A Hybrid Reliability Metric for SLA Predictive Monitoring*

Marco Comuzzi
Ulsan National Institute of Science and Technology
Ulsan, Republic of Korea
mcomuzzi@unist.ac.kr

Alfonso E. Marquez-Chamorro
Universidad de Sevilla
Sevilla, Spain
amarquez6@us.es

Manuel Resinas
Universidad de Sevilla
Sevilla, Spain
resinas@us.es

ABSTRACT

Modern SLA management includes SLA prediction based on data collected during service operations. Besides overall accuracy of a prediction model, decision makers should be able to measure the reliability of individual predictions before taking important decisions, such as whether to renegotiate an SLA. Measures of reliability of individual predictions provided by machine learning techniques tend to depend strictly on the technique chosen and to neglect the features of the system generating the data used to learn a model, i.e., the service provisioning landscape in this case. In this paper, we consider business process-aware service provisioning and we define a hybrid measure of reliability of an individual SLA prediction for classification models, which accounts for both the reliability of the chosen prediction technique, if available, and features capturing the variability of the service provisioning scenario. The metric is evaluated empirically using SLAs and process event logs of a real world case.

KEYWORDS

SLA monitoring,

business process,

predictive monitoring,

reliability.

1 INTRODUCTION

The ubiquitous support of information systems and the emerging availability of Internet-of-Things(IoT) technology enable the collection of large amount of data during service operations with the objective of improving service analysis, design and enhancement. One typical application of data-driven service analysis is Service Level Agreement (SLA) monitoring, whereby data captured during service operations are analysed to assess to what extent service objectives and guarantees negotiated by providers and requesters have been achieved. Because of the growing importance of AI and machine learning techniques in data analysis, modern SLA monitoring includes SLA prediction, that is, using collected data to infer, with certain levels of accuracy, whether and to what extent SLAs are going to be achieved or violated [11]. In this paper, we consider business process-aware service provisioning, i.e., scenarios in which services are implemented privately by the provider through a business process, enacted using a Process-Aware Information System (PAIS).

A prominent approach in predictive monitoring of service-based systems or business processes, involves the adaptation of existing machine learning techniques to solve new predictive monitoring problems with higher accuracy [17]. However, the accuracy of a predictive model is calculated by aggregating prediction results across a test set of previous cases and, as such, it does not give a precise indication of how much decision makers can trust an *individual* prediction based on new data or, in other words, about the likelihood that a new individual prediction is eventually correct [16]. From a practical standpoint, however, having a means to gauge the *reliability* of individual SLA predictions is sometimes even more important to decision makers than the overall accuracy of predictions. For instance, when deciding whether to renegotiate an agreement with a client to extend the service completion due date, a service provider needs to know to what extent it can rely on, or trust, a prediction made for this particular client to be eventually correct.

Machine learning techniques often define specific metrics for the reliability of an individual prediction, such as the classification probability in decision trees or other measures based on sensitivity of predictions [4]. However, these measures are based only on the training data, may depend strictly on the chosen machine learning technique, and most importantly do not take into account domain-specific features of the system generating data used to learn a model [2].

The aim of this paper is to define a *hybrid* measure of SLA prediction reliability in classification models that combines reliability metric(s) available for the chosen prediction technique, such as classification probability, with domain-specific features of the service provisioning scenario that may affect the reliability of an SLA prediction. Given the variety of

*This work has partially received fundings from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 645751 (RISE_BPM), grants TIN2015-70560-R (MINECO/FEDER, UE) and P12-TIC-1867 (Andalusian R&D&I program), and NRF Korea Project Number 2017076589.

possible service provisioning scenarios, we restrict our analysis to service-based systems that are process-aware, i.e., in which a service is implemented as the public view of a private business process executed privately by a service provider [9]. In this scenario, our hypothesis is that the reliability of an individual prediction partially depends on the *variability* of the scenario in which a prediction is made. A prediction, for instance, is likely to be more reliable when the service provisioning is almost complete or, more generally, when the choices available to service requesters and providers to complete the provisioning of a service are limited. In other cases, variability may also be associated with the time elapsed to serve a service request, e.g., the longer a service request has been served, the more likely it is closer to its termination and, therefore, the fewer the possible choices available to providers and requesters.

Therefore, our proposal combines reliability metrics derived from the chosen prediction technique with other factors accounting for the variability of a business process, such as measures of time elapsed/remaining in the execution of a process instance or number of different alternatives available to conclude the execution of a process instance. Part of this knowledge about variability of a service provisioning scenario may be already embedded by the predictive model in the learning phase, particularly in the case of complex non linear models, such as neural networks. Similarly to the problem of explaining a prediction made by a machine learning model, however, it is often practically impossible to disentangle this knowledge from the internal functioning of a training algorithm in order to obtain a measure of reliability for an individual prediction [1].

The proposed metrics are evaluated using real world business process event logs and state of the art predictive monitoring techniques. The experimental results show that, in many cases, a prediction reliability measure based only on the chosen predictive monitoring technique is not the best one. That is, a hybrid reliability measure accounting for the variability of the service provisioning scenario is often the one that minimises the reliability error. As such, the contributions of this paper are (i) to put forward the issue of prediction reliability in the context of process-aware service-based systems, (ii) to provide an initial definition of prediction reliability, which includes also features related with the variability of the scenario in which a prediction is made and (iii) to show empirically, using two real event logs, how the proposed reliability metric is often better than metrics typical of the chosen predictive technique as an estimator of the likelihood of a prediction to be correct.

This paper is organised as follows. Section 2 defines the service provisioning scenario considered by this paper. Section 3 introduces a model to define the hybrid SLA monitoring reliability metric. An experimental evaluation is discussed in Section 4 and related work is presented in Section 5. Conclusions are eventually drawn in Section 6.

2 PROCESS-AWARE SERVICE PROVISIONING SCENARIO

We consider a scenario (see Fig. 1) in which a service is implemented, on the provider's side, as a business process. That is, a service can be seen as a *public view* of a private business process run by a process provider [9]. There is a service level agreement (SLA) in place between the service provider and the requester to specify the terms of service provisioning. These terms usually include service level objectives (SLOs) that specify the guarantees made by the provider regarding some service level metrics, and the penalties and rewards that shall be applied if the service provider does not meet or exceeds the SLOs, respectively [6, 7].

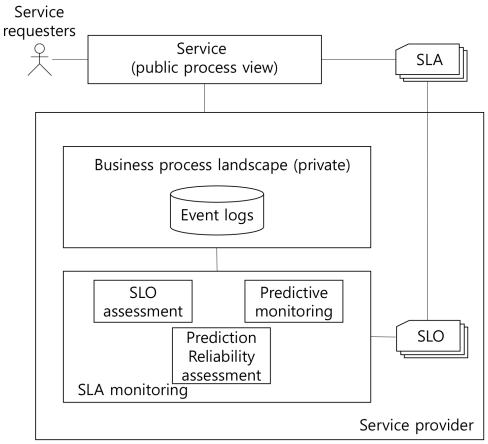


Figure 1: SLA monitoring scenario.

For instance, let us consider an IT service provider providing incident resolution services to several public administration organisations. SLAs between the provider and individual clients (service requesters) may include an SLO that states that, in any 6-month time window, incidents must be solved in at most 2 days in 90% of the cases. Furthermore, it may also impose a penalty of a 10% discount of the billing for that period if the SLO is not met. In this context, SLO monitoring is associated with monitoring the value of specific PPIs for the process cases that are targeted by an SLA, e.g., monitoring execution time for the incident resolution cases that run in any given 6-months window.

We assume that SLO monitoring is comprised of two types, namely *SLO assessment* and *predictive monitoring*. SLO assessment focuses on verifying the satisfaction of SLOs ex-post, i.e., after the process cases targeted by an SLA have completed their execution. In a process-aware scenario, this type of monitoring relies on the availability of event logs, in which execution data of process cases, such as names of activities that have been executed, timestamps of activity execution, and value of SLOs are recorded. Predictive monitoring aims at predicting the values of SLOs for cases targeted by an SLA using some kind of predictive monitoring technique. It also relies on event logs, which are used in this case to

train a model that is then used for making predictions [14]. Information from SLO assessment and predictive monitoring are combined to address a number of provider's use cases, such as anticipating violations of SLAs or determining the likelihood of incurring into penalties or rewards specified in an SLA.

Based on this definition, SLO monitoring involves a deterministic component, i.e., SLO assessment based on actual data generated by completed process cases, and a predictive component, i.e., from predictive monitoring. Any decisions taken about a specific SLA, e.g., prioritising an incident or allocating more resources to it, has important implications for the provider, in terms of cost, time and/or resource allocation. Therefore, any generated prediction should be accompanied by a *reliability* value, which captures the extent to which service providers can rely on (or *trust*) this individual prediction in their decisions.

SLOs are normally defined by aggregating conditions on individual process metrics at the level of cases, e.g., case execution should be less than 2 days, across several cases targeted within the time scope of an SLA, e.g., all cases in a 6-months time window. Consequently, one can be interested in two kinds of predictions. Predictions made on individual process cases to ensure that the case is going to finish successfully, e.g., in less than 2 days in the example considered thus far, and aggregated predictions made across a set of process cases, e.g., whether the case execution of 90% of cases in the 6-months time window is going to finish on time. In this paper we focus on defining a metric of reliability for predictions made on individual process cases, leaving the definition of the reliability of aggregated measures for future work.

Based on the scenario presented in this section, the objective of this paper is to define an *hybrid* reliability metric for prediction about SLO values in individual process cases. Specifically, we define a generic metric, which can be customised based on different features of the service provisioning scenario and process landscape.

3 A HYBRID METRIC OF PREDICTION RELIABILITY

Let $\mathcal{S}$, $\mathcal{P}$, and $\mathcal{T}$ represent the universe of services, processes, and the time domain, respectively. A service $S \in \mathcal{S}$ is provided by a service provider to one or more service requesters. A service is implemented internally by a provider through a business process $P \in \mathcal{P}$.

Based on the scenario presented in the previous section, we assume that a service provider associates a service S with a set of M service level objectives SLO_m, that is, $S = \{SLO_m\}_{m=1,\ldots,M}$. An SLO SLO_m assumes values within a domain D_m. This can be numerical, i.e., $D_m \subseteq \mathbb{R}$ or categorical, in which case D_m is constituted by a (possibly infinite) set of values $v_{m,k}$. Usually SLOs assume boolean values, such as whether the incident has been resolved in time, or if the root cause of the incident has been informed. However, other domains are also possible. For instance, SLOs related with time assume values in a continuous domain, or they can assume general categorical values if clients are allowed to give a numerical score, e.g., from 1 to 5, when rating an implemented solution.

Let $\mathcal{I}$ be the universe of cases (or instances) of a process P. We define the value v and the predicted value $\hat{v}$ of an SLO for a case as follows:

- $v : \mathcal{I} \times S \times \mathcal{T} \rightarrow D_m \cup \{\perp\}$, written $v_j^t SLO_m$, mapping a case $j \in \mathcal{I}$ and an SLO $SLO_m \in S$ onto a value in the domain D_m at a given time instant $t \in \mathcal{T}$. Note that the undefined value $\perp$ is used when the value of SLO_m cannot be calculated at time t. For instance, the execution time of a case or the score assigned by a client to an incident solution are known only after a case has completed its execution;
- $\hat{v} : \mathcal{I} \times S \times \mathcal{T} \rightarrow D_m \cup \{\perp\}$, written $\hat{v}_j^t SLO_m$, mapping a case $j \in \mathcal{I}$ and an SLO $SLO_m \in S$ onto a value a *predicted* value in the domain D_m at a given time instant $t \in \mathcal{T}$. A predicted value is obtained using some prediction technique using data generated during process execution, i.e., event logs. The undefined value $\perp$ is used when a predicted value cannot be calculated based on available data.

The objective of this paper is to define a hybrid metric to measure the reliability of predicted SLO values $\hat{v}_j^t SLO_m$. The problem of defining a hybrid reliability metric for SLA prediction is the problem of defining the function $r : \mathcal{I} \times S \times \mathcal{T} \rightarrow 0, 1$, written as $r_j^t SLO_m$, to indicate the reliability of an individual predicted value of a service level objective SLO_m for the j-th case of process P at time t. The following principles drive the design of this metric:

1) It should indicate the likelihood that an individual prediction is eventually correct: the main purpose of a prediction reliability metric is to assess to what extent a decision maker can trust an individual prediction, as opposed to model accuracy, which accounts for prediction performance across a set of test cases. As such, the proposed reliability metric should be highly and positively correlated with the fact that a prediction $\hat{v}_j^t SLO_m$ at time t is eventually going to be correct, that is at some point in time $t' > t$ (once a case is completed at the latest), $\hat{v}_j^t SLO_m = v_j^{t'} SLO_m$;

2) It should include existing predictive monitoring reliability metrics (if available): the metric should consider, if available, the value of reliability of the prediction techniques used for process predictive monitoring. For instance, if a decision tree is used to make predictions based on event logs, then the probability associated with the class chosen for a prediction can be seen a measure of reliability, i.e., an indication of the extent to which the decision maker can trust the provided prediction. Other machine learning models may provide other reliability measures, or this can be defined ad hoc, for instance by considering the sensitivity of predictions [3];

3) It should consider process case variability: as we outlined before, we argue that reliability of a prediction should be related with the variability of the scenario in which a prediction is made. In our case, since we consider predictions

made at the level of process cases, this means that reliability should account for the variability of a process case. Obviously, this variability concerns the future execution of the case, i.e., how the case will complete. Therefore, the metric should consider the variability associated with the execution of a case from the current state of advancement at time t until its termination. Specifically, the higher this variability, e.g., higher number of predicted activities to be executed, higher expected completion time, or higher number of choices in an a process case to make before termination, the lower the reliability of a prediction made at time t;

4) It should hold typical properties of a metric: in order to be a *metric*, the proposed reliability metric can only assume values between 0 and 1, with 1 signifying that there is a 100% likelihood that the predicted value of an SLO is eventually correct and it should assume the value 1, i.e., 100% reliability, when the actual value of an SLO $v_j^t SLO_m$ becomes available.

The last principle indicates a set of properties that are guaranteed by design in the proposed definition of an hybrid SLA prediction reliability metric given below. Principles 1 to 3 drive the design of the proposed reliability metric and, in particular, principle 1 also determines the way in which we evaluate the proposed metric in Section 4. Based on them, we define $rel_j^t SLO_m$ as follows:

$$rel_j^t SLO_m = w_1 \cdot adv_j^t SLO_m + w_2 \cdot time_j^t SLO_m + w_3 \cdot pred_j^t SLO_m. \quad (1)$$

That is, with $w = 1$, the proposed reliability metric is comprised of the weighted sum of the following 3 terms:

- $adv_j^t SLO_m$, which consider process case variability at the level of execution advancement, i.e., focusing on execution of activities;
- $time_j^t SLO_m$, which considers time-related process case variability, i.e., focusing on time-related information regarding the current case;
- $pred_j^t SLO_m$, which refers to a value of reliability of a prediction defined by the prediction technique in use, e.g., prediction probability in decision tree-based classification. Note that this value may not be available when the prediction technique in use does not provide any kind of prediction reliability.

In order to specify each term in $rel_j^t SLO_m$, let us first introduce some required notation about cases and event logs. Let $\mathcal{E}$ be the universe of all events, an event log L is a K-sized set of completed process instances or cases, $L = \{\sigma_i | i = 1, \ldots, K\}$, with $\sigma_i = e_i^1, \ldots, e_i^{n_i} \in \mathcal{E}^*$ being the ith case with length n_i. Let σ_j be the running case, with $\sigma_j = \{e_j^1, \ldots, e_j^{l_j}\}$, on which a prediction is being made for which we want to compute the reliability.

Regarding the first term $adv_j^t SLO_m$, let l_j be the number of activities executed thus far in case σ_j. We assume that an estimate of the remaining number of activities to be executed in the σ_j $\hat{k}_j$ is available. Then, $adv_t^j = fl_j, \hat{k}_j$, where f is a monotonic increasing *activation function* valued between 0 and 1 and for which $\lim_{\hat{k}_j \to 0} fl_j, \hat{k}_j = 1$, e.g., $fl_j, \hat{k}_j = \frac{l_j}{l_j + \hat{k}_l}$.

In some situations (see Fig. 2a), e.g., when the estimate $\hat{k}_j$ is obtained through some predictive monitoring technique [18] or by matching the current execution trace with previous similar cases, calculating $\hat{k}_j$ does not imply the existence of a process model for P, since both l_j and $\hat{k}_j$ can be calculated from an event log. Not relying on the existence of a process model best suites scenarios with high case-level variability, in which each case may be executed in a different way. A process model, in this scenario, is likely to be very complex and practically unusable (i.e., a *spaghetti* model [19]), but an event log can be used, for instance, to match the current case execution trace with previous cases to identify previous similar cases and use them to estimate the number of remaining activities in a case with a certain level of confidence.

An alternative way of calculating $\hat{k}_j$ (see Fig. 2b), which requires a process model, but does not rely on techniques for predicting the remaining number of activities in a case, is based on the notion of *paths* to terminate a process case [5]. Given l_j, let Z be the number of possible paths $z = 1, \ldots, Z_j$ to complete the execution of the σ_j, $p_j z$ the probability that path $path_{z_j}$ is executed and $|path_{z_j}|$ the length, i.e., number of activities, in path $path_{z_j}$, then:

$$adv_j^t SLO_m = \sum_{z_j=1}^{Z_j} p_{z_j} \cdot fl_j, |path_{z_j}|,$$

where f is a monotonic increasing *activation function* with values between 0 and 1 and with $\lim_{|path_{z_j}| \to 0} fl_j, |path_{z_j}| = 1$ e.g., $fl_j, p_{z_j}, |path_{z_j}| = \frac{l_j}{l_j + |path_{z_j}|}$.

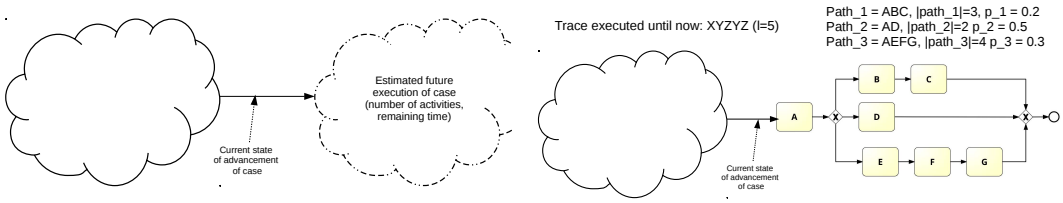
The calculation of the second term $time_j^t SLO_m$ is conducted in a similar way. Let t_j^{ex} be the time elapsed from the start of case σ_j and $\hat{t}_j$ an estimate of the remaining time required to complete case σ_j. This estimate can be derived, for instance, using predictive monitoring techniques. In this case, $time_t^j = ft_j, \hat{t}_j$, where f is a monotonic increasing *activation function* with values between 0 and 1, e.g., $ft_j, \hat{t}_j = \frac{t_j}{t_j + \hat{t}_l}$.

Alternatively, by relying on the notion of possible paths to terminate the execution of a case , let $\hat{t}_{z_j}$ be an estimation of the time to complete the execution of a path $path_{z_j}$ to terminate a case j, then:

$$time_j^t SLO_m = \sum_{z_j=1}^{Z_j} p_{z_j} \cdot ft_j^{ex}, \hat{t}_{z_j},$$

where $ft_j^{ex}, \hat{t}_{z_j}$ is a monotonic increasing function, e.g. $ft_j^{ex}, \hat{t}_{z_j} = \frac{t_j^{ex}}{t_j^{ex} + \hat{t}_{z_j}}$.

Note that, unlike with $adv_j^t SLO_m$, in this case even when considering paths, an estimation of the time remaining to complete each path $\hat{t}_{z_j}$ is required, which can be calculated in a number of different ways, e.g., by aggregating historical average execution time of activities in a path $path_{z_j}$ or using predictive monitoring techniques.



(a) Process model not available

(b) Using *paths* to complete case execution

Figure 2: Obtaining an estimate of remaining number of activities or time to complete a case.

4 EVALUATION

This section presents the experimentation results obtained to asses the validity of our proposal. Our evaluation focuses on the following two research questions:

(1) Is the hybrid measure of reliability proposed in this paper more accurate than using the classification probability alone in estimating the likelihood that an individual prediction is eventually correct?
(2) Which combination of weights w assigned to different terms of the proposed reliability metric achieves the best results in estimating the likelihood that an individual prediction is eventually correct?

4.1 Datasets

Two real-life event logs have been considered in our experiments: a private log from an IT Department of a Public Administration (PA) and publicly available event log of the BPI Challenge 2013 (BPIC13).

PA dataset represents the incident management log of the IT Department of a Public Administration in Spain. In this scenario, a service level agreement (SLA) defines several SLOs and the penalties that are imposed in case the SLOs are not met. For the experimentation, we consider the SLO K20, which indicates an abuse of the stopping time (idle time > 0). Idle time is used by the provider to stop the clock when they cannot advance in the incident resolution because of factors beyond their control such as waiting for the user's response. This event log consists of 174.989 events, each of them with 15 attributes extracted from an incident management system.

BPIC13 dataset was extracted from Volvo IT incident management log[1]. This log contains all the information of the management of incidents registered at the Volvo IT department. A solution should be established for each incident in order to restore the service with minimum disruption to the business. The incident is closed after providing a solution to the problem and verifying that the service is restored. In this context, the SLO to be predicted is whether the remaining time to solve an incident is less than a predetermined threshold (12 days). This event log consists of 65.533 events,7554 process instances,and a total number of 12 attributes.

[1] https://www.win.tue.nl/bpi/doku.php?id=2013:challenge/

4.2 Experimental procedure

The experimental procedure can be decomposed in several steps. First, the log is encoded using a sliding window of 2 events, since empirical evaluation has showed this is a good window size [15]. Thus, each feature vector is composed by the different attributes of the 2 events of the event window, while the last position corresponds to the class, which indicates a value of the SLO to be predicted. In this case, all predicted SLOS assume boolean values as usually happens with SLOs. The other attributes can be nominal or a real number. In addition, the attributes of the dataset are extended with two additional attributes: the number of events that have occurred in the case and the time elapsed since the beginning of the case. More detailed information of the encoding is provided in [15].

In the next step, we separate the dataset between the training and the test datasets. We use 10-fold validation over the cases of the dataset, that is, the cases of the dataset are divided into 10 groups: 9 of them are used for training and the remaining one is used for testing. This process is repeated 10 times so that each group is used for testing once. The results are the average obtained across the 10 repetitions.

Three predictive models are built using the training dataset. The first predictive model ($\hat{v}_j^t$) corresponds with the SLOs that we are considering, i.e., K20 for the PA dataset and remaining time for the BPIC13 dataset. The predictive model is built using a random forest classifier because they have shown good results for the prediction of this kind of metrics [13, 18]. The other two models are built also using random forests and predict the total number of events of a case σ_j ($\hat{n}_j$) and the total time of a case σ_j ($\hat{t}_j$). These last two models are used later on to compute the hybrid reliability metric of a prediction. For all three predictive models, we have used the random forest implementation of scikit-learn[2] with the default parameters.

Finally, we obtain a prediction and a reliability value for each of the events of the test dataset. The prediction is obtained by directly applying the predictive model built in the previous step. The reliability value is also computed for each event according to the formula described in Section 3: $rel_j^t SLO_m = w_1 \cdot adv_j^t SLO_m + w_2 \cdot time_j^t SLO_m + w_3 \cdot pred_j^t SLO_m$, where j represents a case of a process at time t. In our case, SLO_m corresponds to K20 for the PA dataset

[2] http://scikit-learn.org/

and remaining time for the BPIC13 dataset. To compute the term $adv_j^t SLO_m$, we use the activation function described in the previous section using the predictive model $\hat{n}_j^t$ obtained in the previous step. Thus, adv_j^t is equal to the number of activities l_j^t executed thus far in a case j divided by the cited prediction: $adv_j^t SLO_m = \frac{l_j^t}{\hat{n}_j^t}$. The term $time_j^t SLO_m$ is computed in a similar way. For each event of the case, we estimate the total time of the case using the predictive model $\hat{t}_j$ and we compute $time_j^t SLO_m = \frac{t_j^{ex}}{\hat{t}_j}$. Finally, the term $pred_j^t SLO_m$ is obtained from the classification probability provided by the random forest model $\hat{v}_j^t$.

The whole experimentation on the public dataset is available in a Jupyter notebook at https://github.com/isa-group/predictive-monitoring-reliability so that it can be easily replicated.

4.3 Evaluating the reliability metric

Once we have the prediction and reliability values, it is necessary to assess the validity of the proposed reliability metric. As it often happens in the monitoring of SLAs, SLOs and their predictions in our case assume boolean values. This means that ideally 50% of the predictions that have a reliability value of 0.5 should be correct. Similarly, ideally 80% of the predictions with reliability equal to 80% should be correctly predicted.

Therefore, to assess the validity of the reliability metric, we have divided the predictions according to their calculated reliability value in intervals n_i of size 0.1. For instance, the reliability interval $n_{0.4} = 0.4, 0.5$ contains all predictions for which $0.4 \leq rel_j^t SLO_m < 0.5$. Then, for each interval n_i, with $i = \{0, \ldots, 0.9\}$, we compute the accuracy of the predictions of the interval as $Acc_i = \frac{TP+TN}{TP+TN+FP+FN}$, where TP, TN, FP and FN are the number of true positives, true negatives, false positives and false negatives, respectively.

We use these values of Acc_i to compute three different metrics. First, we determine the deviation of this value Acc_i from the center of the interval, i.e., $|Acc_i - i + i + 0.12|$. If the proposed reliability metric is valid, this means, for instance, to assume that the interval $0.4, 0.5$ should contain approximately 45% of correct predictions, 55% for the next interval and so on. We use these deviations to compute the average error (avg_err), which is the average of the deviation of each Acc_i from the center of its interval. Similarly, the weighted average error ($wavg\_err$) is computed as the weighted average of the deviations, where the weights are defined based on how many predictions fall in each interval. By doing so, $wavg\_err$ gives more importance to the intervals with more predictions. The third metric is the Pearson correlation coefficient (and the significance level of the correlation) between Acc_i and the center of the intervals. This metric is inspired by [4] and it reflects the fact that these two values should be positively correlated for an accurate estimation of the reliability, i.e., the accuracy of interval $0.4, 0.5$ should be lower than the accuracy of interval $0.7, 0, 8$.

4.4 Results

We have computed the values of reliability $rel_j^t SLO_m$ for each event j in the event log and each possible combination of the weights w_1, w_2 and w_3, sampling weights values at intervals of 0.1. An extract of the results showing the combinations of weights values with lowest average error (10 best combinations) is shown in Tables 1 and 2.

In both tables, the three first columns represent the weights for each parameter of the reliability metric (w_{adv}, w_{time} and w_{pred}). The fourth and fifth columns indicate the non-weighted and weighted average error (avg_err and $wavg\_err$) previously described, respectively. Finally, the sixth column shows the Pearson correlation coefficient between Acc_i and the center of intervals, whereas the seventh column depicts the significance level ($\alpha \leqslant 0.05$) of the correlation. As we can see, the positive correlation has statistical significance for the best combinations of weights cited above.

Both tables show that the average error is lower in most cases when considering the terms adv and $time$ in the reliability definition as opposed to using only the classification probability of the random forest model (that is, the weight combination with w_{pred}=1).

We can notice in Table 1 that better results of avg_err are achieved for w_{adv}=0.1, w_{time}=0.1 and w_{pred}=0.8 ($wavg\_err$=0.0264), while for w_{pred} = 1 the average error is 0.1032. An improvement of 8 percentage points is achieved in this case by including the variability terms adv and $time$. We can also highlight the second best result w_{adv}=0.3 and w_{pred}=0.7 ($wavg\_err$=0.0298), which is also remarkable. In this case we appreciate a significant difference between the non-weighted and weighted average error of almost 3 points. Observing Table 2, we can appreciate that for w_{pred}=1 the average error is 0.0784, while the best outcomes are obtained for w_{adv}=0.2 and w_{pred}=0.8 ($wavg\_err$=0.0200) or w_{time}=0.1 and w_{pred}=0.9 (avg_err=0.0247). A significant decrease of the avg_err (5.8%) is identified. We can also appreciate in both cases how, on average, the weighted average error shows better results than the non-weighted one.

Similar insights can also be appreciated in Table 3, which shows the number of predictions belonging to each interval (n_0 - $n_{0.9}$), and the accuracy values for each interval (Acc_0 - $Acc_{0.9}$) for two different combinations of weights in the PA dataset: the case of w_{pred} = 1 and the best combination of weights. The results show how the deviation of values Acc_i from the center of the different intervals are lower (i.e., higher accuracy) for the best combination of weights. Note that, in Table 3, the reliability metric starts from $Acc_{0.5}$ for the first case (w_{pred}=1) and $Acc_{0.4}$ for the second case (best weights combination). This is due to the fact that, in a binary classification, the classification probability cannot be higher than 0.5, because only two classes exist. Therefore, for w_{pred}=1, reliability (which coincides with the classification probability) can never be lower than 0.5. For the second weight combination, while this limitation does not exist, the value of weight w_{pred} (0.8) causes that the reliability values are always above 0.4.

w_{adv}	w_{time}	w_{pred}	avg_err	$wavg\_err$	P_corr	p_value
0	0	1	0.1056	0.1032	0.9789	0.0036
0	0.1	0.9	0.0346	0.0348	0.9883	0.0002
0	0.2	0.8	0.0526	0.0544	0.9820	0.0004
0.1	0	0.9	0.0607	0.0551	0.9546	0.0030
0.1	0.1	0.8	**0.0254**	**0.0264**	**0.9913**	0.0001
0.2	0	0.8	0.0479	0.0399	0.9553	0.0029
0.2	0.1	0.7	0.0561	0.0355	0.9701	0.0002
0.3	0	0.7	0.0583	0.0298	0.9568	0.0007
0.3	0.1	0.6	0.0685	0.0550	0.9693	0.0003
0.4	0	0.6	0.0784	0.0505	0.9391	0.0016

Table 1: Experimental results for PA dataset.

w_{adv}	w_{time}	w_{pred}	avg_err	$wavg\_err$	P_corr	p_value
0	0	1	0.0897	0.0784	0.9724	0.0059
0	0.1	0.9	0.0297	0.0247	0.9853	0.0005
0	0.2	0.8	0.0429	0.0435	**0.9925**	0.0001
0.1	0	0.9	0.0344	0.0296	0.9824	0.0008
0.1	0.1	0.8	0.0293	0.0276	0.9897	0.0002
0.2	0	0.8	**0.0243**	**0.0200**	0.9857	0.0005
0.2	0.1	0.7	0.0613	0.0497	0.9888	0.0003
0.3	0	0.7	0.0466	0.0278	0.9763	0.0002
0.4	0	0.6	0.0629	0.0458	0.9890	0.0003
0.5	0	0.5	0.1033	0.0734	0.9633	0.0006

Table 2: Experimental results for BPIC13 dataset.

w_{adv}	w_{time}	w_{pred}	n_0	$n_{0.1}$	$n_{0.2}$	$n_{0.3}$	$n_{0.4}$	$n_{0.5}$	$n_{0.6}$	$n_{0.7}$	$n_{0.8}$	$n_{0.9}$
0	0	1	0	0	0	0	0	491	974	946	983	3158
0.2	0	0.8	0	0	0	0	291	965	1201	1138	1333	1626
w_{adv}	w_{time}	w_{pred}	Acc_0	$Acc_{0.1}$	$Acc_{0.2}$	$Acc_{0.3}$	$Acc_{0.4}$	$Acc_{0.5}$	$Acc_{0.6}$	$Acc_{0.7}$	$Acc_{0.8}$	$Acc_{0.9}$
0	0	1	0	0	0	0	0	0.487	0.562	0.629	0.723	0.9
0.2	0	0.8	0	0	0	0	0.489	0.549	0.622	0.717	0.858	0.957

Table 3: Number of predictions and Acc_i value for two examples of the PA dataset.

In conclusion, this experimentation has tested that the inclusion of features that capture the variability (advancement- and time-related) of the running cases of business processes in a reliability measure, is more accurate than simple probability metrics of machine learning techniques, i.e, predicted class probability of random forest in our case, in estimating the likelihood that a prediction is eventually correct. Specifically, for all possible combinations of weights in the proposed reliability metric, our proposal outperforms clearly the results for w_{pred}=1 in 14 cases for the PA dataset, and in 12 cases for the BPIC13 dataset.

As for the second research question, concerning the best combination of weights, in the best cases of both datasets, the values of w_{adv} range from 0 to 0.5 and values of w_{time} range from 0 to 0.2. Therefore, in both cases w_{time} seems to be less relevant than the others, and the best results are obtained with a combination of two or three weights in which w_{pred} takes the biggest share.

5 RELATED WORK

Predictive monitoring of service-based and process-aware systems initially considered statistical methods [20] or system modelling techniques [11, 12]. More recently, machine learning techniques have been applied extensively for predicting service/process time-related indicators and for providing recommendation during service/process execution [14, 15].

However, while learning models, i.e., for classification or regression, are usually evaluated in terms of accuracy of their prediction, the problem of estimating the reliability of an individual prediction made by a learning model, which is the goal of this paper, has received relatively less attention in the literature. It is recognised, however, that the latter is a problem at least as important for decision makers in the real world as the accuracy of a learning model [2]. Reliability and local accuracy of predictions is also an important issue when selecting a classifier among a set of competing ones [21].

Prediction reliability estimates of ensembles of neural networks [16], decision trees and random forests [8, 14] have been considered in the context of predicting violations of

process performance metrics and constraints. In these works, however, only the reliability provided by the adopted machine learning technique is considered. In our work, we integrate into this reliability other factors capturing the variability associated with process instance execution.

Some prediction models provide output measures that can be directly used as reliability measures, e.g., the classification prediction in decision trees or support vector machines. In the general case, reliability of individual predictions can be calculated by means of (i) sensitivity analysis [4], by (ii) perturbing the training set, or by (iii) transduction [10]. With sensitivity analysis, a prediction is considered more reliable if the variability of predictions made for similar input data is limited. By perturbing the training data, an individual prediction is reliable if it does not change with adding or removing learning examples. Finally, with transduction reliability is assessed by comparing predictions using models trained with and without a particular new example [3]. However, most of these approaches have been designed for regression instead of classification. Furthermore, they do not consider domain-specific factors accounting for the variability of the system at hand, which, as our results have shown, are useful to improve the performance of reliability metrics. This may be due to the fact that machine learning research tends to focus on model-specific improvements, that are then validated across a set of different datasets. As such, model designers often do not have access to the system generating data. This is not the case of the scenario considered in this paper, in which we can assume some knowledge about the system generating the data, i.e., the business process for which SLOs are predicted.

6 CONCLUSIONS

This paper has presented a novel definition of reliability of SLA predictions, which includes terms related with the variability of the service provisioning scenario in which a prediction is made. The experimental evaluation has confirmed that the proposed reliability metric gives a better estimate of whether an individual prediction is correct when compared to typical reliability indicators of machine learning algorithms, such as class prediction probabilities of decision trees.

The work presented can be extended in several ways. First, we aim at refining the proposed reliability metric, by considering more complex ways of capturing variability of process cases and their impact on the correctness of predictions. While in this paper we only focused on classification problems in individual process cases, possible extensions concerns considering regression problems, such as estimation of time-related service level objectives, in a multi-case scenario. Finally, we also plan to develop methods to learn and adapt the best possible combination of weights values in the reliability metric based on historical process execution data.

REFERENCES

[1] Or Biran and Courtenay Cotton. 2017. Explanation and justification in machine learning: A survey. In *IJCAI-17 Workshop on Explainable AI (XAI)*. 8.

[2] Zoran Bosnić and Igor Kononenko. 2008. Comparison of approaches for estimating reliability of individual regression predictions. *Data & Knowledge Engineering* 67, 3 (2008), 504–516.

[3] Zoran Bosnić and Igor Kononenko. 2009. An overview of advances in reliability estimation of individual predictions in machine learning. *Intelligent Data Analysis* 13, 2 (2009), 385–401.

[4] Zoran Bosnić and Igor Kononenko. 2008. Estimation of individual prediction reliability using the local sensitivity analysis. *Applied Intelligence* 29, 3 (Dec. 2008), 187–203. https://doi.org/10.1007/s10489-007-0084-9

[5] Marco Comuzzi. 2017. Optimal Paths in Business Processes: Framework and Applications. In *Business Process Management Workshops*. 107–123. https://doi.org/10.1007/978-3-319-74030-0_7

[6] Adela Del-Río Ortega, Manuel Resinas, Cristina Cabanillas, and Antonio Ruiz-Cortés. 2013. On the definition and design-time analysis of process performance indicators. *Information Systems* 38, 4 (2013), 470–490.

[7] Adela del–Río–Ortega, Antonio Manuel Gutiérrez, Amador Durán, Manuel Resinas, and Antonio Ruiz–Cortés. 2015. Modelling Service Level Agreements for Business Process Outsourcing Services. In *Advanced Information Systems Engineering*. Springer, Cham, 485–500. https://doi.org/10.1007/978-3-319-19069-3_30

[8] Chiara Di Francescomarino, Marlon Dumas, Marco Federici, Chiara Ghidini, Fabrizio Maria Maggi, and Williams Rizzi. 2016. Predictive business process monitoring framework with hyperparameter optimization. In *Proc. CAiSE 2016*. Springer, 361–376.

[9] Rik Eshuis and Paul Grefen. 2008. Constructing customized process views. *Data & Knowledge Engineering* 64, 2 (2008), 419–438.

[10] Matjaž Kukar and Igor Kononenko. 2002. Reliable Classifications with Machine Learning. In *Machine Learning: ECML 2002*. Springer, Berlin, Heidelberg, 219–231. https://doi.org/10.1007/3-540-36755-1_19

[11] Philipp Leitner, Johannes Ferner, Waldemar Hummer, and Schahram Dustdar. 2013. Data-driven and automated prediction of service level agreement violations in service compositions. *Distributed and Parallel Databases* 31, 3 (April 2013), 447–470. https://doi.org/10.1007/s10619-013-7125-7

[12] Philipp Leitner, Anton Michlmayr, Florian Rosenberg, and Schahram Dustdar. 2010. Monitoring, prediction and prevention of sla violations in composite services. In *IEEE Int. Conf. on Web Services (ICWS)*. IEEE, 369–376.

[13] Anna Leontjeva, Raffaele Conforti, Chiara Di Francescomarino, Marlon Dumas, and Fabrizio Maria Maggi. 2015. Complex Symbolic Sequence Encodings for Predictive Monitoring of Business Processes. In *Proc. BPM 2015*. Springer International Publishing, 297–313. https://doi.org/10.1007/978-3-319-23063-4_21

[14] Fabrizio Maria Maggi, Chiara Di Francescomarino, Marlon Dumas, and Chiara Ghidini. 2014. Predictive monitoring of business processes. In *International Conference on Advanced Information Systems Engineering*. 457–472.

[15] A.E. Márquez-Chamorro, M. Resinas, A. Ruiz-Cortés, and M. Toro. 2017. Run-time prediction of business process indicators using evolutionary decision rules. *Expert Systems with Applications* 87, Supplement C (2017), 1–14.

[16] Andreas Metzger and Felix Föcker. 2017. Predictive business process monitoring considering reliability estimates. In *Proc. CAiSE 2017*. Springer, 445–460.

[17] Alfonso Eduardo Márquez-Chamorro, Manuel Resinas, and Antonio Ruiz-Cortes. 2017. Predictive monitoring of business processes: a survey. *IEEE Transactions on Services Computing* (2017).

[18] Arik Senderovich, Chiara Di Francescomarino, Chiara Ghidini, Kerwin Jorbina, and Fabrizio Maria Maggi. 2017. Intra and Inter-case Features in Predictive Process Monitoring: A Tale of Two Dimensions. In *Proc. BPM 2017*. 306–323.

[19] Wil MP van der Aalst and Christian W Gunther. 2007. Finding structure in unstructured processes: The case for process mining. In *7th Int. Conf. on Application of concurrency to system design*. IEEE, 3–12.

[20] W. M. P. van der Aalst, M. H. Schonenberg, and M. Song. 2011. Time prediction based on process mining. *Information Systems* 36, 2 (April 2011), 450–475. https://doi.org/10.1016/j.is.2010.09.001

[21] Kevin Woods, W. Philip Kegelmeyer, and Kevin Bowyer. 1997. Combination of multiple classifiers using local accuracy estimates. *IEEE transactions on pattern analysis and machine intelligence* 19, 4 (1997), 405–410.