Interpreting Predictive Process Monitoring Benchmarks
An Exploratory Study

Renuka Sindhgatta, Chun Ouyang, Catarina Moreira, and Yi Liao
Queensland University of Technology, Brisbane, Australia
{renuka.sr,c.ouyang,catarina.pintomoreira}@qut.edu.au

Abstract. Predictive process analytics has recently gained significant attention, and yet its successful adoption in organisations relies on how well users can trust the predictions of the underlying machine learning algorithms that are often applied and recognised as a ‘black-box’. Without understanding the rationale of the black-box machinery, there will be a lack of trust in the predictions, a reluctance to use the predictions, and in the worse case, consequences of an incorrect decision based on the prediction. In this paper, we emphasise the importance of interpreting the predictive models in addition to the evaluation using conventional metrics, such as accuracy, in the context of predictive process monitoring. We review existing studies on business process monitoring benchmarks for predicting process outcomes and remaining time. We derive explanations that present the behaviour of the entire predictive model as well as explanations describing a particular prediction. These explanations are used to reveal data leakages, assess the interpretability of features used by the model, and the degree of the use of process knowledge in the existing benchmark models. Findings from this exploratory study motivate the need to incorporate interpretability in predictive process analytics.

Keywords: predictive process monitoring · interpretability · prediction explanation · machine learning

1 Introduction

Predictive process analytics is a relatively new discipline that aims at predicting future observations of a business process by learning from event log data capturing the process execution history. A vast majority of work over the past decade has used supervised machine learning algorithms to predict outcomes of a business process execution (or a case) [1], the next activity in a case [2], or the remaining time for a case to complete [3]. Evaluation of a predictive method has so far been assessed in terms of the quality of the learning model and thus evaluated using conventional metrics in machine learning (such as accuracy, precision, recall, F1-score).

As an important branch of state-of-the-art data analytics, predictive process analytics is also faced with a challenge concerning the lack of explanation to the reasoning and outcome of its predictive methods. While more and more complex
machine learning techniques (and more recently, deep learning techniques) are used to build advanced predictive capabilities in process analytics, they are often applied and recognised as a ‘black-box’. The ‘black-box’ use of machine learning models makes it difficult for business and/or data analysts to understand the logic or reasoning and obtain insights into why certain decision or prediction was drawn. Without understanding the rationale of the ‘black-box’ machinery, there will be a lack of trust in the predictions, a reluctance to use the predictions, and in the worse case, consequences of an incorrect decision based on the prediction.

The recent body of literature in machine learning has emphasised the need to understand and trust the predictions (e.g., [4,5]). It has led to an increasing interest in the research community on interpretable machine learning [6]. “To interpret means to give or provide the meaning or to explain and present in understandable terms” [6]. Having an interpretable or explainable model is a necessary step towards obtaining the right level of understanding of the predictions.

In this paper, we aim to emphasise the importance of interpreting the predictive models in the context of business process monitoring. The availability of two exhaustive benchmarks enables this study. Teinemaa et al. [1], and Verenich et al. [3], present the two business process monitoring benchmarks established to assess state-of-the-art predictive methods for predicting process outcomes [1] or remaining time [3]. We apply existing methods to interpret the predictions of these benchmarks and derive explanations that present the behaviour of the entire predictive model (global explanation) as well as explanations describing a particular prediction (local explanation). These explanations reveal data leakage issues and help with assessing the interpretability of the features used by the benchmark models, and the degree of the use of process knowledge in the models. Findings drawn from this work are expected to motivate the need to incorporate interpretability in predictive process analytics. To the best of our knowledge, the closest study to this work is an illustration of the potential of explainable models for a manufacturing business process [7].

The rest of the paper is structured as follows. Sect. 2 provides necessary background information on predictive process monitoring benchmarks and interpretability of predictive models. Sect. 3 presents a detailed review of two existing predictive process monitoring benchmarks by applying interpretable machine learning techniques. Sect. 4 discusses and summarises our findings as some lessons and challenges. Finally, Sect. 5 concludes the paper with an outlook of future work.

2 Background

2.1 Predictive Process Monitoring Benchmarks

In this work, two studies on predictive process monitoring [1,3] are used. These predictive process monitoring benchmarks mainly apply supervised machine learning algorithms. A supervised machine 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: 1) an outcome of a case defined by using a labelling function, and 2) the remaining time for a case to complete. The availability of the processed event logs and source code for both these benchmarks enables reproducibility of models and further study.

**Predictive Process Monitoring Workflow.** The predictive process monitoring workflow used for predicting the future state of a case, such as the outcome label or remaining time for case completion, consists of two phases [1,3]. The first is an **offline** training phase where a predictive model (which is often a machine learning model) is trained using event log data of historical process executions; and the second is an **online** stage where the future state of a running case is predicted using the trained model from the offline phase (see Fig. 1).

![Predictive process monitoring workflow](image)

**Fig. 1.** Predictive process monitoring workflow described in [1,3]

A *trace* $\sigma$ is a sequence of *events* of the same case. During the offline phase, *prefixes* are generated for each trace. A prefix function $\text{prefix}(\sigma, l)$ takes as input a trace $\sigma$ and a prefix of length $l$, and returns the first $l$ events of the trace. The prefixes are grouped into *buckets* based on their similarities (such as length, process states, or events) using a *bucketing mechanism*. The prefixes in each bucket are encoded as feature vectors following an *encoding mechanism*. The buckets of feature vectors are then used to train a predictive model underpinned by a (machine or deep) *learning algorithm*. More specifically, since the future state for each case is known from the training data, pairs of the encoded prefix and the future state are used to train a predictive model for each bucket. During the online phase, the future state for a running trace is predicted by identifying the appropriate bucket, using an appropriate encoding, and finally applying the above trained predictive model.

**Evaluation Measures.** Existing predictive process monitoring methods are built on different combinations of bucketing mechanisms, encoding mechanisms,
learning algorithms. The two predictive process monitoring benchmarks evaluate these methods using the following quality measures:

- **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 and distinguishing 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 predictive model.

- **Earliness**: in both predictions, this is defined as the smallest prefix length with the desired level of accuracy.

### 2.2 Interpretability of Predictive Models

In recent years, the topic of *interpretable* or *explainable* machine learning has gained attention. To avoid ambiguity, we apply the terms *interpretability* and *explainability* as discussed in the machine learning literature [6,8]. Model interpretability can be addressed by having *interpretable models* and/or providing *post hoc interpretations*. Fig. 2 illustrates an overview of existing interpretable machine learning methods and techniques discussed in [9].

An *interpretable model* [8], is able to provide transparency at the level of entire model (*simulatability*), the level of individual components (*decomposability*), and the level of the learning algorithm (*algorithmic transparency*). 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.*

Fig. 2. Overview of methods and techniques to address model interpretability [9]
Existing techniques can be divided into two categories: i) partial dependence models and ii) surrogate models.

Among partial dependence models, Partial Dependence Plots (PDP) show the marginal effect of at most two features on the predicted outcome of a machine learning model [10]. However, PDP assumes that features are not correlated between each other, which may not always be true. To address such limitations of PDP, two algorithms are proposed, namely Individual Conditional Expectation (ICE) [11] and Accumulated Local Effects (ALE) [12]. ICE is similar to PDP, but focuses on individual data instances, and ALE describes the influence of features on an average on the prediction. The Permutation feature importance measures the importance of a feature by calculating its impact on prediction error after permuting the feature. Shuffling the value of an important feature would increase the model error because of the reliance of the model on the feature for prediction while shuffling value of an unimportant feature would result in minimal impact on the model error [13].

Surrogate models use the input data and a black box model (i.e., a trained machine learning model) and emulate the black box model [14]. 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 [9]. 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. In more detail, the general algorithm for surrogate models is presented in Algorithm 1 [9].

**Algorithm 1** General algorithm for surrogate models [9].

**Require:** Dataset $\mathcal{X}$ used to train black box, Prediction model $\mathcal{M}$

**Ensure:** Interpretable Surrogate model $\mathcal{I}$

1: Get the predictions for the selected $\mathcal{X}$, using the black box model $\mathcal{M}$
2: Select an interpretable model: linear model, regression tree, ...
3: Train interpretable model on $\mathcal{X}$, obtaining model $\mathcal{I}$
4: Get predictions of interpretable model $\mathcal{I}$ for $\mathcal{X}$
5: Measure the approximation of interpretable model $\mathcal{I}$ with the black box model $\mathcal{M}$
6: return Interpretation of $\mathcal{I}$

### 3 Interpreting Predictive Monitoring Benchmarks

In this section, we aim to interpret and gain deeper insights into the two process monitoring benchmarks for predicting process outcome or remaining time. This
is addressed from the following three aspects: first, to derive explanations for a model with high performance and evaluate the reliability of a model prediction; second, to analyse the influence of bucketing and encoding methods on model predictions, thus evaluating the reliability of the features used by the model; and third, to understand the degree of process knowledge used by the machine learning models.

| Encoding  | Attributes | Feature Extraction |
|-----------|------------|--------------------|
| Static    | Case       | as-is              |
| Aggregation | Events (unordered) | min, max, mean, std. deviation |
|           | Aggregation | frequencies or # of occurrences |
|           | Static     | as-is for each index |
|           | Index      | one-hot for each index |

Table 1. Encoding Methods

3.1 Design and Configuration

While several combinations of bucketing, encoding, and classification algorithms have been evaluated in the benchmarks, in this work we choose the following techniques, because the methods built upon a combination of these techniques usually outperform others according to the benchmark evaluation [1,3].

- Bucketing: i) single bucket, where all prefixes of traces are considered in a single bucket, and a single classifier is trained; and ii) prefix length bucket, where each bucket contains partial traces of a specific length, and one classifier is trained for each possible prefix length.

- Encoding: i) aggregation encoding, where the trace in each bucket is transformed by considering (only) the frequencies of event attributes such as activity, resource and computing four features for the numeric event attributes (max, mean, sum and standard deviation), and note that this way the order of the events in a trace is ignored; and ii) index encoding, where each event attribute (e.g., activity, originator/resource) of an executed event can be represented as a feature, and this way the order of a trace in each bucket can be maintained by encoding each event in the trace at a given index. iii) static encoding, where the trace (or case) attributes that remain the same throughout the 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. The encoding methods are summarised in Table 1.

- Classifier: gradient boosted trees [15] (specifically XGBoost) is the machine learning algorithm used in both the benchmarks. The classifier outperformed the other classification algorithms (random forest, support vector machines, logistic regression) when predicting outcomes. This study is restricted to machine learning algorithms and hence Long Short Term Memory (LSTM) [3] has not been considered.
We further apply selected techniques of post hoc interpretability to derive explanations for the benchmarks. To explain the predictions of the overall model, we use permutation feature importance provided by gradient boosted trees. To explain prediction for a trace, a representative local surrogate method, LIME [16] is used and local explanations are derived.

3.2 Datasets

We present results of three event logs that are representative for our analysis. A brief description of the event logs are presented:

**BPIC 2011:** The event log contains cases from the Gynaecology department of a Dutch hospital. For interpreting outcome-oriented predictions, the labelling function that uses the occurrence of the activity “histological examination - big resectiep” is used (bpic2011). The preprocessing of the event log has the trace for each case cut exactly before this event occurs. For predicting the remaining time, the log is used as-is without any truncation (bpic2011).

**BPIC 2012:** The event log contains the traces of a loan application process at a Dutch financial institution. For the outcome prediction, each trace in the log is labelled as accepted, declined, or cancelled based on whether the trace contains the activity 0_ACCEPTED, 0_DECLINED, 0_CANCELLED. One of the three logs (loan acceptance) is considered for understanding model explanations (bpic2012). For the remaining time prediction, three logs are generated depicting loan application, loan offers and loan processing by human workers. Explanations are presented for the model trained on the loan offers (bpic2012).

**BPIC 2015:** The event log contains the traces of a permit application process at a Dutch municipality. For the outcome prediction, the rule stating every occurrence of activity 01_HOOFD_020 is eventually followed by activity 08_AWB45_020 is used to label the trace as positive. Explanations are derived for one of the municipalities (bpic2015).  

**Notations:** The feature representations in the graphical figures in the rest of the sections should be read as follows:

- Frequency of an event attribute (activity, resource) in the trace using aggregation encoding is \( \text{agg}_[\text{Activity}]|[\text{Resource}] \).
- The index \( \text{idx} \) at which an activity or a resource of a given \( \text{name} \) occurs using index encoding is: \( \text{index}_[\text{Activity}][\text{Resource}][\text{idx}][\text{name}] \).
- The case attribute for a trace is: \( \text{static}_[\text{attribute}] \).
- The descriptive statistics (maximum, mean, standard deviation) of numeric event attributes using aggregation encoding: \( \text{agg}\max/\text{mean}/\text{std}_[\text{Attribute}] \).

3.3 Interpreting Outcome-Oriented Prediction

This section presents a summary of our findings on the prediction of case outcomes. While we present a key insights, our detailed post hoc explanations as well as the source code is available at https://git.io/Je186.
Understanding models with the highest performance: The predictive model trained for (bpic2015_5) log, has the best performing AUC and F1-score when using a single bucket with aggregation encoding (AUC=0.85, F1-score=0.77). We use this model and generate global model explanations. The feature importance value of XGBoost indicates the usefulness of the feature in the model prediction. The higher an improvement in accuracy brought by a feature, the higher its relative importance.

Two of the most important features are 08_AWB45_010 and 08_AWB45_020_2. Another important feature is 08_AWB45_020_1 (Figure 3 (a)). There are two observations to be made here: 1) The outcome to be predicted is the occurrence of 08_AWB45_020_1 which is also a feature in the model, 2) The correlation (pearson correlation coefficient) between the occurrence of 08_AWB45_020_1 and the label 08_AWB45_020_1 is 0.933 (highly correlated values), and correlation of 08_AWB45_020_2 with 08_AWB45_020_1 is 0.733. Hence, the observations for the single bucket present interesting insights into the model predictions. Single bucket aggregation encoding model has a problem of data leakage [17], “where information about the label of prediction that should not legitimately be available is present in the input”. The outcome or features highly correlated to the outcome are used by the model to make predictions.

![Fig. 3. Feature importance of XGBoost model, Local explanations for prefix length 5,10, and 25 respectively.](image)

To understand the predictions for a single trace, we use LIME [16] and generate local explanations for a few randomly chosen traces that have a positive outcome label (outcome=1). Figure 3 further presents explanations for three traces with a positive label having the prefix length of 5, 10, and 25, respectively. The green bar in the figure shows the feature and the expression it satisfies that increases the prediction probability of the positive class label, while the red bars indicate the feature and the expression it satisfies that reduces prediction proba-
bility of the positive class label. The local explanations for prefix length 5 and 10 show why the model predicted the outcome incorrectly. From the explanations it can be seen that the model relies on the two activities. Their non-occurrence \((08\_AWB45\_010 <= 0)\) leads to an incorrect prediction by the model for the prefix length of 5 and 10. The occurrence of the activity for prefix length 25 results in a correct outcome prediction. The local explanations reveal that the model is influenced by the occurrence of two activities or features (highly) correlated to the output, once again confirming the data leakage.

**Understanding bucketing and encoding techniques:** We generate global and local explanations to gain a deeper understanding of the influence of the encoding and bucketing techniques on the model prediction using \(bpic2012\_1\). The following three combinations are presented: i) single bucket and aggregation encoding, ii) prefix bucket and aggregation encoding, and iii) prefix bucket and index encoding.

![AUC performance plot, Feature importance for single bucket, distribution of the minimum prefix length of the activity occurrences used by the predictive model, local explanations for a trace of prefix length 5](image)

**Model explanations for buckets.** The **single bucket** using aggregation encoding method has high AUC and earliness measure at all prefix lengths compared to prefix bucket (Figure 4 (a)). The single bucket contains a single classifier trained for all prefix lengths of all traces. Figure 4(b) shows the important features used by the model trained on the single bucket and aggregation encoding. The model uses the occurrence of activities \(O\_SENT\_BACK, W\_Validate\)
Request, ADECLINED, ODECLINED, ACANCELLED as important features. The event log reveals that the top six activities occur in the traces at a minimum prefix length of 14 (Figure 4 (c)). In the online phase for traces with lower prefix lengths, these features will not be available for a single bucket model to make the prediction. From the global explanations of the model, it can be therefore be inferred that for traces of lower prefix lengths, the features considered important by the model for prediction are not available and the single bucket model uses future events for prediction.

We further use LIME and generate local explanations for two randomly chosen traces that have a positive outcome label (outcome=1). Figure 4 (d) shows explanations for a trace with a prefix length of 5. At lower prefix length, while the model predicts accurately, the non-occurrence of the activity ACANCELLED influences the prediction, which is but obvious and not a valuable information. We, therefore, conclude that while single bucket has higher accuracy and earliness, it would not be suitable for traces with smaller prefix length. At the prefix length of 25, the model uses the occurrence of the activity OSENT_BACK as an important feature for the positive outcome (or loan acceptance). The process model reveals that this event is relevant as a loan may be accepted after its occurrence.

Fig. 5. Feature importance (top 10) for prefix bucket and aggregation encoding, length for a randomly chosen trace of prefix lengths 5, 10.

We further analyse prefix length bucket with aggregation encoding where a classifier is trained for each bucket (or prefix length). The global and local explanations for buckets containing prefixes of length 5, and 10 are shown in Figure 5. Buckets with lower prefix lengths do not use information of the control-flow of the trace executed thus far. The global explanations indicate that the important features are resources executing the activities. Local explanations provide some interesting insights: lower values of features such as the time since last event, the time since case start, increase the likelihood of the positive outcome (Figure 5(b),(d)). Some model features used in local explanations reduce its in-
Interpreting Predictive Process Monitoring Benchmarks

Fig. 6. Global explanations for index encoding for prefix lengths 10 and 20.

Terpretability and are referred to as engineered features by Lipton [8] e.g. the standard deviation of activity duration, minimum number of open cases when the event is executing.

Model explanations for encoded features. The global explanations of the models trained at distinct prefix lengths using index encoding (Figure 6), shows that the model uses activity or resource information about the prior events. While this model has lower AUC, the features it uses represent the control-flow of the trace. Models trained on lower prefix lengths use information about the resources executing activities at certain indexes (temporally ordered). Models trained on higher prefix lengths use information about the prior activities executed in the trace.

Fig. 7. Global and local explanations 10 and 25 for BPIC 2011 log.

Understanding the degree of business process knowledge used by the model: Model explanations further provide an understanding of the degree of business process knowledge used in the outcome prediction. The BPIC 2011 log has the outcome label set as positive based on the occurrence of an event.
We use the prefix length bucket and index encoding to identify features responsible for predicting the outcome. Prefix lengths of 10 and 25 are used to depict two scenarios as the case progresses. As seen in Figure 7, the model predominantly uses the static attributes or the attribute of the first event that remain the same throughout the execution of the trace. Use of case attributes indicates limited use of event attributes or the control-flow perspective. Here, the post hoc explanations provide insight on the extent of the use of the information encoded in the event log.

3.4 Interpreting Remaining Time Prediction

This section presents the results of the explanations for the task of remaining time prediction. The task of remaining time prediction can be defined in the following way: “Given an event log of complete cases of a business process, and a prefix case of this process as obtained from an event stream, we want to predict a performance measure of such prefix case in the future” [3]. An example of such a performance measure is the prediction of the time that remains for a case to be completed. The regression model XGBoost is used with the bucketing and encoding methods detailed in Section 3.1. Due to space limitations, we present the key insights drawn from the model explanations. Our detailed analysis, as well as the source code, is available at [https://git.io/Je1XZ](https://git.io/Je1XZ).

Fig. 8. Analysis of the bpic2011 event log, using XGBoost algorithm to make remaining time with local and global explanations.

Understanding models with the highest performance: To verify if performance metrics such as MAE is reliable, we analyze the best performing predictive model with low MAE on the event logs bpic2011 and bpic2012o. Figures 8 and 9 show this analysis, respectively.
In Figure 8, one can see that the algorithm relies on features such as Diagnosis Treatment Combination ID, Treatment code and Diagnosis in order to determine the remaining time of the case. For a regression task, like time prediction, this explanation indicates that the model is relying mostly on static features; in other words, features that do not change much throughout the lifetime of a case. The usage of static features for regression thus suggests that the process execution does not rely on the case activities or case execution and that a rule-based system for predicting or computing remaining time could suffice.

The analysis of bpic2012o leads to different observations. In this event log, the explanations are more accurate as the most important features used by the model have a positive impact on the prediction of the remaining time of the process. Some of the features identified as highly relevant are either dynamic, like agg_opencases and agg_resource, or time-dependent, such as agg_elapsed time. Although the features agg_opencases and agg_elapsed time are not part of the original event log and are additionally introduced, these features have a positive impact on the performance of the classifier. One potential problem with adding engineered features is that they might contribute to the loss of interpretability of the regression model [8]. However, the features are engineered in a meaningful way and are aligned with understanding the business process. The features further enable the extraction of meaningful insights from the regression model that enable us to assess the trustworthiness of the model.

**Fig. 9.** Analysis of the bpic2012o event log, using XGBoost algorithm with local and global explanations.

**Understanding encoding techniques:** The aggregation and prefix encoding further apply one-hot encoding to encode the activities, resources and other categorical data in the event log [3]. This specific encoding method increases the size of the dataset exponentially, because each attribute value of a feature,
becomes a new feature by itself with the possible value of 0 or 1. For instance, a feature $F$ with three attributes values $f_1$, $f_2$ and $f_3$ will be represented as three new binary features. This encoding of features generates very sparse datasets, which impacts negatively, both the performance metrics of the prediction model and the ability to generate interpretations out of this data. This problem was noticed in the three datasets analysed: bpic2011, bpic2012o and bpic2015.

Consider Figure 8, where a part of the training file is used to illustrate the one-hot encoding. The original dataset increased from approximately 20 features to 823 features with this representation. Further, when it comes to extracting local explanations, a majority of the explanations were statements of the following: if feature X is absent (value $\leq 0$), then it influences the remaining time prediction. When a dataset is so sparse, then it is reasonable to have such type of outcomes in terms of explainability. The question that remains is: to what extent can this provide a meaningful understanding of why the predictive model made a certain prediction?

This analysis is further extended to the bpic2012o event log (Figure 9). In this event log, the predictive model learned to ignore the sparse features and used time-dependent features or dynamical features. It raises an important question on the need to add sparse vectors and raise the complexity of the model input, when the core of the regression model is in a limited number of time-dependent features. In summary, one-hot encoding is a technique that should be used only for very large datasets. For event logs with small size, this technique is not advantageous to represent the data.

**Understanding the degree of business process knowledge used by the model:** An understanding of the business process is vital when it comes to the generation of predictive models because it enables the assessment of whether a performance metric makes a good representation of the underlying business process or not. For example, in the bpic2011 event log, a set of engineered features, such as max/mean/std. deviation of elapsed time, were added to enrich the event log. Paradoxically, the regression models did not use a single time-dependent feature in the prediction of the remaining time of the process. Figure 8, indicates that the most significant feature for remaining time prediction is the Diagnosis Code = DC822. In our analysis, we discovered that 82% of the events associated with the diagnosis code had the feature elapsed time = 0, which means that the activity starts and ends immediately. Lack of business process knowledge limits the ability to provides us deep insights about the relationship between these static diagnosis codes, with 0-valued elapsed times and its relation with the remaining time of the process.

**4 Lessons and Challenges**

In this section, we summarize the lessons based on our analysis of two predictive process benchmarks [3,1]. The models were analysed to assess: 1) the reliability of using a performance metric to assess the reliability of the predictive model,
2) the reliability of representing the process and the log using different encoding techniques, and 3) the degree of business process information used by the predictive model. From this analysis, we have learned three important lessons with respect to predictive process monitoring:

– **Lesson 1: Understand the underlying mechanisms of the business process.** Mapping business processes to an event log that captures the fundamental business knowledge is a complex and challenging task. The mapping entails a deep understanding of the underlying logic and execution of the business process and its connection to the event log data. This understanding is necessary to avoid data leakage problems, such as the ones found in the outcome-oriented predictions using BPIC2015 log. Further, a model that uses information about events that occur much later in a case execution to make a current prediction can be unreliable as seen in a single bucket model trained on BPIC2012 event log. It will also prevent the incorporation of engineered features that will add complexity to the event log and also limit the interpretability of the predictive model.

– **Lesson 2: Build Interpretable Models.** The goal of predictive process monitoring is to provide business users with interesting and useful insights about the underlying business executions. As we have seen in this analysis, relying on performance measures is not adequate to guarantee a good predictive model. We have to build models that are easily understood to a business user and can provide additional insights into why a particular prediction was made. Providing such an insight is, however, a very challenging task, and remains as an open research question in the scientific community [5].

– **Lesson 3: Provide a standard way of representing features.** Feature construction and representation requires both knowledge of business process and data mining techniques. As we have seen in our analysis, the majority of the encoding methods has its advantages and challenges in what concerns the accurate representation of the business process. Our analysis showed that the one-hot-encoding technique generated very sparse event logs, which impacted negatively in the explainability power of the prediction model. The aggregation encoding is a method that loses information from the process that could be valuable from an interpretability point of view. This leads to the challenge and dilemma of representing event log data in a way that it is interpretable and accurate.

5 **Conclusions**

The goal of this paper is to emphasise the importance of interpreting the predictive models in the context of predictive process monitoring. We reviewed two existing benchmarks in predictive process monitoring, and presented many examples of accurate models that could potentially be biased or unreliable. Findings drawn from the work indicate several challenges in building a reliable model: data leakage, use of relevant events, use of interpretable features, and the degree of the use of process knowledge in predictive process monitoring. Hence,
we suggest to incorporate interpretability in addition to the evaluation using conventional quality metrics (such as accuracy) in predictive process analytics. As a first step into future work, it is important to develop a systematic approach for incorporating interpretability in predictive process analytics.

References

1. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. TKDD 13(2) (2019) 17:1–17:57
2. Evermann, J., Rehse, J., Fettke, P.: Predicting process behaviour using deep learning. Decision Support Systems 100 (2017) 129–140
3. Verenich, I., Dumas, M., Rosa, M.L., Maggi, F.M., Teinemaa, I.: Survey and cross-benchmark comparison of remaining time prediction methods in business process monitoring. ACM TIST 10(4) (2019) 34:1–34:34
4. Lakkaraju, H., Kamar, E., Caruana, R., Leskovec, J.: Faithful and customizable explanations of black box models. In: Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society, AIES. (2019) 131–138
5. Rudin, C.: Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. Nature Machine Intelligence 1(5) (2019) 206–215
6. Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., Pedreschi, D.: A survey of methods for explaining black box models. ACM Comput. Surv. 51(5) (August 2018) 93:1–93:42
7. Rehse, J., Mehdiyev, N., Fettke, P.: Towards explainable process predictions for industry 4.0 in the dfki-smart-lego-factory. KI 33(2) (2019) 181–187
8. Lipton, Z.C.: The mythos of model interpretability. CACM 61(10) (2018) 36–43
9. Molnar, C.: Interpretable Machine Learning: A Guide for Making Black Box Models Explainable. Leanpub (2018)
10. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. Annals of Statistics 29 (2000) 1189–1232
11. Goldstein, A., Kapelner, A., Bleich, J., Pitkin, E.: Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. Journal of Computational and Graphical Statistics 24(1) (2015) 44–65
12. Apley, D., Zhu, J.: Visualizing the effects of predictor variables in black box supervised learning models (2016)
13. Breiman, L.: Random Forests. Machine Learning 1573-0565 (2001) 5–32
14. Ascione, F., Bianco, N., Masi, R.D., Stasio, C.D., Mauro, G., Vanoli, G.: Artificial neural networks for predicting the energy behavior of a building category: A powerful tool for cost-optimal analysis. In Pacheco-Torgal, F., Granqvist, C.G., Jelle, B.P., Vanoli, G.P., Bianco, N., Kurnitski, J., eds.: Cost-Effective Energy Efficient Building Retrofitting. Woodhead Publishing (2017) 305–340
15. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. The Annals of Statistics 29(5) (2001) 1189–1232
16. Ribeiro, M.T., Singh, S., Guestrin, C.: ”why should I trust you?”: Explaining the predictions of any classifier. In: Proceedings of the 22nd ACM SIGKDD. (2016) 1135–1144
17. Kaufman, S., Rosset, S., Perlich, C.: Leakage in data mining: Formulation, detection, and avoidance. In: Proc. of the 17th ACM SIGKDD. (2011) 556–563