IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 12, 2021, accepted September 22, 2021, date of publication September 27, 2021,
date of current version October 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3115759

Stability Metrics for Enhancing the Evaluation of Outcome-Based Business Process Predictive Monitoring

JONGCHAN KIM AND MARCO COMUZZI
Department of Industrial Engineering, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea

Corresponding author: Marco Comuzzi (mcomuzzi@unist.ac.kr)

This work was supported in part by the 0000 Project Fund of Ulsan National Institute of Science and Technology (UNIST) under Project 1.210079.01.

ABSTRACT Outcome-based predictive process monitoring deals with predicting the outcomes of running cases in a business process using feature vectors extracted from completed traces in an event log. Traditionally, in outcome-based predictive monitoring, a different model is developed using a bucket containing different types of feature vectors. This allows us to extend the traditional evaluation of the quality of process outcome predictions models beyond simply measuring the overall performance, developing a quality assessment framework based on three metrics: one considering the overall performance on all feature vectors, one considering the different levels of performance achieved on feature vectors belonging to individual buckets, i.e., the stability of the performance across buckets, and one considering the stability of the individual predictions obtained, accounting for how close the predicted probabilities are to the cutoff thresholds used to determine the predicted labels. The proposed metrics allow to evaluate, given a set of alternative designs, i.e., combinations of classifier and bucketing method, the quality of the predictions of each alternative. For this evaluation, we suggest using either the concept of Pareto-optimality or a scenario-based scoring method. We discuss an evaluation of the proposed framework conducted with real-life event logs.

INDEX TERMS Process mining, predictive monitoring, outcome prediction, quality, stability.

I. INTRODUCTION

Process mining aims at extracting process-relevant information from so-called event logs [1], which contain data logged during the execution of business processes. It provides a broad range of analytical tools, from process model discovery, which discovers models of business processes from event logs, e.g., BPMN models, to conformance checking, which detects the extent to which events in a log fit a process model.

In process mining, an event log refers to multiple executions of a given business process, e.g., order-to-cash. An individual execution of a process, e.g., the handling of a particular order placed a customer, is called a process *case*. An *event* contains an id of the case to which it belongs, a timestamp, information about the task or activity that it represents, and possibly other attributes, such as the (human) resources who executed or supervised the task. The events related to a process case, ordered in time, form a *trace* of events.

Predictive monitoring recently has emerged in process mining as a set of techniques that aim at predicting various aspects of interests of a process using event log data, such as the next activity that will be executed in a running process case [2], time-related aspects [3], or the outcome of a case [4].

This paper focuses on the outcome prediction use case. In this prediction use case, each trace in an event log is associated with an outcome label (usually binary), which can be given or reconstructed from information in an event log. The objective is then to develop a model that can predict the value of the outcome label for running traces, that is, for process cases that have not terminated yet, using data from cases that have already completed their execution. Since process outcomes are captured by categorical labels, this requires the use of machine learning classification techniques [5].

Approaches to outcome prediction and, more in general, predictive process monitoring, have evolved in the literature

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

with a clear focus on performance improvement. Several authors have in fact developed quantitative predictive monitoring benchmarks using different classification techniques, encoding schemes, and event log types [4], [6], [7], comparing the overall performance, i.e., on all the traces in an event log, using standard measures, such as AUC, precision, or recall, in different predictive monitoring tasks. To improve the model performance, research also has focused on the engineering of new features and encoding methods for event log data, e.g., inter-case features [8], or adapting cutting-edge machine learning models, such as deep learning [3], to the problem at hand.

In this paper, we consider a broader notion of *quality* of the models obtained for outcome prediction that goes beyond the standard performance measures that have been adopted in the literature. There is no standard definition for the quality of a machine learning classification model in the literature and, therefore, different notions of model quality may be developed depending on the specific characteristics of a classification problem. For instance, in some cases the quality of the model is evaluated by probing whether it is either overfitted or underfitted to the samples in the training set [9], [10], while other approaches take a different perspective on the quality, taking data imbalance into consideration [11]. As far as classification in predictive process monitoring is concerned, the work by [12] is the only one that has provided a notion of model quality that goes beyond standard performance measures, focusing in particular on the temporal stability of predictions of process outcomes. Specifically, the authors have developed a set of metrics to define how stable outcome predictions are as a running trace evolves, i.e., more events are executed.

We take a different perspective on model quality, analysing the stability and performance of outcome predictions over *buckets* of trace prefixes. Typically, in a process outcome prediction model, in the offline phase trace prefixes are first extracted from events in a log, where a prefix of length l is constituted by features extracted from the first l events of a given process trace. Prefixes are then divided into a number of disjoint *buckets*, e.g., applying a clustering algorithm or grouping prefixes by length, i.e., the number of events from which prefixes are extracted. Then, a classifier is trained for each bucket. In the online phase, a running trace is first assigned to a bucket, then a prediction for it is obtained using the classifier trained for that particular bucket.

Intuitively, in some cases, good overall performance of an outcome-based prediction model on all prefixes may result from aggregating excellent performance on some buckets of prefixes and poor performance on some others. Even though the overall performance of such a model, e.g., AUC or accuracy, may appear acceptable, this model may not be a *good* model in many practical situations, since decisions taken using it for specific types of cases, i.e., the ones falling in buckets associated with low performance, are likely to be highly inaccurate. There is one more critical aspect to be considered when considering the practical applications of outcome-based predictive monitoring, which deals with the stability of predictions in respect of the probability thresholds chosen for classification. A decision tree in a binary classification problem, for instance, given a new observation, outputs a probability for it to be classified in each class. The class associated with a probability higher than 0.5 is then chosen as the classification label. The closer the highest classification probability to 0.5, the more likely such a classification to be *unstable*, i.e., to change with only a slight modification of either the input data or the training set from which the model was obtained. A similar rationale may be applied to other classification techniques, such as feed-forward neural networks or random forests. Generally, we argue that a decision-maker would be more confident when taking decisions using predictions obtained from a classifier that is stable.

In this paper, we assume that the quality of an outcome-based prediction model should be evaluated considering both the performance of individual classifiers within it, i.e., the ones obtained for each bucket of trace prefixes, and the stability of the performance inside buckets and across different buckets. More in detail, our quality framework relies on the following principles: first, the higher the performance of an outcome prediction model on individual buckets, the higher its quality. Second, individual classifiers should also be *stable*, i.e., the likelihood that a prediction made for a running case changes in respect of the value set for the classification threshold should be low. Then, a third principle considers the stability of the performance across classifiers created using different buckets. Specifically, we consider a low quality classifier yielding good and reliable prediction only on particular buckets, but outputs a bad and unreliable classification for other buckets.

Based on these principles, this paper develops a framework containing a set of metrics for evaluating the quality of an outcome prediction model in process predictive monitoring. Specifically, similarly to the notion of quality in other fields, such as quality of service (QoS) [13] or data quality [14], we consider the quality of an outcome prediction framework as a multi-dimensional concept defined by the following three metrics:

- Overall Bucket Performance (OBP), which focuses on the predictive performance of classifiers developed for each bucket;
- Intra-bucket Prediction Stability (IBS), which measures how distant are the actual prediction probabilities from the cutoff thresholds, as a proxy of their stability in respect of slight changes of the prediction model or input data;
- Cross-bucket Performance Stability (XBS), which measures the extent of the difference among the performance across different buckets in an outcome prediction model.

The proposed framework is evaluated considering real-world event logs publicly available and different combinations of classifiers and trace bucketing methods. The proposed framework can be used as an objective way of assessing the quality of given combinations of classifier and trace bucketing methods chosen for an outcome prediction model. Therefore, the metrics that we propose in this paper do not aim at substituting the performance evaluation metrics traditionally defined for classification techniques, but rather complement them in the specific context of outcome-based process predictive monitoring. From a practical standpoint, the proposed framework can aid decision-makers when assessing whether a prediction model outputs stable predictions across different buckets of input observations. Based on the type of bucketing chosen, such stability may assume a different meaning. For instance, if clustering-based bucketing is chosen, then the proposed framework assesses stability across different groups of similar traces, whereas if prefix length-based is chosen, stability is assessed across the amount of knowledge, i.e., events, known for each historic trace execution.

The paper is organised as follows. Section II presents the related work and Section III gives a formal introduction to the problem of developing predictive models of process outcomes using event logs. Section IV describes the metrics to evaluate the models and discusses how they can be used in practical scenarios. Section V presents the experimental evaluation. Lastly, section VI summarizes the findings along with implications for future work.

II. RELATED WORKS

Predictive process monitoring [15] concerns various prediction tasks, such as predicting the outcome of a process [4], [16], the next event of a running case [3], [6], or the remaining time until the termination of a running case [3]. In outcome-based predictive monitoring, the outcome of a case is binary in general. Approaches in the literature tend to define outcomes as the satisfaction of service level agreements or the satisfaction of linear temporal logic constraints defined on the order and occurrence of activities in a case [4]. While traces in an event log can be split into training and test set for learning, traces can also be bucketed to train the classifier intended to be built exclusively for the corresponding bucket, where samples in the same bucket share similar characteristics. In addition, features can be encoded to better characterise the data. Since predictive monitoring of process outcomes handles an event log as an input which is a set of the sequence of events, sequence information can be used to encode the features. For example, index-based encoding generates one feature for each attribute on each executed event [17].

To enhance the performance of predictions in predictive monitoring of process outcomes, most of the efforts have been made from the algorithmic side, such as selecting which type of classifiers or hyperparameters to use, and the feature engineering side, such as the feature encoding and feature selection. Tree-based classifiers such as random forest or extreme gradient boosting have successfully been adopted to predict process outcomes along with various bucketing techniques, such as prefix-length bucketing or clustering bucketing, and sequence encoding techniques such as index-based encoding and Hidden Markov Models (HMM)-based encoding [17], [18]. Most recently, the focus of the research community appears to have shifted to complex deep learning architectures [19], [20], which however appear more suited to use cases such as next activity or time prediction, and to generating and interpreting explanations for the output of process predictive monitoring [21]–[23].

An extensive research dedicated to the predictive monitoring of process outcomes has been conducted by comparing not only the classifiers from different families (random forest, extreme gradient boosting, logistic regression, and support vector machine) but also different trace bucketing and sequence encoding techniques [4]. For selecting optimal values or hyperparameters, a tuning-enhanced predictive process monitoring framework has been devised in [24], which evaluates the predictions using three metrics: accuracy, failure rate, and earliness. A search heuristic based on genetic algorithm has been used to efficiently scan the hyperparameter optimization space in [25]. Beyond the approaches from the algorithmic side, from a practical standpoint focusing on service level agreement, a hybrid metric to measure the reliability of predicting the service level agreement has been proposed in [26].

The evaluation of the quality of predictive models covers a broad scope of topics, such as the reliability and the stability of predictions. Overfitting is a critical problem related to the reliability in both classification and regression problems. In fact, overfitting not only makes the model less parsimonious by introducing irrelevant terms, but it also harms the predictive performance disturbing the way in which the values of the model parameters are calculated and bringing random errors and variations to the predictions [9]. Underfitting, the opposite concept of overfitting, is also a problem related to the reliability of predictions, since it prevents a model, while being learned, to properly investigate the underlying relationship among data samples [10].

Data imbalance also decreases the validity of predictions by masking the predictive performance of the minority class samples with the overall performance [11]. Along with the perturbations of learning data, learning with unlabeled input data and sensitivity analysis can also be used for estimating the reliability [27]. The transductive method is used for estimating the reliability, where a classifier is fitted to a modified training set [28]. Beyond the traditional reliability metrics, explainability in predictions has recently emerged as a new dimension of reliability metrics [29], [30].

The stability of predictions in supervised learning has been extensively discussed in various prediction tasks. In general, a classifier is said to be stable if its predictive performance does not vary with changes in input datasets, generally measured by the variance of the performance metrics, such as AUC [31], [32]. Depending on the context, a classifier can also be considered stable even if its predictive performance does not vary with not only the changes in datasets, but also changes in parameters, repeated trials, or time slots (in case of time series data). In case of selecting the training variables, the models for selecting variables are considered stable if they select similar variables after the repeated trials [33]. Especially in the case of decision trees, the issue of the stability of predictions has been highlighted, as the predictions provided by decision trees have been found unstable. In order to solve this problem, random forest, an ensemble of decision trees, has been developed [34], [35]. Other than ensemble methods, non-ensemble methods, such as Info-Fuzzy Network (IFN), has been developed to enable the interpretation of the results while preserving a high level of stability at the same time [36]. These kinds of approaches to mitigate the problems related to the stability of decision trees can also be applied to other classifiers or regression methods showing poor stability [34]. In order to estimate the stability of regression methods, such as multiple linear regression, support vector regression and artificial neural network, the bootstrapping method has been proposed [37].

In predictive monitoring of business processes, the concept of temporal stability of predictions has been introduced. Given the predictive performance provided for each prefix, the classifier is considered temporally stable if the classifier outputs similar predictions to successive prefixes [12].

III. PRELIMINARIES

An event log EL contains events. An event e is a tuple $e = \langle c, a, t, r, (d_1, v_1), \ldots, (d_m, v_m)\rangle$, where c is the case id, a is the activity to which the task recorded by this event belongs, t is the timestamp at which the event has been recorded, r is the resource that executed the task and $(d_1, v_1), \ldots, (d_m, v_m)$, with $m \geq 0$, are other domain specific attributes and their values. For instance, the event $e = (45, \mathit{assess}, 2020.1.2, \mathit{Alice}, \mathit{amount} = 1000, \mathit{type} = \mathit{deep})$ captures the fact that, in a process case associated with loan request number 45, the resource Alice has executed a deep assessment of a loan request of 1000 USD on January 2nd, 2020. The universe of all events is denoted by $\mathcal{E}$. We use a dotted notation to identify attributes of events, e.g., $e.c$ to identify the case id of event e.

The sequence of events generated in a given case forms a trace $\sigma = [e_1, \ldots, e_n]$, where $\forall i \in [1, n]$, $e_i \in \mathcal{E}$, and $\forall i, j \in [1, n]$, $e_i.c = e_j.c$, i.e., all events of a trace belong to the same case, and $\forall i \in [1, n-1]$, $e_i.t < e_{i+1}.t$, i.e., events in a trace can be ordered in time using the timestamp attribute. The universe of all traces is denoted by $\mathcal{S}$.

Given a trace σ and an integer $l < n$, the prefix function returns the first l events of σ, that is, $\mathit{prefix}(\sigma, l) = [e_1, \ldots, e_l]$. We refer to $P \subseteq \mathcal{S}$ as the set of prefixes that can be generated from the events in an event log EL. A prefix bucketing B_N of size N is a partition of the prefixes P in N subsets, that is, $B_N(P) = \{B_i\}_{i=1\ldots N}$ with $\bigcup_i B_i = P$ and $\bigcap_i B_i = \emptyset$.

A labeling function $y : \mathcal{S} \longrightarrow \mathcal{Y}$ is a function mapping a trace $\sigma \in \mathcal{S}$ (or any prefix derived from it) to its class label $y(\sigma) \in \mathcal{Y}$, with $\mathcal{Y}$ being the domain of the class labels. Typically, outcome predictions involve a binary outcome, that is, $\mathcal{Y} = \{0, 1\}$. Note that all prefixes generated from a trace σ have the same class label.

In the specific case of outcome-based predictive monitoring, predictions are made using a classifier that takes as input a fixed number of independent variables (*features*) and learns a function to estimate the dependent variable (class *label*). This implies that, in order to use the data in an event log as input of a classifier, each trace in the log must be encoded as a feature vector.

A sequence (or trace) encoder $f : \mathcal{S} \longrightarrow \mathcal{X}_1 \times \ldots \times \mathcal{X}_P$ is a function that takes a (partial) trace σ and transforms it into a feature vector in a D-dimensional vector space $\mathcal{X}_1 \times \ldots \times \mathcal{X}_D$ with $\mathcal{X}_d \subseteq \mathbb{R}, 1 \leq d \leq D$ being the domain of the d-th feature.

Given a bucketing B_N of prefixes in an event log, a process outcome classification model pom normally is constituted by N process outcome classifiers poc_i, with $i = 1, \ldots N$, each developed using the prefixes in a bucket B_i. A process outcome classifier poc is defined by a label predictor function lp that assigns a class label to a feature vector, i.e. $lp : \mathcal{X}_1 \times \ldots \mathcal{X}_P \longrightarrow \mathcal{Y}$.

In binary classification, classifiers such as neural networks or tree-based classifiers normally output a classification probability for each of the available class labels. The class label associated with the highest probability is then chosen as the predicted class label. Note that this implicitly means to set the value of a cutoff threshold for assigning class labels at 0.5. Since the classification probabilities sum to 1, the highest probability is, in fact, always higher than 0.5. Given the classification probabilities, a poc can also be defined by a classification probability estimator cpe, which is a multi-valued function assigning a probability value for each of the two possible class labels, i.e., $cpe : \mathcal{X}_1 \times \ldots \times \mathcal{X}_D \longrightarrow \{0, 1\} \times \{0, 1\} \subseteq \mathbb{R}^2$, with $cpe[f(\sigma)] = \{p_0, p_1\}$. Note that, given $p_{max} = \max\{p_0, p_1\}$ the label predictor function can alternatively be defined as follows:

$$lp(\sigma) = \begin{cases} 1 & \text{if } p_{max} = p_1 \\ 0 & \text{otherwise} \end{cases}$$

IV. MODEL QUALITY FRAMEWORK

Figure 1 depicts the typical application scenario of outcome-based predictive monitoring. As commonly recognised in the literature [38], the events of completed traces in an event log are first encoded to obtain the set of all prefixes. Then the prefixes are divided into a number of buckets. The literature considers mainly three different types of bucketing [4], [17], [18], [39]:

- *Clustering*, where buckets are defined by applying a clustering algorithm to the set of prefixes;
- *Prefix-length*, where buckets contain all the prefixes of a given length;
- *State-based*, where buckets contain all the prefixes that have reached a certain state during the process execution. This way of bucketing relies on the existence of a process model and it is not considered further in this paper.

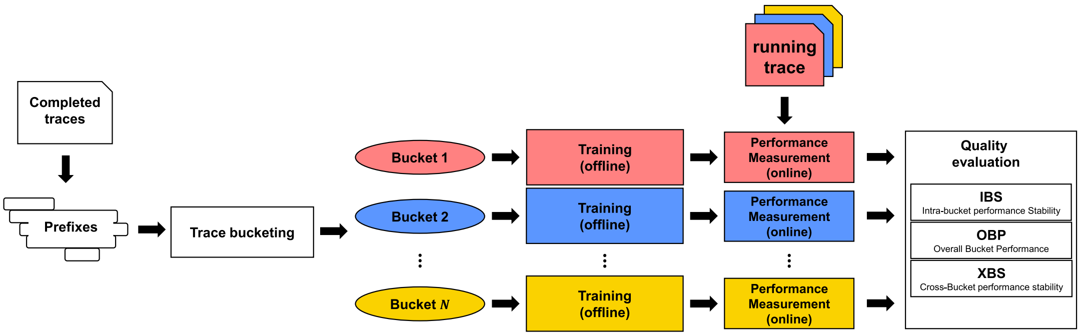
Completed traces
Prefixes
Trace bucketing
running trace
Bucket 1
Bucket 2
Bucket N
Training (offline)
Training (offline)
Training (offline)
Performance Measurement (online)
Performance Measurement (online)
Performance Measurement (online)
Quality evaluation
IBS
Intra-bucket performance Stability
OBP
Overall Bucket Performance
XBS
Cross-Bucket performance stability

FIGURE 1. Outcome-predictive monitoring framework, with model quality evaluation.

A process outcome classifier is trained for each bucket of prefixes obtained. In the online phases, running (incomplete) traces are first classified into a bucket. After the bucket is assigned, the classifier trained for the chosen bucket is used to predict the outcome of the running trace. From the standpoint of our framework, the outcome of a classifier is the predicted probabilities p_0 and p_1 given by the classification probability estimator *cpe* and the predicted outcome label y given by the label predictor function *lp*. These, together with the ground truth outcome labels $\bar{y}$ available in the event log, are the input of the proposed quality evaluation framework. As remarked earlier, this is comprised of three elements:

- Evaluation of OBP: measures the predictive performance of a classifier within each bucket;
- Evaluation of IBS: measures how much the predicted probabilities within each bucket are diffused with respect to both the distance between each predicted probability and the distance between the classification cutoff and each predicted probability;
- Evaluation of XBS: measures how much the performance across different buckets are diffused, considering the different sizes of each bucket.

Next, we define the metrics OBP, IBS, and XBS of our framework.

A. INTRA-BUCKET PREDICTION STABILITY (IBS)

The IBS metric considers how close the prediction probabilities p_0 and p_1 are to the cutoff threshold, which is 0.5 in the case of binary classification considered in this paper. The closer a prediction probability to 0.5, the more likely it is to change if the experiment is repeated, for instance because cross-validation is adopted, or if the input data vary slightly, for instance because some observations are left out from the training set. Conversely, prediction probabilities distant from 0.5, i.e., close to 1 or 0, are considered stable, because they are not likely to change in a different repetition of the experiment.

Regarding the design of the metric, since we consider bucketing techniques to group the prefixes extracted from an event log and we are looking at the stability of predicted labels across different repetitions of experiments, we call this first metric Intra-Bucket prediction Stability (IBS). The value of the IBS metric for a given combination of classifier and bucketing method, i.e., a process outcome prediction model (*pom*), is calculated as follows: first, we calculate an IBS value for each bucket B_i; then, we aggregate the values $IBS(B_i)$ to obtain the overall value $IBS(pom)$.

While we could have considered simple statistical measures, such as standard deviation or mean absolute deviation of the difference between predicted probabilities and the cutoff thresholds, for calculating the value of this metric we consider a more refined approach inspired by the *Western Electric Rule* [40] in the field of statistical process control. This rule has been created to evaluate the quality of the output of a manufacturing production line, helping workers to monitor and understand the control charts. In a nutshell, it is built by comparing the quality of the output to the expected quality, dividing the differences obtained according to their absolute value, and penalizing, i.e., weighting more heavily, the higher values of these differences. Similarly, while defining IBS, for all the prefixes σ in a bucket B_i, we calculate the absolute difference between $p_{max} = \max\{p_0, p_1\}$, which determines the label y assigned to σ, and the cutoff threshold 0.5. Then, a weight w, with $w \in (0, 1)$ is used to penalize more heavily the differences that are higher. In the experiments, we consider the value $w = 0.05$.

Based on the rationale discussed above, given a bucketing $B_N = \{B_i\}$ and noting the number of prefixes in a bucket B_i as $|B_i|$, we calculate the stability of predictions within one bucket B_i as follows. First, Eq. 1 defines the stability of the prediction for an individual prefix σ. This is then normalised (see Eq. 2) using its minimum and maximum value in a bucket B_i.

$$sta(\sigma) = \begin{cases} (1+2w)\cdot p_{max} & \text{if } 0.4 < |p_{max}-0.5| \le 0.5, \\ (1+w)\cdot p_{max} & \text{if } 0.3 < |p_{max}-0.5| \le 0.4, \\ p_{max} & \text{if } 0.2 < |p_{max}-0.5| \le 0.3, \\ (1-w)\cdot p_{max} & \text{if } 0.1 < |p_{max}-0.5| \le 0.2, \\ (1-2w)\cdot p_{max} & |p_{max}-0.5| \le 0.1 \end{cases} \tag{1}$$

$$sta_{norm}(\sigma) = \frac{sta(\sigma) - \min_{\sigma \in B_i}\{sta(\sigma)\}}{\max_{\sigma \in B_i}\{sta(\sigma)\} - \min_{\sigma \in B_i}\{sta(\sigma)\}} \tag{2}$$

Finally, the IBS value of a bucket B_i is calculated as the weighted average of the normalised stability of prefixes in it:

$$IBS(B_i) = \frac{1}{|B_i|} \sum_{\sigma \in B_i} sta_{norm}(\sigma), \tag{3}$$

The IBS of an outcome prediction model *pom* can now be defined as the average of the IBS of all buckets in it:

$$IBS(pom) = \frac{1}{N} \sum_{i=1}^{N} IBS(B_i) \tag{4}$$

B. OVERALL BUCKET PERFORMANCE (OBP)

The objective of OBP is to evaluate the overall test set performance obtained by a process outcome prediction model *pom*. Therefore, to define an OBP metric, we simply consider the average across all buckets of a given measure of prediction performance *perf* , such as accuracy, F1-score, or AUC.

Let us consider $perf(B_i)$ as the average performance of a classifier poc_i achieved over the prefixes in a bucket B_i, according to a given performance measure normalised between 0 and 1, such as accuracy, F1-score, or AUC. The OBP of an outcome prediction model *pom* is the average performance across all buckets weighted by the bucket size:

$$OBP(pom) = \frac{1}{T} \sum_{i=1}^{N} |B_i| \cdot perf(B_i) \tag{5}$$

where T is the number of prefixes generated, that is, the sum of the size of all buckets: $T = \sum_i^N |B_i|$.

Note that this type of metric is the one normally considered by predictive monitoring benchmarks in the literature to evaluate the performance. For instance, [17], [18] consider AUC, while [38], [41] consider average F1-score as performance measure. In the evaluation of our work, we consider AUC for measuring the performance, as also recommended by literature [42].

C. CROSS-BUCKET PERFORMANCE STABILITY (XBS)

XBS measures the extent to which the performance achieved by a process outcome prediction model *pom* on different buckets varies across buckets. The design of the XBS metric is inspired by the temporal stability metric for outcome-based predictive monitoring, which calculates the stability of a classifier measured by two consecutive prediction scores [12]. What makes the formula for XBS different from the formula for temporal stability is that the former considers prediction scores from any of the two buckets of different sizes, while the latter only considers prediction scores from two events that are sequentially located next to each other.

XBS is calculated as shown in Eq. 6. Firstly, the performance of two different buckets is considered, and their absolute difference is calculated. Then, this is weighted considering the relative number of prefixes in the two buckets. This procedure is repeated for all pairs of buckets in *pom*. The weighted performance differences are summed up and weighted by the number $N-1$ of pairs of buckets (N buckets, in fact, result in $N-1$ pairs of buckets to compare). Finally, the weighted sum obtained is subtracted to 1 to obtain the value of $XBS(pom)$. Note in fact, that the metric value should be higher when the differences of performance across buckets are lower.

$$XBS(pom) = 1 - \frac{1}{N-1} \sum_{t=2}^{N} \sum_{r=1}^{t-1} \times \left[\frac{|B_t| + |B_r|}{T} \cdot |perf(B_t) - perf(B_r)| \right] \tag{6}$$

D. METRIC INTERPRETATION

In this section, we describe how to apply the proposed framework. The framework allows comparing the quality of different combinations of classifier and bucketing method (i.e., process outcome models *pom*). After having evaluated the metrics IBS, OBP, and XBS for each model, we propose two different methods to interpret the values obtained. The first method proposes to identify the models that are Pareto optimal, whereas the second method allows a more qualitative scenario-based analysis of the values obtained.

1) PARETO OPTIMALITY

Pareto optimality, or Pareto efficiency, is the general condition where no individual or preference criterion can be better off without making at least one individual or preference criterion worse off [43]. Given a set of alternatives that can be scored along with a multi-criteria definition of quality, Pareto optimality is a typical way to select the best alternatives. A Pareto optimal alternative, in fact, is such that no alternatives exist that improve at least one criterion without decreasing the value of at least another one. In the context of the proposed framework, the alternatives are different models *pom*, i.e., combinations of classifier and bucketing method, whereas the criteria are the values of the three metrics IBS, OBP and XBS.

While Pareto optimality can be a reasonable way to identify *good* quality alternatives, there are cases in which it clearly does not indicate good model quality. In fact, Pareto optimal alternatives often may result from extreme combinations of values achieved for the quality criteria. Let us consider the case of a model A for which the three metrics (IBS, OBP, XBS) evaluate to (0.999, 0.2, 0.2). Such a model is likely to be Pareto optimal: while many alternatives are likely to have values of OBP and XBS higher than A, they are not likely to improve or even match A's value of IBS, which is extremely high. Therefore, the model A would be Pareto optimal even though it scores rather poorly on two of the three metrics in the proposed framework. To overcome this issue with Pareto optimality, we define next a method to evaluate alternatives based on a more qualitative assessment of the value assumed by the metrics.

2) SCENARIO-DRIVEN SCORING OF ALTERNATIVES

Table 1 describes 8 possible qualitative scenarios resulting from the combination of the values assumed by the three

TABLE 1. Description of scenarios for scenario-based scoring.

Scenario	IBS	OBP	XBS	Description	Score
Scenario 1	High	High	High	The best scenario: performance of most buckets is high, and the classifier has a high chance of consistently showing overall high performance across different repetitive trials of the experiment.	5
Scenario 2	High	High	Low	While the performance of most buckets is above average, for few buckets it is below average. Nevertheless, the classifier has a high chance of consistently showing overall high performance across different repetitive trials of the experiment.	4
Scenario 3	Low	High	High	While the performance of most buckets is consistently above average, the classifier is likely to output inconsistent predictions, which may change across different repetitive trials of the experiment	4
Scenario 4	Low	High	Low	While the performance of most buckets is above average, for few buckets it is below average. Moreover, the classifier is likely to output inconsistent predictions, which may change across different repetitive trials of the experiment.	3
Scenario 5	Low	Low	Low	While the performance on few buckets can be high, the performance achieved is below average on most buckets. Moreover, the classifier is likely to output inconsistent predictions, which may change across different repetitive trials of the experiment.	2
Scenario 6	High	Low	Low	While the performance on few buckets can be high, the performance achieved is below average on most buckets. Moreover, the predictions of the classifier are stable, i.e., unlikely to change across different repetitive trials of the experiment	1
Scenario 7	Low	Low	High	The performance achieved is below average on most buckets.While the performance on few buckets can be high, the performance achieved is below average on most buckets. Moreover, the classifier is likely to output inconsistent predictions, which may change across different repetitive trials of the experiment.	1
Scenario 8	High	Low	High	The worst-case scenario: The performance achieved on all buckets is consistently below average, and the predictions of the classifier are stable, i.e., unlikely to change across different repetitive trials of the experiment.	0

metrics IBS, OBP and XBS for a given model. Each scenario assigns a score from 0 to 5 to each combination of classifier and bucketing method (i.e., a model), with 0 corresponding to *poor* model quality and 5 to *good* model quality. Each metric can assume the value *low* or *high*. These are established comparing the value of the metric assumed for a model with the average of the values of the same metric across all models considered for a given event log (*high* if above average, *low* if below average).

Note that, as general principles:

- For OBP, a high value signifies that the performance achieved by a model on most buckets is at least above average, while still there can be few buckets for which the performance is below average;
- For IBS, a high value signifies that the prediction probability for most observations is far away from the 0.5 cutoff threshold. Therefore, most predictions are stable, i.e., they are not likely to change in a different repetition of the experiment and/or with small perturbations of the input data. The opposite happens for low values of IBS;
- For XBS, a high value signifies that on average the differences among the performance achieved on different buckets are limited. Therefore, the classifier is fairly balanced, predicting most types of prefixes with similar accuracy. When XBS is low, there can be the case that a few buckets show excellent performance, while most other buckets showing poor one (or vice versa).

The best scenario (Scenario 1, associated with the highest score 5) is the one in which all metric values for a model are *high*. This corresponds to combinations of classifier and bucketing method for which:

- the performance in most buckets is high (as interpreted from the high value of OBP);
- the classification probabilities are normally far from the cutoff threshold, which leads to stable (consistent) classifications across different trials of the experiments (as interpreted from the high value of IBS);
- the performance differences between buckets are normally low (as interpreted from the high value of XBS).

Conversely, the worst scenario (Scenario 8) is the one for which, obviously, the value of OBP is low, i.e., the performance of the classifier is rather low on most buckets, but for which, perhaps counter-intuitively, the value of IBS and XBS are both high, i.e., the low performance is consistent across buckets (high XBS = low performance differences across buckets) and bad predictions are rather stable, i.e., not likely to change with small perturbations of the input data or the model across different repetitions (i.e., high IBS).

Scenario 2 and 3 are considered the second-best possible scenarios (associated with a score equal to 4) because they are characterised by high values of OBP, which means that the performance on most buckets is high. However, they suffer from either one of the following problems: the performance level is not particularly stable across buckets (low XBS, for Scenario 2), or the individual predictions are not particularly stable (low IBS, for Scenario 3). Scenario 4 is scored below Scenario 2 and 3 because it is still characterised by high values of OBP, but it suffers from both problems enumerated above.

Scenario 5 is characterised by the value *low* for all the metrics. Perhaps counter-intuitively, again, it is associated with a higher score than other scenarios for which at least one of the metrics evaluates to high (Scenario 6 and 7, and 8). This is due to the fact that the combination of low OBP and low XBS signals the case in which the low overall performance of the classifier is due to the combination of excellent performance on some buckets and poor performance on many other buckets. That is, this scenario signals that at least on some buckets the classifier shows high performance. Additionally, the low value of IBS signals that the predictions are not particularly stable and, therefore, they may change, possibly improving, across different repeated trials of experiments.

Finally, scenario 6 and 7 are similar to Scenario 2 and 3, but with the exception that the overall performance across buckets OBP is now *low*, which means that these scenarios show overall low performance either stable across buckets (Scenario 6) or associated with highly stable predictions, i.e., unlikely to change across different repetitions of experiments (Scenario 7).

V. EVALUATION

Section V-A describes the event logs and classifiers that we considered for the evaluation. The experimental results are reported and discussed in Section V-B.

A. DATASETS AND EXPERIMENTAL SETUP

We have evaluated the proposed framework using publicly available[1] event logs. The event logs used for the experiments are the ones published by the BPIC (Business Process Intelligence Challenge) in 2011 (4 event logs) and BPIC 2015 (5 event logs), and the Sepsis event logs (3 event logs). These have been chosen because they contain outcome labels and have been used often by previous research on predictive process monitoring [4], [6], [24].

All prefixes obtained from events in an event log are encoded using index-based encoding [17]. This type of encoding has been chosen because it is lossless, it is widely adopted in the literature, and it requires no particular configuration (as opposed to, for instance, the aggregation encoding, which would require to specify a different aggregation method for each attribute or attribute type). As bucketing methods, we consider prefix-length and clustering. The former creates buckets containing all prefixes having the same length, i.e., same number of events. The maximum prefix length is set to 10. In the latter, k-means clustering with $k = 5$ is employed and prefixes are clustered based on Euclidean distance of vector of the count of occurrences of the activity label values.

We consider the following 3 classifiers: Random Forest (RF), Gradient Boosting Machine (GBM) and Extreme Gradient Boosting (XGB). The hyperparameters of the classifiers have been tuned using Tree-structured Parzen Estimator (TPE), which is performed separately for each combination of the dataset and trace bucketing method, performing 3-fold cross-validation for each configuration of hyperparameter values to pick the best-performing configuration [44].

For RF, the optimal values of the hyperparameter, *max_features*, are selected in the following interval:

$$max\_features \in [0, 1]$$

For GBM, the optimal values of the three hyperparameters *learning_rate*, *min_samples_split* and *max_depth* are selected in the following intervals: $learning\_rate \in [0, 1]$, $min\_samples\_split \in \{x \in \mathbb{N} | 4 \leq x \leq 30\}$, $max\_depth \in \{x \in \mathbb{N} | 4 \leq x \leq 30\}$.

For XGB, the optimal values of the four hyperparameters *learning_rate*, *subsample*, *max_depth*, *colsample_bytree* and *min_child_weight* are selected in the following intervals: $learning\_rate \in [0, 1]$, $subsample \in [0.5, 1]$, $max\_depth \in \{x \in \mathbb{N} | 4 \leq x \leq 30\}$, $colsample\_bytree \in [0.5, 1]$, $min\_child\_weight \in \{x \in \mathbb{N} | 1 \leq x \leq 6\}$.

When training/testing a classifier on a given bucket, we consider a temporal split of traces to separate the samples into 80:20 (training:test). As for performance measure *perf* to evaluate the quality metrics defined in the proposed framework, we consider the area under the receiver operating characteristic curve (AUC), which has been considered consistently by other outcome-based prediction models proposed in the literature [4], [17]. The code to reproduce the experiments is available at https://github.com/paai-lab/bucket-stability-outcome-prediction.

B. EXPERIMENTAL RESULTS

We first analyse the quality of the alternatives considered in this experimental evaluation using the Pareto optimality. Table 2, 3, and 4 show the results achieved for the three metrics in the proposed framework for the different combinations of classifier and bucketing method. The performance of the Pareto optimal alternatives is highlighted in boldface. Note that Pareto optimality is calculated for each different class of event logs (i.e., considering all the results shown in each individual table). The largest number of Pareto optimal alternatives have GBM as a classifier. In the BPIC 2015 and the sepsis event logs, the smallest number of Pareto-optimal alternatives have RF as a classifier, while for the BPIC 2011 event log, the smallest number of Pareto-optimal alternatives has XGB as a classifier. As far as bucketing methods are concerned, it can be concluded that it is not possible to identify one of the analysed alternatives (prefix-length or clustering) as more likely to be associated with a Pareto-optimal outcome.

[1] https://data.4tu.nl/portal

TABLE 2. IBS, OBP, and XBS of predictions using BPIC 2011 event log.

		Prefix-length bucketing			Clustering bucketing		
		IBS	OBP	XBS	IBS	OBP	XBS
BPIC2011_1	RF	0.503	0.875	0.952	0.529	0.860	0.903
	XGB	0.671	0.858	0.945	0.634	0.849	0.858
	GBM	0.955	0.889	0.953	0.722	0.861	0.908
BPIC2011_2	RF	**0.660**	**0.923**	**0.978**	0.579	0.871	0.825
	XGB	0.750	0.840	0.971	0.840	0.802	0.877
	GBM	**0.715**	**0.904**	**0.976**	**0.952**	**0.878**	**0.923**
BPIC2011_3	RF	**0.680**	**0.945**	**0.976**	**0.549**	**0.918**	**0.987**
	XGB	0.800	0.912	0.956	**0.812**	**0.899**	**0.958**
	GBM	**0.970**	**0.949**	**0.975**	**0.981**	**0.875**	**0.751**
BPIC2011_4	RF	**0.612**	**0.874**	**0.985**	0.665	0.875	0.934
	XGB	0.774	0.776	0.964	0.597	0.849	0.957
	GBM	**0.808**	**0.861**	**0.987**	**0.992**	**0.823**	**0.759**

TABLE 3. IBS, OBP, and XBS of predictions using BPIC 2015 event log.

		Prefix-length bucketing			Clustering bucketing		
		IBS	OBP	XBS	IBS	OBP	XBS
BPIC2015_1	RF	0.404	0.644	0.977	0.409	0.608	0.906
	XGB	0.794	0.573	0.941	0.766	0.612	0.914
	GBM	0.964	0.615	0.956	0.961	0.627	0.939
BPIC2015_2	RF	0.548	0.724	0.976	0.380	0.667	0.866
	XGB	**0.686**	**0.740**	**0.993**	**0.709**	**0.682**	**0.618**
	GBM	**0.988**	**0.713**	**0.978**	**0.990**	**0.601**	**0.853**
BPIC2015_3	RF	0.585	0.657	0.957	0.630	0.612	0.908
	XGB	0.877	0.655	0.958	0.764	0.632	0.835
	GBM	0.982	0.661	0.948	**0.983**	**0.642**	**0.948**
BPIC2015_4	RF	0.517	0.715	0.990	**0.474**	**0.707**	**0.997**
	XGB	0.683	0.686	0.990	**0.784**	**0.681**	**0.971**
	GBM	0.952	0.680	0.953	**0.977**	**0.645**	**0.956**
BPIC2015_5	RF	0.437	0.688	0.982	0.472	0.690	0.959
	XGB	0.718	0.683	0.978	**0.410**	**0.712**	**0.989**
	GBM	**0.962**	**0.669**	**0.988**	**0.966**	**0.666**	**0.951**

Generally, it can be noted that alternatives involving GBM are associated with the highest values of IBS for a given event log, which points to the fact that GBM is most likely to output stable (consistent) predictions. In contrast, the IBS of alternatives involving RF is, most of the times, the lowest for all event logs, which means that RF is most likely to exhibit inconsistent predictions. This finding prompts us to be cautious when using RF, as it is likely to fail at giving stable predictions, even though its overall performance often can be high. This is a remarkable result, considering in particular that RF is often considered a well-performing classifier in process predictive monitoring [6], [17] and other classification use cases [45], [46].

In addition, the results of Table 2, 3, and 4 also show that the values of OBP and XBS are often positively correlated, i.e., high OBP often appears together with high XBS. This is mainly due to the fact that high overall performance OBP is achieved only if the performance across all buckets is high, whereas below-average OBP can be achieved even if few very poor-performing buckets exist. There can be few but critical exceptions to this situation. For example, for the event log BPIC2015_2, in the case of clustering bucketing and XGB classifier, OBP is the highest, while XBS is the lowest. If we drill down to the performance of discrete buckets, the low XBS value for XGB is due to the exceptional difference between the highest-performing bucket (0.662) and the lowest-performing one (0.214).[2]

TABLE 4. IBS, OBP, and XBS of predictions using the sepsis event log.

		Prefix-length bucketing			Clustering bucketing		
		IBS	OBP	XBS	IBS	OBP	XBS
sepsis_cases_1	RF	0.404	0.644	0.977	0.660	0.463	0.943
	XGB	0.794	0.573	0.941	0.856	0.384	0.858
	GBM	0.964	0.615	0.956	**0.992**	**0.513**	**0.928**
sepsis_cases_2	RF	0.548	0.724	0.976	0.804	0.808	0.967
	XGB	**0.686**	**0.740**	**0.993**	**0.918**	**0.888**	**0.971**
	GBM	**0.988**	**0.713**	**0.978**	**0.981**	**0.843**	**0.951**
sepsis_cases_4	RF	0.437	0.688	0.982	**0.682**	**0.717**	**0.987**
	XGB	0.718	0.683	0.978	0.848	0.692	0.967
	GBM	**0.962**	**0.669**	**0.988**	0.973	0.718	0.937

TABLE 5. Score table of predictions using BPIC2011 dataset.

		Prefix-length bucketing	Clustering bucketing	Total
BPIC2011_1	RF	2	1	3
	XGB	2	2	4
	GBM	4	1	5
BPIC2011_2	RF	4	3	7
	XGB	0	1	1
	GBM	4	5	9
BPIC2011_3	RF	4	4	8
	XGB	4	5	9
	GBM	5	4	9
BPIC2011_4	RF	1	4	5
	XGB	1	1	2
	GBM	0	1	1
Total		31	32	

TABLE 6. Score table of predictions using BPIC2015 dataset.

		Prefix-length bucketing	Clustering bucketing	Total
BPIC2015_1	RF	1	2	3
	XGB	1	0	1
	GBM	1	0	1
BPIC2015_2	RF	4	3	7
	XGB	4	3	7
	GBM	5	1	6
BPIC2015_3	RF	2	1	3
	XGB	1	1	2
	GBM	1	0	1
BPIC2015_4	RF	4	4	8
	XGB	4	5	9
	GBM	4	0	4
BPIC2015_5	RF	4	4	8
	XGB	4	4	8
	GBM	0	5	5
Total		40	33	

TABLE 7. Score table of predictions using sepsis cases dataset.

		Prefix-length bucketing	Clustering bucketing	Total
sepsis_cases_1	RF	2	2	4
	XGB	2	4	6
	GBM	1	4	5
sepsis_cases_2	RF	3	2	5
	XGB	4	5	9
	GBM	4	4	8
sepsis_cases_4	RF	4	4	8
	XGB	4	2	6
	GBM	5	4	9
Total		29	31	

As far as the scenario-based scoring of alternatives is concerned, Table 5, 6 and 7 show the scores achieved by different models and for the different event logs considered in this evaluation. For the BPIC 2011 and Sepsis event logs, the classifier GBM has the highest score, while for the BPIC 2015 event log, RF has the highest score. Depending on the bucketing technique, scores may significantly vary even with the same event log and classifier.

It is interesting to investigate in depth the relation between the Pareto optimality and the scenario-based scoring. To do this, we consider as *high* scenario-based scores the values 5 and 4 (associated with Scenario 1, 2 and 3 in Table 1) and

[2] Note that, to keep the paper concise, drilled down results by bucket are not shown in Table 2, 3, and 4.

low scenario-based scores the values 0 and 1 (associated with Scenario 8, 6, and 7 in Table 1).

Now, considering the results shown in Table 5, 6 and 7, across all event logs, models are scored high with the scenario-based scoring in 47% of the cases (34/72) and low in 32% of the cases (23/72). If we restrict this analysis to the Pareto-optimal alternatives (identified in boldface in Table 2, 3 and 4), then the proportion of models scoring high in the scenario-based scoring increases to 72% (21/29), while the proportion of low-scoring models decreases to 24% (7/29). This highlights that, while on the one hand the Pareto-optimal alternatives are not always *good* if interpreted through the lens of scenario-based scoring, Pareto-optimal alternatives are more likely to have high scenario-based scoring.

Most of the Pareto-optimal models associated with low scenario-based scores have GBM as a classifier (86%, 6/7). This is because alternatives with GBM usually have high IBS, which prevents them to be dominated by other alternatives, even though they score low on other metrics. However, Pareto-optimal alternatives associated with GBM still have high scenario-based scores in 63% of the cases (10/16). Most of the Pareto-optimal models having RF as classifiers have high scenario-based scores (83%, 5/6), and all Pareto-optimal models using XGB have high scenario-based scores (7/7).

Regarding the choice of bucketing method, when considering the scenario-based scores, it can be noted (see Table 6) that for some event logs the prefix-length bucketing appears more likely to lead to stable classifiers, as acknowledged by the higher total score. Moreover, it is possible to identify specific data sets for which one bucketing method is scored consistently equal or higher than the other one across all classifiers and should therefore be preferred. For instance, this is the case of the prefix-length bucketing for BPIC2015_2 or clustering bucketing for sepsis_cases_1.

VI. CONCLUSION

This paper has proposed a framework for the evaluation of the quality of outcome-based predictive process monitoring models. It comprises three novel metrics that evaluate the overall performance across buckets of prefixes (OBP), the stability of performance in respect of the classification cutoff threshold (IBS), and the stability of the performance across different buckets of prefixes (XBS). The aim of the framework is to provide a more nuanced means to evaluate the quality of predictive models that goes beyond the typical focus on overall performance measures derived from a confusion matrix, e.g., overall accuracy or recall. To apply the framework in practical scenarios, we have proposed to compare alternative combinations of classifier and bucketing method using the concept of Pareto-optimality and a scenario-based scoring system. We then have evaluated the proposed framework using several real-life event logs, 3 classifiers (RF, XGB, and GBM) and 2 bucketing methods (prefix length and clustering).

The experimental results have shown that while RF is a classifier that often shows good predictive overall performance, it also often fails to give consistent predictions across different repetitions of the experiments. This is remarkable considering that RF is considered a well-performing and stable classifier in many classification use cases. The classifier GBM is highly likely to give consistent predictions across different repetitions of the experiments. This can help decision-makers to be more confident in the decisions based on predictions obtained using this classifier. Finally, the comparative analysis of Pareto optimality and scenario-based scoring has shown that both methods share commonalities in aggregating the results of IBS, OBP and XBS, enabling either method to be used to assess the quality of predictions in practice.

For future work, two approaches can be taken into consideration. Firstly, the value of the thresholds for dividing high and low values of IBP, OBP and XBS can be adjusted from the average to other statistical measures, such as the lower and upper quartile, as the value of this threshold changes the way in which the quality is assessed even with same IBP, OBP and XBS values. Secondly, from the perspective of predictive monitoring of process outcomes, additional trace bucketing and sequence encoding methods can be compared to extensively investigate how the quality of predictions differs across different configurations. We will also investigate the applicability of the proposed framework in other predictive monitoring use cases, such as next activity or time prediction. Finally, to increase the practical relevance of the proposed framework, we are planning to assess the effectiveness of the Pareto-optimality and the scenario-based scoring, and possibly other novel interpretation schemes of the metrics proposed in this paper, with business process management experts.

REFERENCES

[1] W. Van Der Aalst, A. Adriansyah, A. K. A. D. Medeiros, F. Arcieri, T. Baier, T. Blickle, J. C. Bose, P. Van Den Brand, R. Brandtjen, J. Buijs, and A. Burattin, "Process mining manifesto," in *Proc. Int. Conf. Bus. Process Manag.* Berlin, Germany: Springer, 2011, pp. 169–194.

[2] M. Ceci, P. F. Lanotte, F. Fumarola, D. P. Cavallo, and D. Malerba, "Completion time and next activity prediction of processes using sequential pattern mining," in *Proc. Int. Conf. Discov. Sci.* Cham, Switzerland: Springer, 2014, pp. 49–61.

[3] N. Tax, I. Verenich, M. L. Rosa, and M. Dumas, "Predictive business process monitoring with LSTM neural networks," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2017, pp. 477–492.

[4] I. Teinemaa, M. Dumas, M. L. Rosa, and F. M. Maggi, "Outcome-oriented predictive process monitoring: Review and benchmark," *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 2, pp. 17:1–17:57, 2019.

[5] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, no. 1, pp. 3–24, 2007.

[6] B. A. Tama and M. Comuzzi, "An empirical comparison of classification techniques for next event prediction using business process event logs," *Expert Syst. Appl.*, vol. 129, pp. 233–245, Sep. 2019.

[7] 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, no. 4, pp. 1–34, Aug. 2019.

[8] A. Senderovich, C. D. Francescomarino, C. Ghidini, K. Jorbina, and F. M. Maggi, "Intra and inter-case features in predictive process monitoring: A tale of two dimensions," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2017, pp. 306–323.

[9] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.

[10] H. Jabbar and R. Z. Khan, "Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)," *Comput. Sci. Commun. Instrum. Devices*, pp. 163–172, 2015.

[11] R. Longadge and S. Dongre, "Class imbalance problem in data mining review," 2013, *arXiv:1305.1707*. [Online]. Available: http://arxiv.org/abs/1305.1707

[12] I. Teinemaa, M. Dumas, A. Leontjeva, and F. M. Maggi, "Temporal stability in predictive process monitoring," *Data Mining Knowl. Discovery*, vol. 32, no. 5, pp. 1306–1338, Sep. 2018.

[13] K. Kritikos, B. Pernici, P. Plebani, C. Cappiello, M. Comuzzi, S. Benrernou, I. Brandic, A. Kertész, M. Parkin, and M. Carro, "A survey on service quality description," *ACM Comput. Surveys*, vol. 46, no. 1, pp. 1–58, Oct. 2013.

[14] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data quality assessment," *Commun. ACM*, vol. 45, no. 4, pp. 211–218, 2002.

[15] 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. 2017.

[16] F. M. Maggi, C. D. Francescomarino, M. Dumas, and C. Ghidini, "Predictive monitoring of business processes," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2014, pp. 457–472.

[17] A. Leontjeva, R. Conforti, C. D. Francescomarino, M. Dumas, and F. M. Maggi, "Complex symbolic sequence encodings for predictive monitoring of business processes," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2016, pp. 297–313.

[18] I. Verenich, M. Dumas, M. L. Rosa, F. M. Maggi, and C. D. Francescomarino, "Complex symbolic sequence clustering and multiple classifiers for predictive process monitoring," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2016, pp. 218–229.

[19] V. Pasquadibisceglie, A. Appice, G. Castellano, D. Malerba, and G. Modugno, "ORANGE: Outcome-oriented predictive process monitoring based on image encoding and CNNs," *IEEE Access*, vol. 8, pp. 184073–184086, 2020.

[20] W. Kratsch, J. Manderscheid, M. Röglinger, and J. Seyfried, "Machine learning in business process monitoring: A comparison of deep learning and classical approaches used for outcome prediction," *Bus. Inf. Syst. Eng.*, vol. 63, no. 3, pp. 261–276, 2021.

[21] W. Rizzi, C. D. Francescomarino, and F. M. Maggi, "Explainability in predictive process monitoring: When understanding helps improving," in *Proc. Int. Conf. Bus. Process Manage.* Cham, Switzerland: Springer, 2020, pp. 141–158.

[22] R. Galanti, B. Coma-Puig, M. D. Leoni, J. Carmona, and N. Navarin, "Explainable predictive process monitoring," in *Proc. 2nd Int. Conf. Process Mining (ICPM)*, Oct. 2020, pp. 1–8.

[23] R. Sindhgatta, C. Ouyang, and C. Moreira, "Exploring interpretability for predictive process analytics," in *Proc. Int. Conf. Service-Oriented Comput.* Cham, Switzerland: Springer, 2020, pp. 439–447.

[24] C. Di Francescomarino, M. Dumas, M. Federici, C. Ghidini, F. M. Maggi, and W. Rizzi, "Predictive business process monitoring framework with hyperparameter optimization," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.* Cham, Switzerland: Springer, 2016, pp. 361–376.

[25] C. Di Francescomarino, M. Dumas, M. Federici, C. Ghidini, F. M. Maggi, W. Rizzi, and L. Simonetto, "Genetic algorithms for hyperparameter optimization in predictive business process monitoring," *Inf. Syst.*, vol. 74, pp. 67–83, May 2018.

[26] M. Comuzzi, A. E. Marquez-Chamorro, and M. Resinas, "A hybrid reliability metric for SLA predictive monitoring," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 32–39.

[27] Z. Bosnić and I. Kononenko, "An overview of advances in reliability estimation of individual predictions in machine learning," *Intell. Data Anal.*, vol. 13, no. 2, pp. 385–401, Apr. 2009.

[28] M. Kukar and I. Kononenko, "Reliable classifications with machine learning," in *Proc. Eur. Conf. Mach. Learn.* Springer, 2002, pp. 219–231.

[29] D. Gunning, M. Stefic, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang, "XAI—Explainable artificial intelligence," *Sci. Robot.*, vol. 4, no. 37, pp. 1–5, 2019.

[30] B. Kailkhura, B. Gallagher, S. Kim, A. Hiszpanski, and T. Y.-J. Han, "Reliable and explainable machine-learning methods for accelerated material discovery," *NPJ Comput. Mater.*, vol. 5, no. 1, pp. 1–9, Dec. 2019.

[31] G. Liang, X. Zhu, and C. Zhang, "An empirical study of bagging predictors for different learning algorithms," in *Proc. 25th AAAI Conf. Artif. Intell.*, 2011, pp. 1802–1803.

[32] A. Kaur and K. Kaur, "An empirical study of robustness and stability of machine learning classifiers in software defect prediction," in *Proc. Adv. Intell. Inform.* Cham, Switzerland: Springer, 2015, pp. 383–397.

[33] R. Tissier, J. Houwing-Duistermaat, and M. Rodríguez-Girondo, "Improving stability of prediction models based on correlated omics data by using network approaches," *PLoS ONE*, vol. 13, no. 2, Feb. 2018, Art. no. e0192853.

[34] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[35] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[36] M. Last, O. Maimon, and E. Minkov, "Improving stability of decision trees," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 16, no. 2, pp. 145–159, Mar. 2002.

[37] M. Olivos-Trujillo, H. A. Gajardo, S. Salvo, A. Gonzalez, and C. Muñoz, "Assessing the stability of parameters estimation and prediction accuracy in regression methods for estimating seed oil content in brassica napus L. Using NIR spectroscopy," in *Proc. Conf. Electr., Electron. Eng., Inf. Commun. Technol. (CHILECON)*, Oct. 2015, pp. 25–30.

[38] I. Teinemaa, M. Dumas, F. M. Maggi, and C. D. Francescomarino, "Predictive business process monitoring with structured and unstructured data," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2016, pp. 401–417.

[39] G. T. Lakshmanan, S. Duan, P. T. Keyser, F. Curbera, and R. Khalaf, "Predictive analytics for semi-structured case oriented business processes," in *Proc. Int. Conf. Bus. Process Manag.* Cham, Switzerland: Springer, 2010, pp. 640–651.

[40] E. Western, *Statistical Quality Control Handbook*, 1st ed. Indianapolis, IN, USA: Western Electric, 1956.

[41] A. E. Márquez-Chamorro, M. Resinas, A. Ruiz-Cortés, and M. Toro, "Run-time prediction of business process indicators using evolutionary decision rules," *Expert Syst. Appl.*, vol. 87, pp. 1–14, Nov. 2017.

[42] 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.

[43] V. Pareto, *Manuale di Economia Politica*. Milan, Italy: Società Editrice Libraria, 1906.

[44] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011, pp. 2546–2554.

[45] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3133–3181, 2014.

[46] R. S. Olson, W. La Cava, Z. Mustahsan, A. Varik, and J. H. Moore, "Data-driven advice for applying machine learning to bioinformatics problems," 2017, *arXiv:1708.05070*. [Online]. Available: http://arxiv.org/abs/1708.05070

JONGCHAN KIM is currently pursuing the Ph.D. degree with the Department of Industrial Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea. His research interests include artificial intelligence for process mining, machine learning applications, data science, and quality of predictions in predictive monitoring.

MARCO COMUZZI received the Ph.D. degree in information technology from the Politecnico di Milano, in 2007. He is currently an Associate Professor at the Department of Industrial Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea. His research interests include business process management, data science, and blockchain.