested in obtaining relevant knowledge about process execution captured by *bpic2012\_1* event log for further investigation. A statistical analysis over the discovered process from the log reveals that the top six important activities identified in the global explanation (Fig. 5(a)) occur in the traces with a *minimum* prefix length of 14 during process execution (refer to Fig. 5(d)). This means that, for any trace with a prefix length less than 14, none of these six activities (including e.g., A\_CANCELLED) can occur, and hence the absence of any of these activities is *irrelevant* for making a prediction for such a trace. Given this knowledge, using the absence of any of these activities as an important feature to make an outcome prediction for the trace with a prefix length of 5 (Fig. 5(b)) is *not reliable*.

In contrast, we observe that for a trace with a (longer) prefix length of 25 (Fig. 5(c)), the absences of activities A\_CANCELLED, A\_DECLINED and A\_APPROVED can be relevant features for prediction, which is also consistent with feature importance provided in the global explanation. Furthermore, the model also uses the occurrence of activity O\_SENT\_BACK as an important feature for predicting a positive outcome, which is also relevant feature as the process discovered from the event log indicates that a loan will likely be accepted after the occurrence of O\_SENT\_BACK (see Fig. 5(e)).

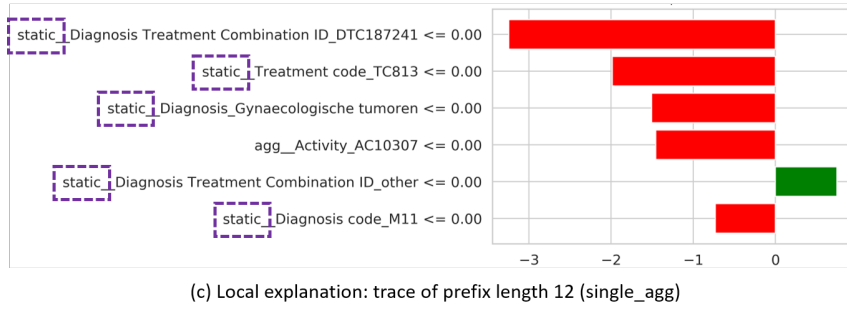
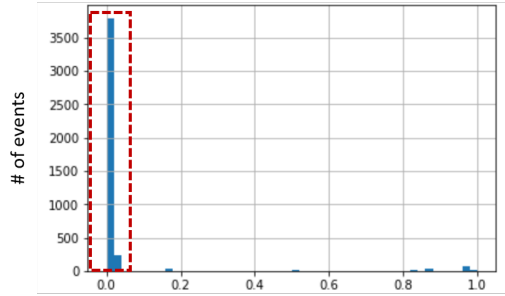
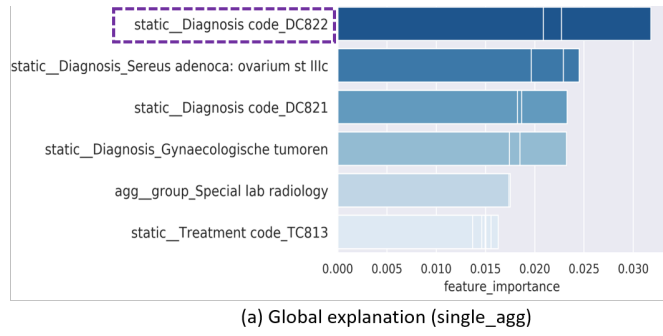
*Analysis II:* We inspect the prediction model trained for process remaining prediction of *bpic2011* event log using a *single\_agg* method with XGBoost.

As an interesting observation, the global explanation shown in Fig. 6(a) indicates that the most significant feature for remaining time prediction is `Diagnosis_code = DC822`. A statistical analysis over the cases of process execution related to this particular diagnosis code reveals that 82% of the events associated with this diagnosis code had the feature `elapsed time = 0` (refer to Fig. 6(b)), which means that the corresponding activity starts and ends immediately. To this end, it is necessary to gain relevant domain knowledge, in addition to process knowledge, to derive further insights about the relationship between this particular diagnosis code, the observation of 0-valued elapsed time, and their relation with the remaining time till a running case of the process is completed.

<sup>7</sup> For brevity, we omit the activity label suffix ‘-COMPLETED’ in the discussion.



**Fig. 5** Inspecting process outcome prediction model for *bpic2012\_1* event log using *single\_agg* method with XGBoost: (a) global explanation; (b) and (c) local explanations for traces of prefix lengths 5 & 25; (d) distribution of certain trace occurrences over prefix length used for prediction; and (e) fragment of discovered process containing activity O\_SENT\_BACK.



**Fig. 6** Inspecting process remaining time prediction for *bpic2011* event log using *single\_agg* with XGBoost: (a) global explanation; (b) normalised elapsed time for diagnosis code 'DC822'; and (c) local explanation for trace of prefix length 12.

Next, from the local explanation of a randomly chosen trace with a prefix length of 12 shown in Fig. 6(c), it can be observed that the prediction model uses features like **Diagnosis Treatment Combination ID**, **Treatment code** and **Diagnosis code** in order to determine the remaining time of the case. Such explanation reveals that the model is relying mostly on **static** features, which are the features that do not change throughout the lifetime of a case. The usage of static features suggests that i) the process execution does not rely on the dynamic aspect of process execution (such as the sequence of activities that have occurred), and ii) the model uses attributes that do not change

during the case execution when making a prediction. Hence, the relevance of these static features for remaining time prediction, which is a typical regression problem, is questionable.

## 5 Conclusion

We have presented an approach to support inspection of business process prediction models by leveraging both the explanations generated by interpretable machine learning mechanisms and the process knowledge extracted from event logs via process mining techniques. The approach has been used to guide the design of several experiments of model inspection in the context of two existing predictive process monitoring benchmarks and three real-life event logs. The results of model inspection have been analysed leading to interesting findings on data leakage and use of irrelevant features in existing models for business process predictions.

From this work, we have also learned that: i) to gain a deeper understanding of model explanations, the domain knowledge of a particular field is also necessary in addition to relevant process knowledge that can be extracted from event logs; ii) relying on conventional model performance measures (e.g., accuracy, precision) is not adequate, and model reliability should be included as another important aspect for evaluation; and iii) results from model inspection can help with model tuning to improve the reliability and robustness of process prediction models. These will inform directions of our future work.

**Acknowledgement:** We thank the authors of the two predictive process monitoring benchmarks (Teinemaa et al., 2019; Verenich et al., 2019) for the high quality code they released, which allowed us to conduct experiments of model inspection on selected business process prediction models.

## References

- Doshi-Velez F, Kim B (2017) Towards a rigorous science of interpretable machine learning, arXiv: 1702.08608v2
- Francescomarino CD, Dumas M, Federici M, Ghidini C, Maggi FM, Rizzi W, Simonetto L (2018) Genetic algorithms for hyperparameter optimization in predictive business process monitoring. *Inf Syst* 74(Part):67–83
- Friedman JH (2001) Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5):1189–1232, URL <http://www.jstor.org/stable/2699986>
- Guidotti R, et al. (2018) A survey of methods for explaining black box models. *ACM Comput Surv* 51(5):93:1–93:42
- Kaufman S, Rosset S, Perlich C (2011) Leakage in data mining: Formulation, detection, and avoidance. In: *Procs. of the 17th ACM SIGKDD*, pp 556–563



- Lakkaraju H, et al. (2019) Faithful and customizable explanations of black box models. In: Proceedings of the 2019 AAAI Conference on AIES 2019, pp 131–138
- Leontjeva A, Conforti R, Francescomarino CD, Dumas M, Maggi FM (2015) Complex symbolic sequence encodings for predictive monitoring of business processes. In: Business Process Management - 13th International Conference, BPM, Proceedings, Springer, Lecture Notes in Computer Science, vol 9253, pp 297–313
- Lipton ZC (2018) The mythos of model interpretability. CACM 61(10):36–43, DOI 10.1145/3233231
- Molnar C (2018) Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Leanpub
- Ribeiro MT, Singh S, Guestrin C (2016) “Why should I trust you?”: Explaining the predictions of any classifier. In: Procs. of the 22nd ACM SIGKDD, pp 1135–1144
- Rudin C (2019) Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1(5):206–215
- Teinemaa I, Dumas M, Rosa ML, Maggi FM (2019) Outcome-oriented predictive process monitoring: Review and benchmark. TKDD 13(2):17:1–17:57
- van der Aalst WMP (2016) Process Mining: Data Science in Action (2nd Edition). Springer, Berlin, Heidelberg
- Verenich I, Dumas M, Rosa ML, Maggi FM, Teinemaa I (2019) Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM TIST 10(4):34:1–34:34